



MVI56-DFCM

ControlLogix Platform

DF1 Half/Full Duplex Master/Slave Serial
Communication Module

December 3, 2020

Your Feedback Please

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

ProSoft Technology, Inc.

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

© 2020 ProSoft Technology, Inc. All rights reserved.

MVI56-DFCM User Manual

December 3, 2020

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

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 at: www.prosoft-technology.com

Important Installation Instructions

Power, Input, and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction. The following warnings must be heeded:

WARNING - EXPLOSION HAZARD - SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS I, DIV. 2;

WARNING - EXPLOSION HAZARD - WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

THIS DEVICE SHALL BE POWERED BY CLASS 2 OUTPUTS ONLY.

MVI (Multi Vendor Interface) Modules

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

AVERTISSEMENT - RISQUE D'EXPLOSION - AVANT DE DÉCONNECTER L'ÉQUIPEMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

Warnings

North America Warnings

Power, Input, and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction.

The following warnings must be heeded:

- A** Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.
- B** Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or rewiring modules.
- C** Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
Avertissement - Risque d'explosion - Avant de déconnecter l'équipement, couper le courant ou s'assurer que l'emplacement est désigné non dangereux.
- D** Suitable for use in Class I, Division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

ATEX Warnings and Conditions of Safe Usage

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

- A** Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.
- B** Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
- C** These products are intended to be mounted in an IP54 enclosure. The devices shall provide external means to prevent the rated voltage being exceeded by transient disturbances of more than 40%. This device must be used only with ATEX certified backplanes.
- D** DO NOT OPEN WHEN ENERGIZED.

Battery Life Advisory

The MVI46, MVI56, MVI56E, MVI69, and MVI71 modules use a rechargeable Lithium Vanadium Pentoxide battery to backup the 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 the battery becomes fully charged. After it is fully charged, the battery provides backup power for the CMOS setup and the real-time clock for approximately 21 days. When the battery is fully discharged, the module will revert to the default BIOS and clock settings.



For professional users in the European Union

If you wish to discard electrical and electronic equipment (EEE), please contact your dealer or supplier for further information.



Prop 65 Warning – Cancer and Reproductive Harm – www.P65Warnings.ca.gov

Agency Approvals & Certifications

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

Contents

Your Feedback Please.....	2
Important Installation Instructions	2
MVI (Multi Vendor Interface) Modules	2
Warnings	3
Battery Life Advisory	3
Guide to the MVI56-DFCM User Manual	6
1 Start Here	7
1.1 System Requirements	8
1.2 Package Contents	8
1.3 Setting Jumpers	9
1.4 Installing the Module in the Rack	10
1.5 Connecting Your PC to the ControlLogix Processor	11
1.6 Using the Sample Ladder Logic	12
1.7 Connecting Your PC to the ControlLogix Processor	12
1.8 Downloading the Sample Program to the Processor	14
1.8.1 Configuring the RSLinx Driver for the PC COM Port	15
1.9 Connecting Your PC to the Module.....	17
2 Using the RSLogix 5000 v16 Add-On Instruction	20
2.1 Creating a New RSLogix 5000 Project.....	20
2.2 Creating the Module	22
2.3 Importing the Add-On Instruction Ladder Rung	25
2.4 Adding Multiple Modules (Optional)	28
3 Ladder Logic	33
3.1 Module Data Object (DFCMModuleDef)	33
3.2 Configuration Objects (DFCMCONFIG)	33
3.2.1 Module Definition (DFCMModule).....	34
3.2.2 DF1 Port Parameters (DFCMPort)	35
3.2.3 DF1 Master Commands (DFCMCmd).....	36
3.3 User Data Objects (DFMC.DATA)	37
3.4 Status Object (DFCM.STATUS).....	38
3.5 Special Functions and Controls (DFCM.CONTROL)	39
3.6 Special Functions and Controls (DFCM.CONTROL)	41
3.7 Adding the Module to an Existing Project	41
4 Diagnostics and Troubleshooting	44
4.1 LED Status Indicators.....	45
4.1.1 Clearing a Fault Condition.....	47
4.1.2 Troubleshooting.....	48
4.2 Reading Status Data from the Module	49
4.2.1 The Configuration/Debug Menu	49
4.2.2 Main Menu.....	51

4.2.3	Data Analyzer	55
4.2.4	Database View Menu	59
4.2.5	Master Command Error List Menu	61
4.2.6	Master Command List Menu	62
5	Reference	63
5.1	Product Specifications	63
5.1.1	General Specifications	63
5.1.2	Hardware Specifications	64
5.1.3	Functional Specifications	65
5.2	Functional Overview	66
5.2.1	General Concepts	66
5.2.2	Data Flow between MVI56-DFCM Module and ControlLogix Processor	70
5.2.3	Normal Data Transfer	74
5.2.4	Configuration Data Transfer Blocks (-9000, -6000 to -6004, -6100 to -6104, 6000 to 6004, 6100- to 6104, and 9000)	77
5.2.5	Event Command Blocks (1000 or 2000)	80
5.2.6	Slave Status Blocks (3002, 3003, 3102, and 3103)	81
5.2.7	Command Control Blocks (5000 to 5006 or 5100 to 5106)	83
5.2.8	Override File Map Configuration Blocks (7000 and 7100, 50 entries per port)	84
5.2.9	Module Time Blocks (9972 and 9973)	84
5.2.10	Block 9999: Cold Boot	85
5.2.11	Warm Boot Block (9998) MVI56-DFCM	86
5.3	DFCM Database Definition	86
5.3.1	Basic Command Set Functions	87
5.3.2	PLC-5 Command Set Functions	88
5.3.3	SLC-500 Command Set Functions	88
5.4	Error Codes	90
5.4.1	Local STS Error Codes	90
5.4.2	Remote STS Error Codes	90
5.4.3	Errors When EXT STS Is Present	91
5.4.4	Module Specific Error (not DFCM Compliant)	91
5.5	Cable Connections	92
5.5.1	RS-232 Configuration/Debug Port	92
5.5.2	RS-232 Application Port(s)	92
5.5.3	RS-422	95
5.5.4	RS-485 Application Port(s)	95
5.5.5	DB9 to RJ45 Adaptor (Cable 14)	96
6	Support, Service & Warranty	97
6.1	Contacting Technical Support	97
6.2	Warranty Information	97

Guide to the MVI56-DFCM User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 7)	This section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Diagnostic and Troubleshooting	→	Diagnostics and Troubleshooting (page 44)	This section describes Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview	→	Reference (page 63) Product Specifications Functional Overview	These sections contain general references associated with this product, Specifications and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service & Warranty (page 97) Index	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

❖ System Requirements	8
❖ Package Contents	8
❖ Setting Jumpers.....	9
❖ Installing the Module in the Rack.....	10
❖ Using the Sample Ladder Logic	12
❖ Connecting Your PC to the ControlLogix Processor.....	12
❖ Downloading the Sample Program to the Processor	14
❖ Connecting Your PC to the Module	16

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

- **Rockwell Automation® RSLogix™ software:** launch the program, configure ladder logic, and transfer the ladder logic to the processor
- **Microsoft Windows®:** install and launch programs, execute menu commands, navigate dialog boxes, and enter data
- **Hardware installation and wiring:** install the module, and safely connect DFCM and ControlLogix devices to a power source and to the MVI56-DFCM module's application port(s)

1.1 System Requirements

The MVI56-DFCM module requires the following minimum hardware and software components:

- Rockwell Automation ControlLogix™ processor, with compatible power supply and one free slot in the rack, for the MVI56-DFCM module. The module requires 800 mA of available power.
- Rockwell Automation RSLogix 5000 programming software version 2.51 or higher
- Rockwell Automation RSLinx communication software
- Pentium® II 450 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
 - Microsoft Windows 10
 - Microsoft Windows 7 Professional (32-or 64-bit)
 - Microsoft Windows XP Professional with Service Pack 1 or 2
 - Microsoft Windows Vista
 - 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)
- ProSoft Configuration Builder, HyperTerminal or other terminal emulator program.

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

1.2 Package Contents

The following components are included with your MVI56-DFCM module, and are all required for installation and configuration.

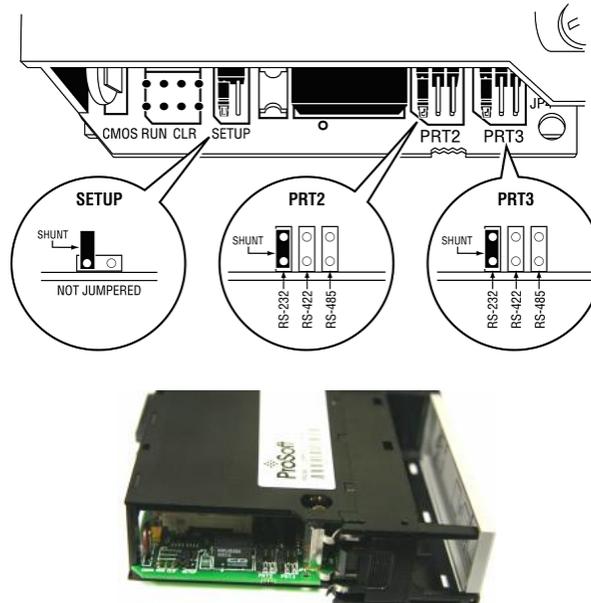
Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MVI56-DFCM Module	MVI56-DFCM	DF1 Half/Full Duplex Master/Slave Serial Communication Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	For DB9 Connection to Module's Port
2	Adapter	1454-9F	(2) DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module

1.3 Setting Jumpers

If you use an interface other than RS-232 (default), you must change the jumper configuration to select the interface you wish to use. There are three jumpers located at the bottom of the module.

The following illustration shows the MVI56-DFCM jumper configuration:



- 1 Set the PRT 2 (for application port 1) and PRT 3 (for application port 2) jumpers select RS232, RS422, or RS485 to match the wiring needed for your application. The default jumper setting for both application ports is RS-232.
- 2 The Setup Jumper acts as "write protection" for the module's flash memory. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support.

1.4 Installing the Module in the Rack

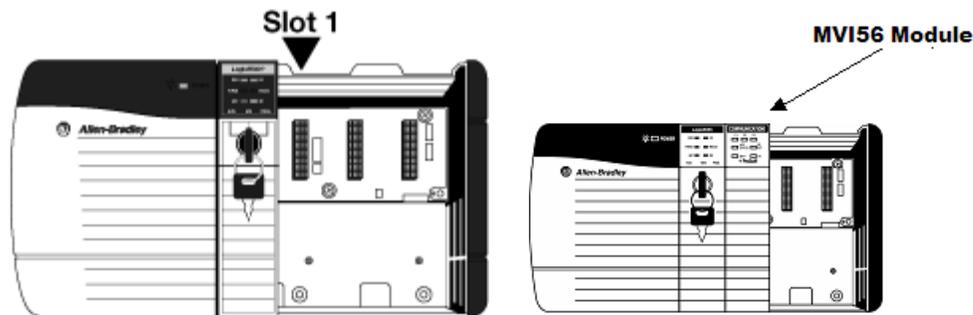
If you have not already installed and configured your ControlLogix processor and power supply, please do so before installing the MVI56-DFCM module. Refer to your Rockwell Automation product documentation for installation instructions.

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

After you have checked the placement of the jumpers, insert MVI56-DFCM into the ControlLogix chassis. Use the same technique recommended by Rockwell Automation to remove and install ControlLogix modules.

Warning: When you insert or remove the module while backplane power is on, an electrical arc can occur. This could cause an explosion in hazardous location installations. Verify that power is removed or the area is non-hazardous before proceeding. Repeated electrical arcing causes excessive wear to contacts on both the module and its mating connector. Worn contacts may create electrical resistance that can affect module operation.

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



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

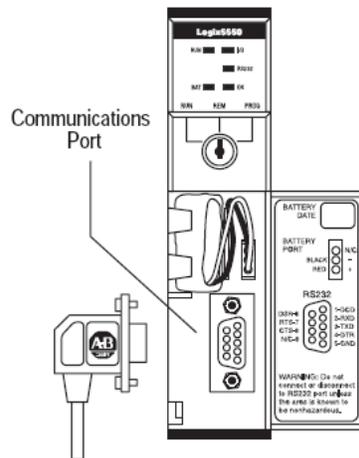
Note: If you insert the module improperly, the system may stop working, or may behave unpredictably.

1.5 Connecting Your PC to the ControlLogix Processor

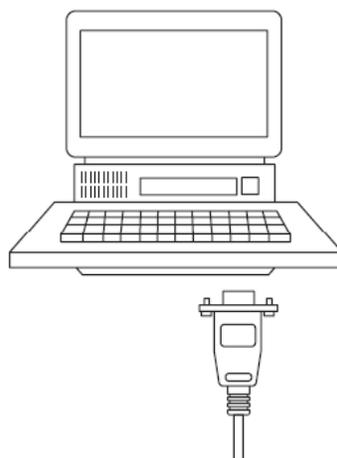
There are several ways to establish communication between your PC and the ControlLogix processor. The following steps show how to establish communication through the serial interface.

Note: It is not mandatory that you use the processor's serial interface. You may access the processor through whatever network interface is available on your system. Refer to your Rockwell Automation documentation for information on other connection methods.

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



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



1.6 Using the Sample Ladder Logic

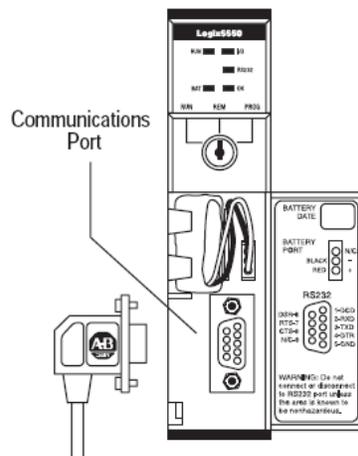
The sample program for your MVI56-DFCM module includes custom tags, data types, and ladder logic for data I/O and status monitoring. For most applications, you can run the sample ladder program without modification, or, for advanced applications, you can incorporate the sample program into your existing application.

1.7 Connecting Your PC to the ControlLogix Processor

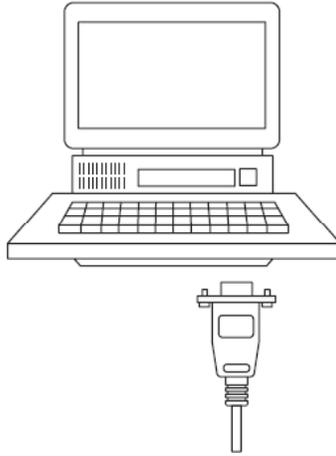
There are several ways to establish communication between your PC and the ControlLogix processor. The following steps show how to establish communication through the serial interface.

Note: It is not mandatory that you use the processor's serial interface. You may access the processor through whatever network interface is available on your system. Refer to your Rockwell Automation documentation for information on other connection methods.

- 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.



1.8 Downloading the Sample Program to the Processor

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

- 1 If you are not already online with the processor, in RSLogix 5000 open the *Communications* menu, and then choose **DOWNLOAD**. RSLogix 5000 will establish communication with the processor. You do not have to download through the processor's serial port, as shown here. You may download through any available network connection.
- 2 When communication is established, RSLogix 5000 will open a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.



- 3 RSLogix 5000 will compile the program and transfer it to the processor. This process may take a few minutes.
- 4 When the download is complete, RSLogix 5000 will open another confirmation dialog box. If the key switch is in the REM position, click **OK** to switch the processor from PROGRAM mode to RUN mode.

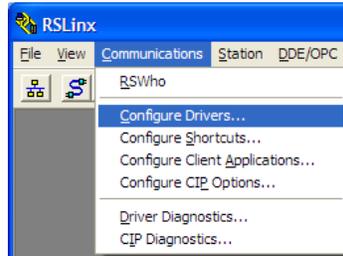


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

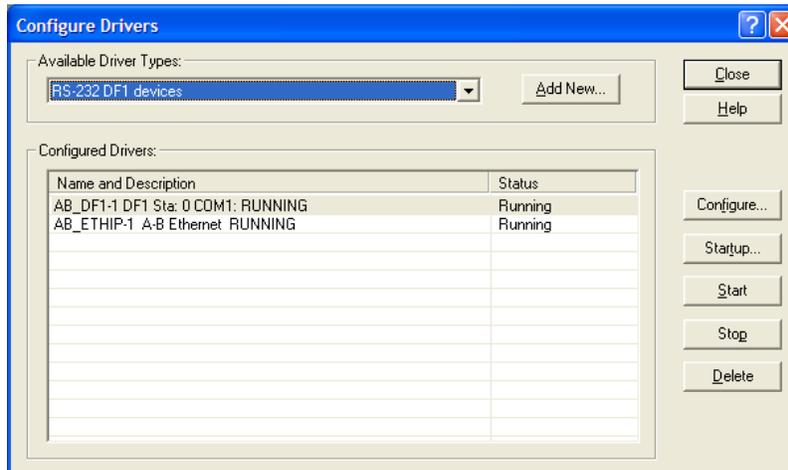
1.8.1 Configuring the RSLinx Driver for the PC COM Port

When trying to connect serially, if RSLogix is unable to establish communication with the processor, follow these steps.

- 1 Open *RSLinx*.
- 2 Open the **COMMUNICATIONS** menu, and click **CONFIGURE DRIVERS**.

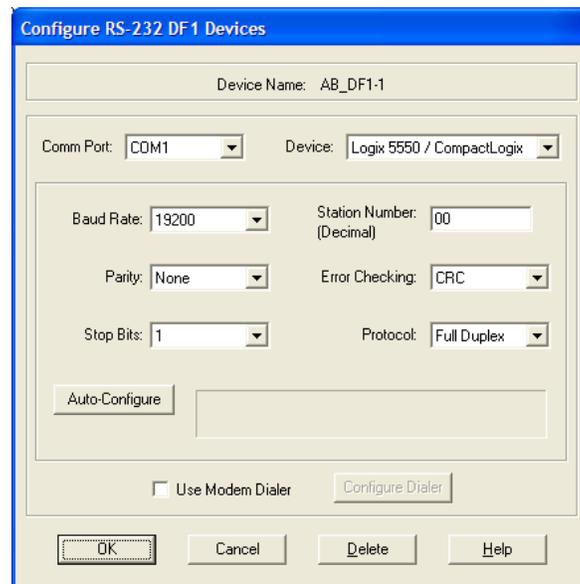


This action opens the *Configure Drivers* dialog box.



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 RS-232 DF1 Devices* dialog box.



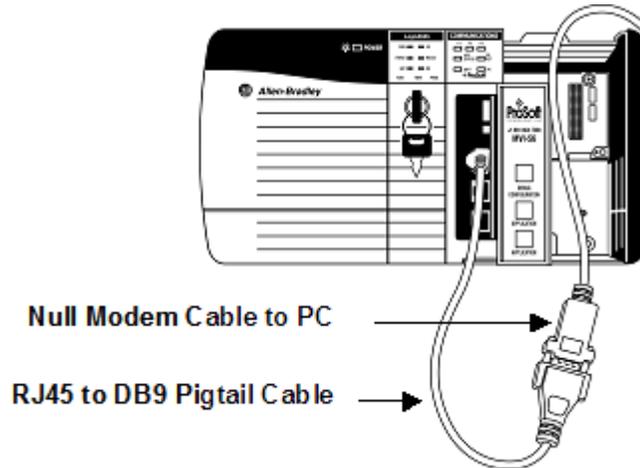
- 4 Click the **AUTO-CONFIGURE** button. RSLinx will attempt to configure your serial port to work with the selected driver.
- 5 When you see the message *Auto Configuration Successful*, click the **OK** button to dismiss the dialog box.

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.

1.9 Connecting Your PC to the Module

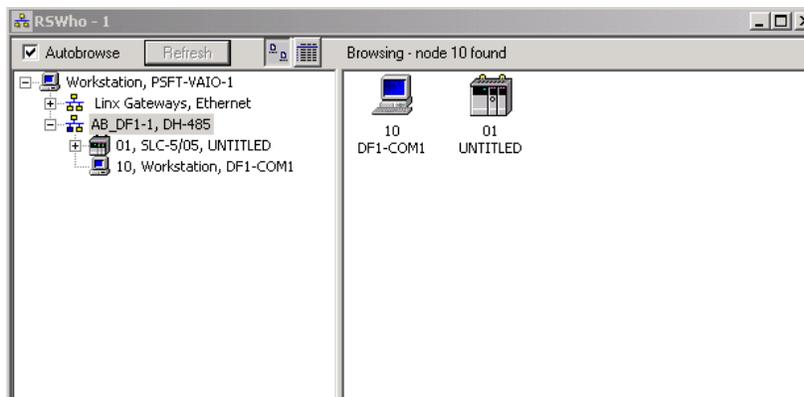
With the module securely mounted, connect your PC to the **Configuration/Debug** port using an RJ45-DB-9 Serial Adapter Cable and a Null Modem Cable.

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

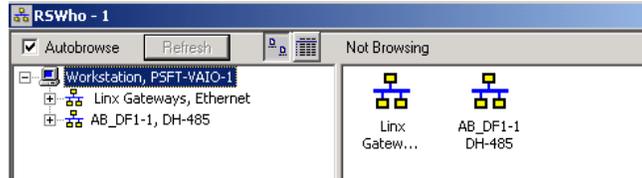


The communication port driver in *RSLinx* can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using *ProSoft Configuration Builder (PCB)*, *HyperTerminal* or another terminal emulator, follow these steps to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

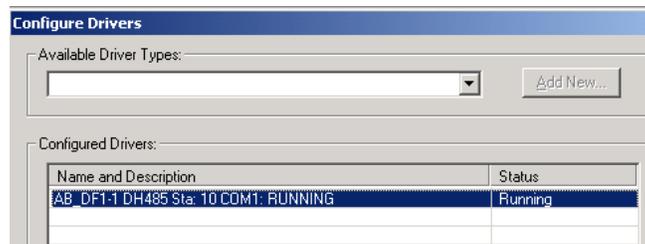


Branches are displayed or hidden by clicking on the  or the  icons.



- 4 When you have verified that the driver is not being browsed, go to **COMMUNICATIONS > CONFIGURE DRIVERS**.

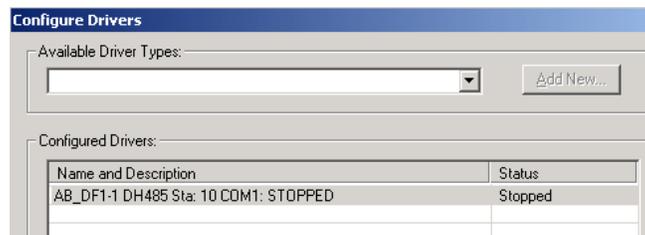
You may see something like this:



- 5 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** button on the side of the window:



- 6 After you have stopped the driver you will see the following.



7 You may now use the COM port to connect to the *Config/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 open, you will not be able to stop the DF1 driver. If *RSLogix* is not open, and you still cannot stop the driver, then reboot your PC.

2 Using the RSLogix 5000 v16 Add-On Instruction

In This Chapter

- ❖ Creating a New RSLogix 5000 Project 20
- ❖ Creating the Module 21
- ❖ Importing the Add-On Instruction Ladder Rung 24
- ❖ Adding Multiple Modules (Optional) 27

Important: If you are using an older version of RSLogix 5000 (version 15 or older), you will not be able to use the Add-On Instruction. Please see www.prosoft-technology.com for standard ladder logic examples.

If you have RSLogix 5000 version 16 or newer, you can use an Add-On Instruction to simplify the task of configuring the module, either as a new application, or within an existing application.

Download the manuals and sample program from the ProSoft Technology web site

You can always download the latest version of the sample ladder logic and user manuals for the MVI56-DFCM module from the ProSoft Technology web site, at www.prosoft-technology.com/support/downloads

From that link, navigate to the download page for your module and choose the sample ladder program to download for your version of RSLogix 5000 and your processor.

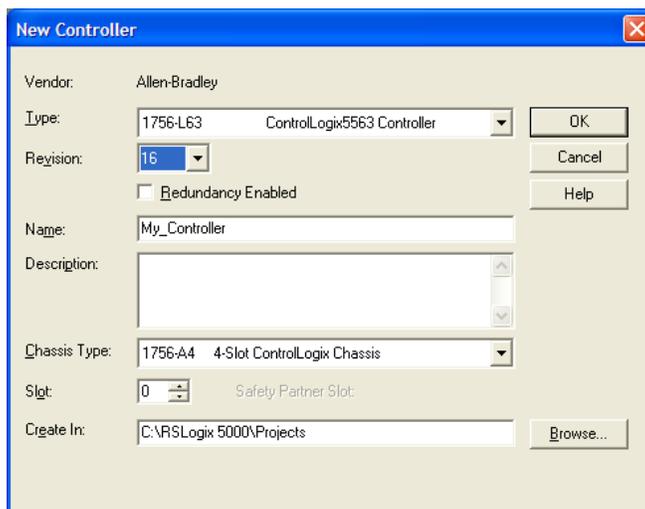
2.1 Creating a New RSLogix 5000 Project

- 1 Open the **FILE** menu, and then choose **NEW**.



- 2 Select your ControlLogix controller model.
- 3 Select the **REVISION** of your controller. Depending on the revision, there may be some small differences in the appearance of dialog boxes.
- 4 Enter a name for your controller, such as *My_Controller*.
- 5 Select your ControlLogix chassis type.

6 Select **SLOT 0** for the controller.



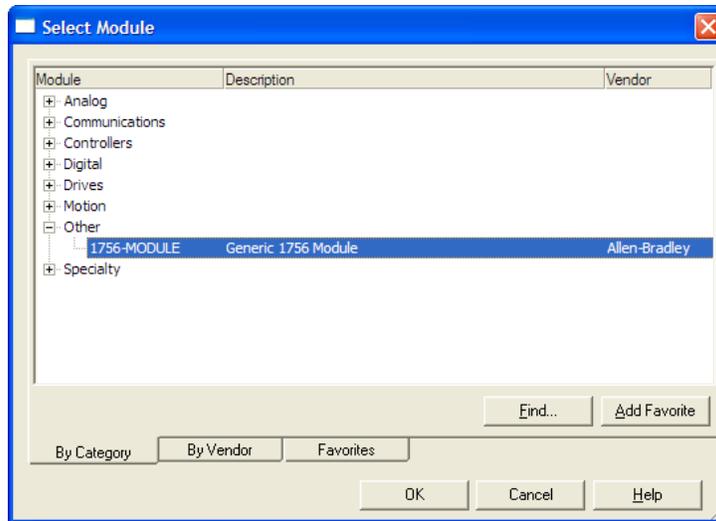
2.2 Creating the Module

- 1 Add the MVI56-DFCM module to the project.

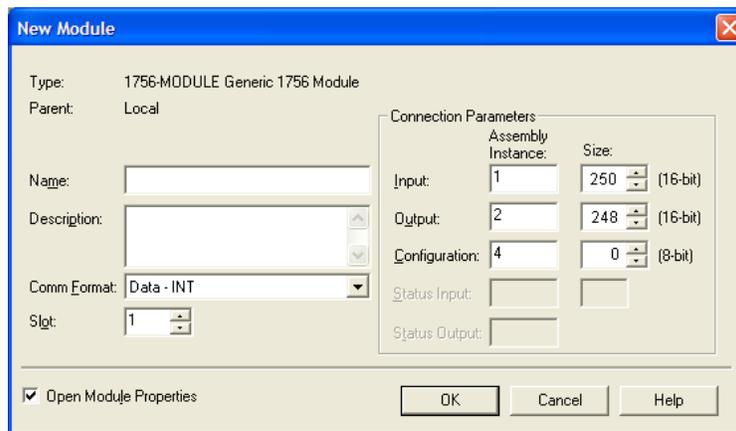
In the **CONTROLLER ORGANIZATION** window, select **I/O CONFIGURATION** and click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW MODULE...**



This action opens the **SELECT MODULE** dialog box.



- 2 Select the **1756-MODULE (GENERIC 1756 MODULE)** from the list and click **OK**. This action opens the **NEW MODULE** dialog box.



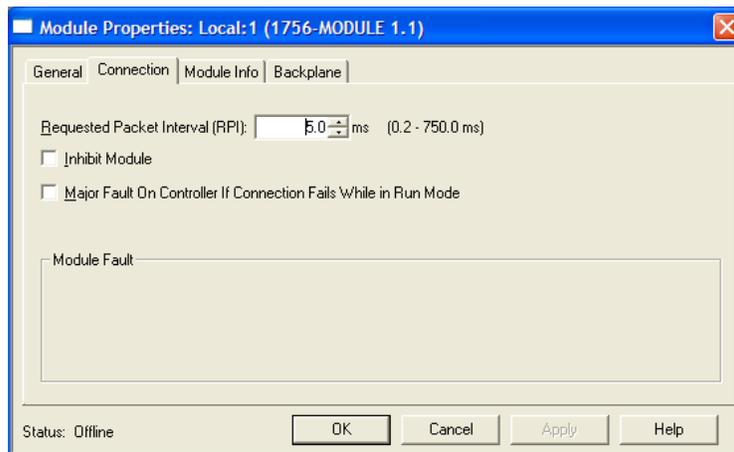
3 In the **NEW MODULE** dialog box, enter the following values.

Parameter	Value
NAME	DFCM
DESCRIPTION	Enter a description for the module. Example: DF1 Half/Full Duplex Master/Slave Serial Communication Module
COMM FORMAT	Select DATA-INT
SLOT	Enter the slot number in the rack where the MVI56-DFCM module is located
INPUT ASSEMBLY INSTANCE	1
INPUT SIZE	250
OUTPUT ASSEMBLY INSTANCE	2
OUTPUT SIZE	248
CONFIGURATION ASSEMBLY INSTANCE	4
CONFIGURATION SIZE	0

Important: You must select the **COMM FORMAT** as **DATA - INT** in the dialog box, otherwise the module will not communicate over the backplane with the ControlLogix processor.

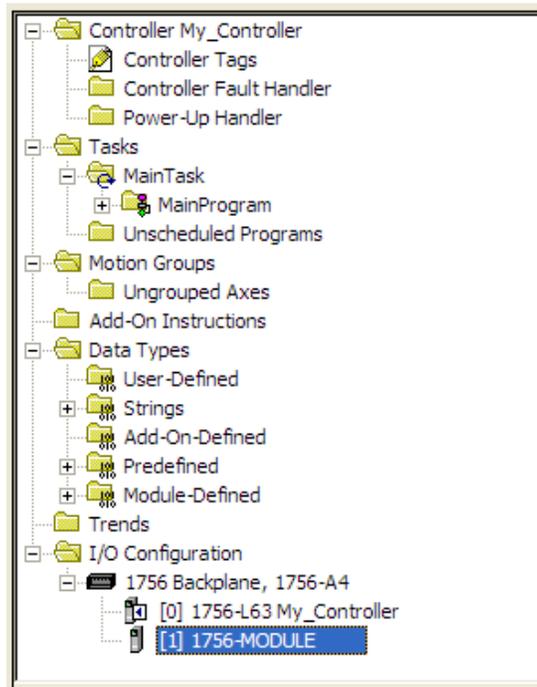
- 4** Click **OK** to continue.
- 5** Edit the Module Properties.

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



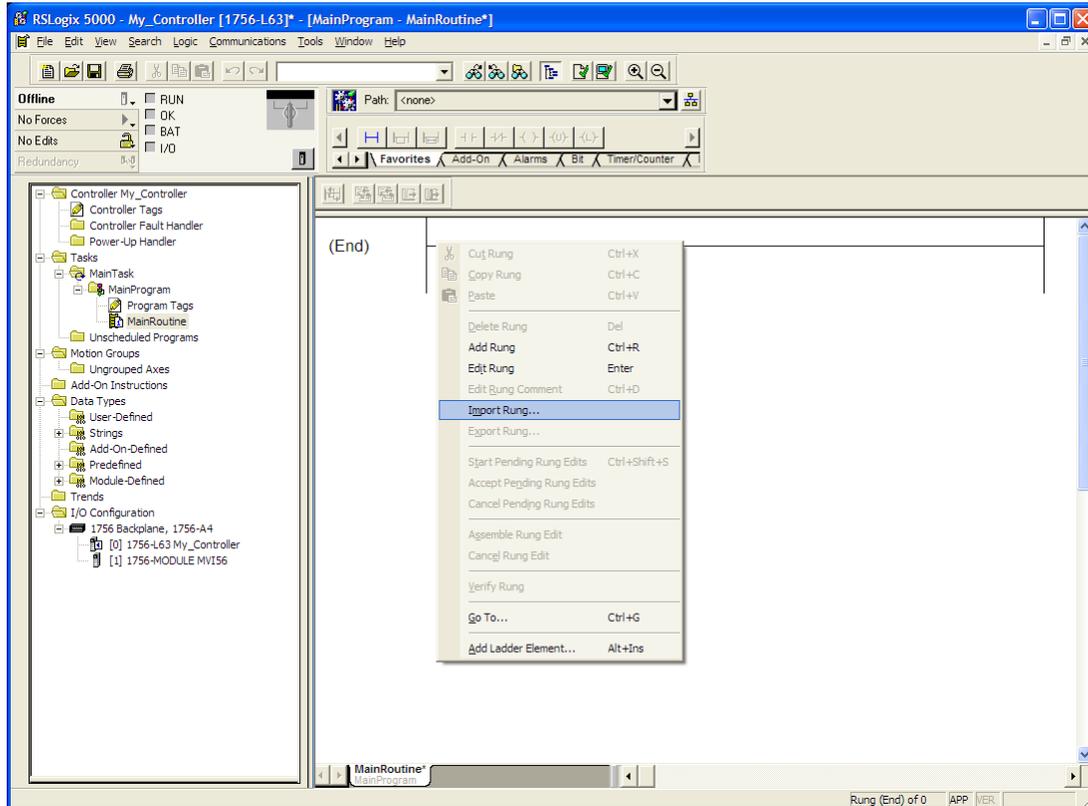
- 6** Save the module.

Click **OK** to close the dialog box. Notice that the module now appears in the **CONTROLLER ORGANIZATION** window.

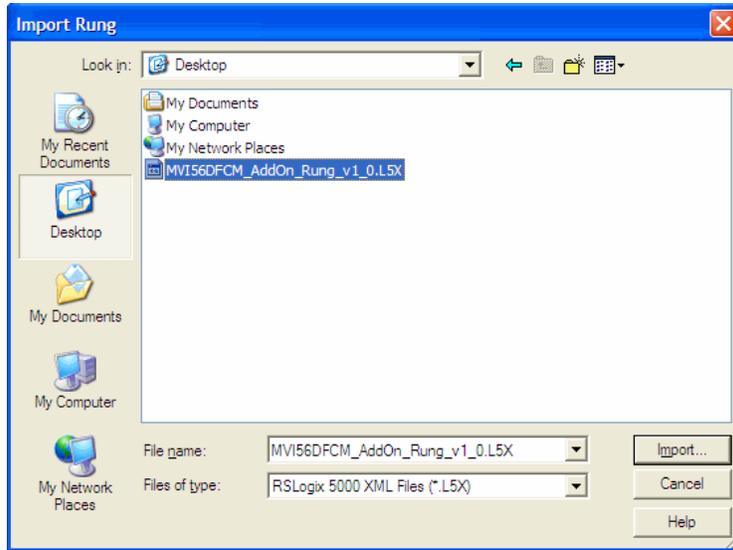


2.3 Importing the Add-On Instruction Ladder Rung

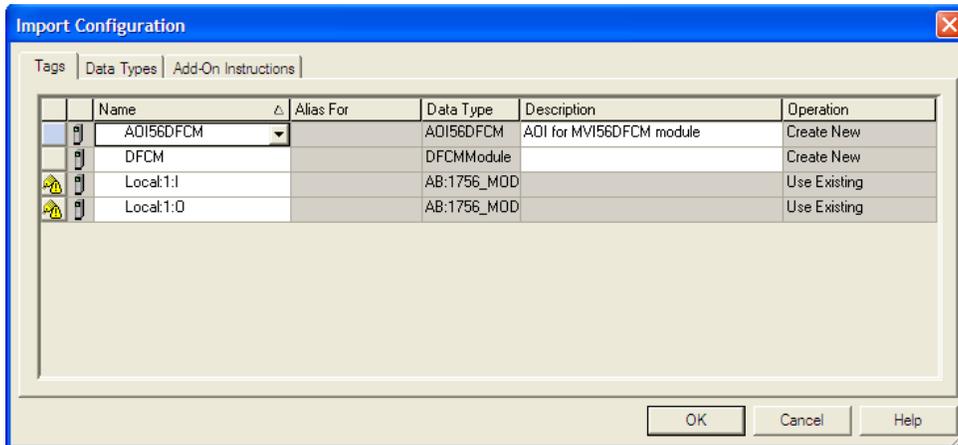
- 1 In the **CONTROLLER ORGANIZATION** window, expand the **TASKS** folder and subfolders until you reach the **MAINPROGRAM** folder.
- 2 In the **MAINPROGRAM** folder, double-click to open the **MAINROUTINE** ladder.
- 3 Select an empty rung in the routine, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **IMPORT RUNG...**



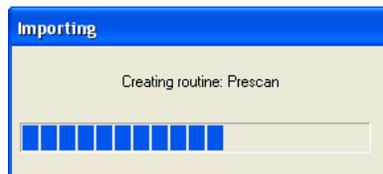
- Navigate to the location on your PC where you saved the Add-On Instruction (for example, *My Documents* or *Desktop*). Select the **MVI56DFCM_ADDON_RUNG_vX_Y.L5X** file



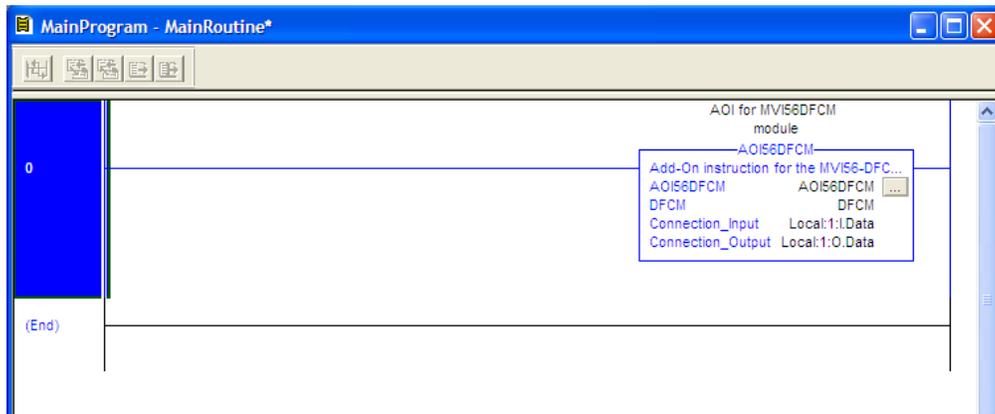
This action opens the **IMPORT CONFIGURATION** dialog box, showing the controller tags that will be created.



- If you are using the module in a different slot (or remote rack), select the correct connection input and output variables that define the path to the module. If your module is located in Slot 1 of the local rack, this step is not required.
- Click **OK** to confirm the import. RSLogix will indicate that the import is in progress:



When the import is completed, the new rung with the Add-On Instruction will be visible as shown in the following illustration.



The procedure has also imported new User Defined Data Types, Controller Tags, and the Add-On instruction for your project.

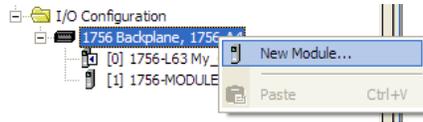


- 7 Save the application and then download the sample ladder logic into the processor.

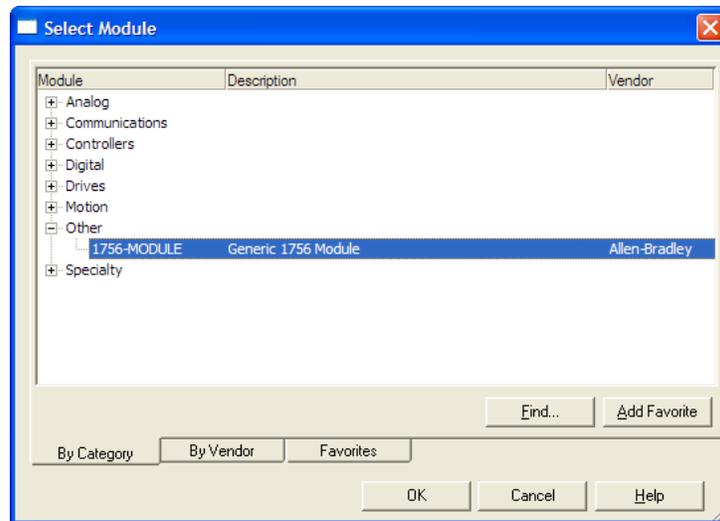
2.4 Adding Multiple Modules (Optional)

Important: If your application requires more than one MVI56-DFCM module in the same project, follow the steps below.

- 1 In the **I/O CONFIGURATION** folder, click the right mouse button to open a shortcut menu, and then choose **NEW MODULE**.



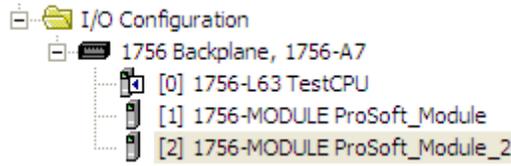
- 2 Select **1756-MODULE**



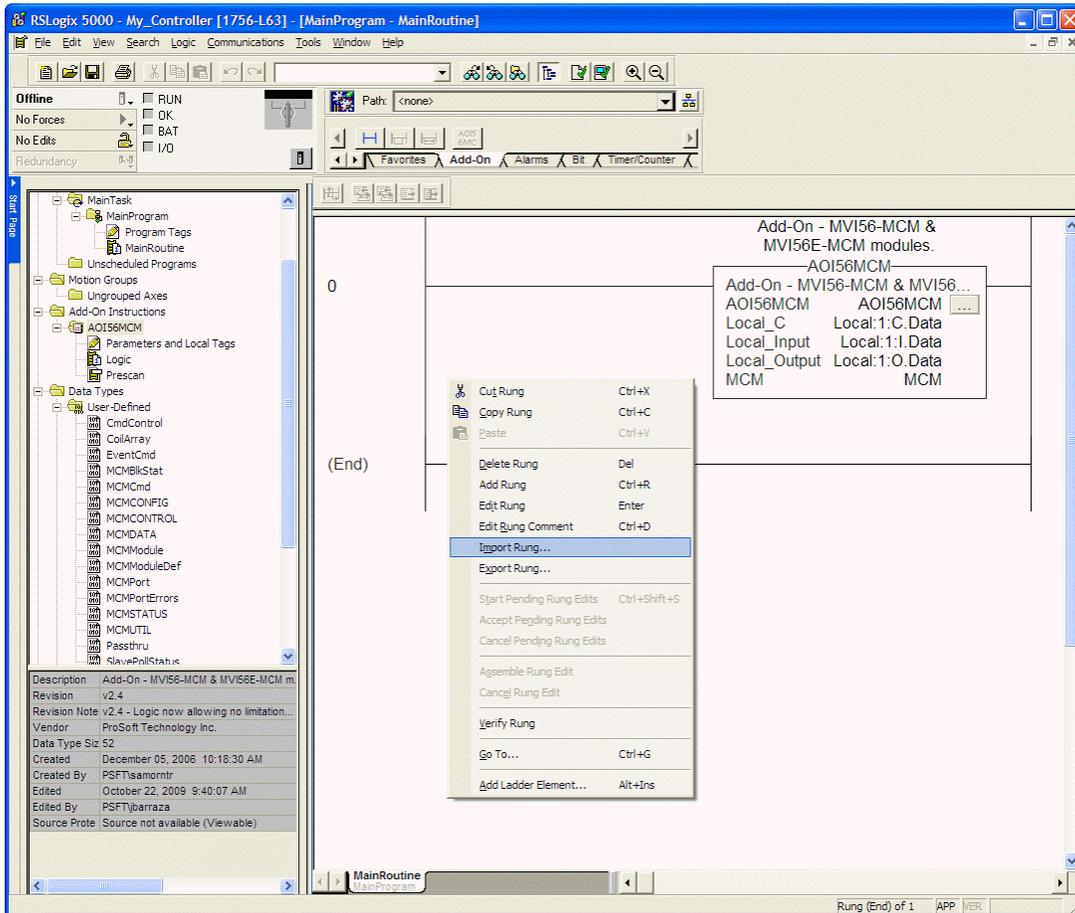
- 3 Fill the module properties as follows:

Parameter	Value
NAME	DFCM_2.
DESCRIPTION	Enter a description for the module. Example: ProSoft DF1 Communication Module.
COMM FORMAT	Select DATA-INT .
SLOT	Enter the slot number in the rack where the MVI56-DFCM module is located.
INPUT ASSEMBLY INSTANCE	1
INPUT SIZE	250
OUTPUT ASSEMBLY INSTANCE	2
OUTPUT SIZE	248
CONFIGURATION ASSEMBLY INSTANCE	4
CONFIGURATION SIZE	0

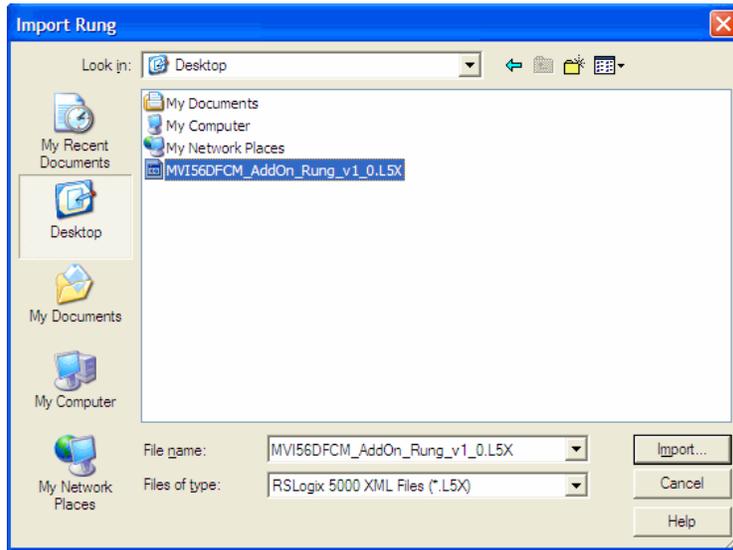
4 Click **OK** to confirm. The new module is now visible:



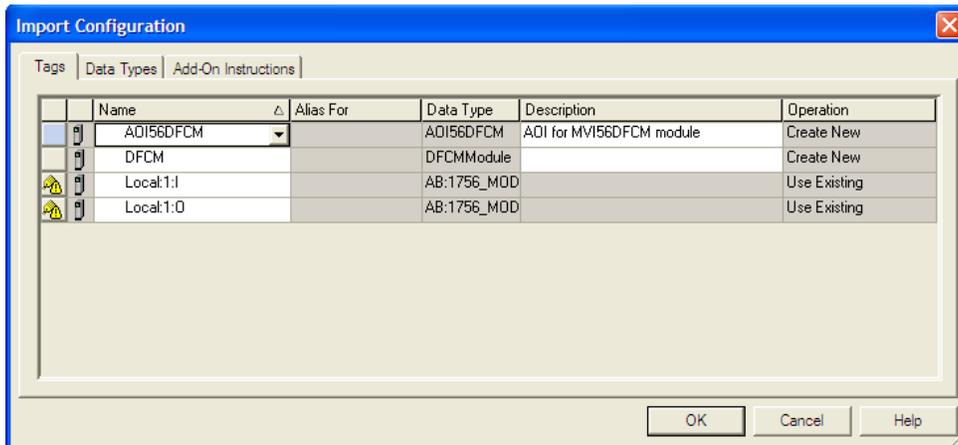
- 5 Expand the **TASKS** folder, and then expand the **MAINTASK** folder.
- 6 In the **MAINPROGRAM** folder, double-click to open the **MAINROUTINE** ladder.
- 7 Select an empty rung in the routine, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **IMPORT RUNG...**



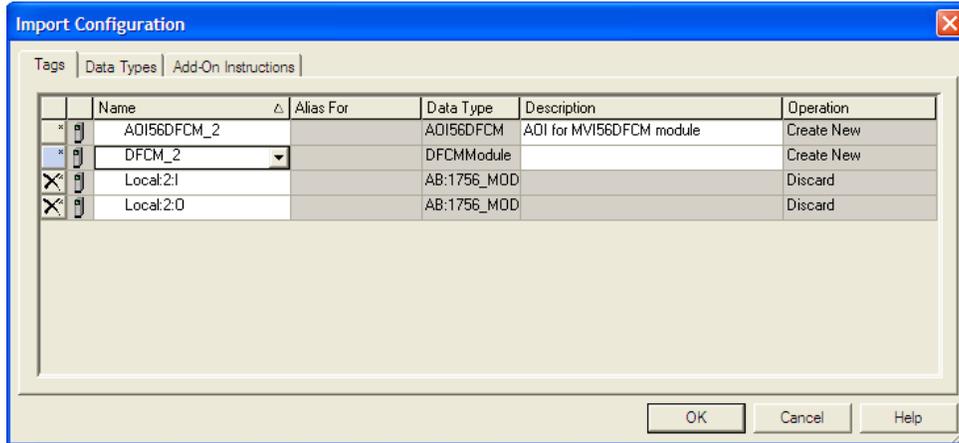
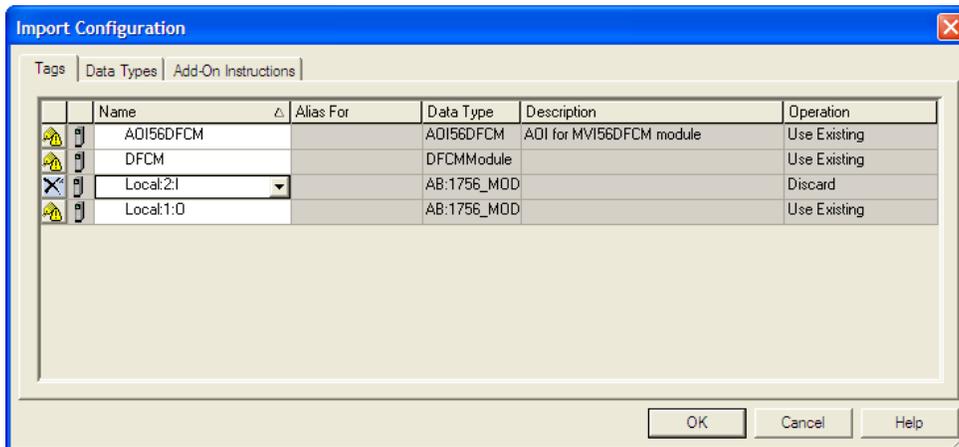
- 8 Select the **MVI56DFCM_ADDON_RUNG_VX_Y.L5X** file, and then click **IMPORT**.



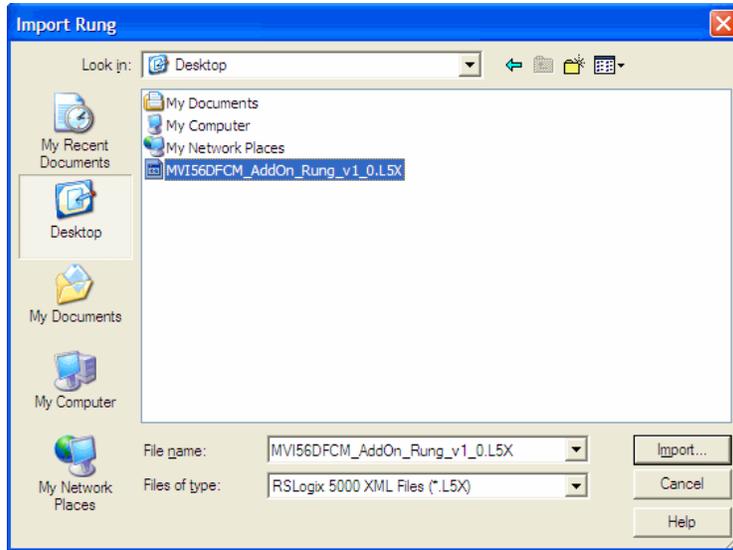
- 9 This action opens the **IMPORT CONFIGURATION** window, which shows the tags that will be imported.



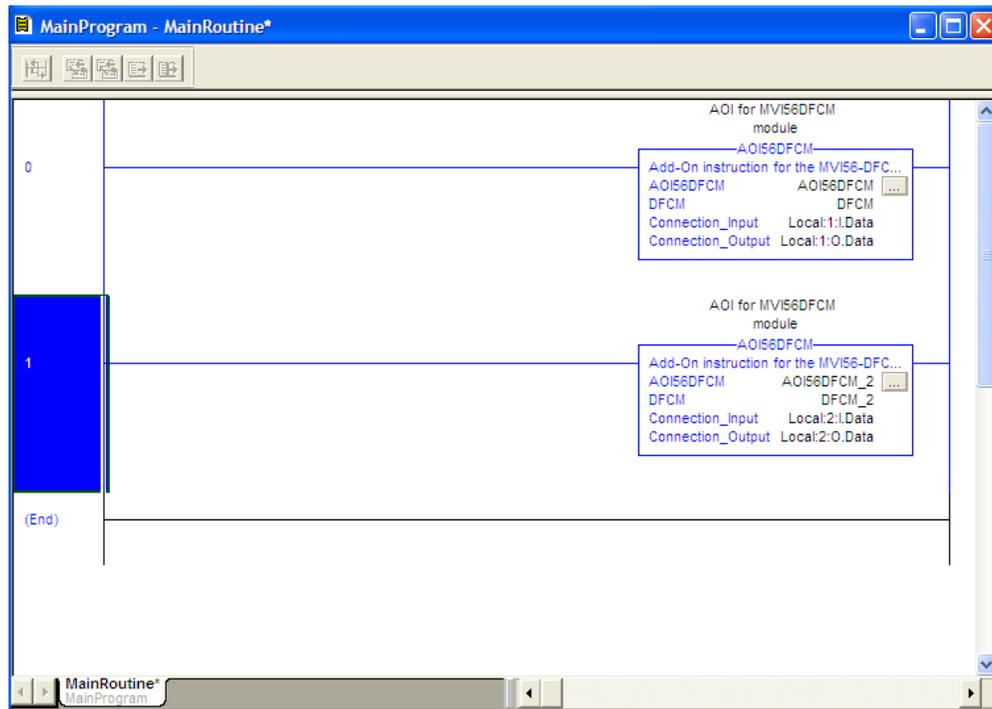
- 10** Associate the I/O connection variables to the correct module. The default values are Local:1:I and Local:1:O so these require change.



- 11 Change the default tags **DFCM** and **AOI56DFCM** to avoid conflict with existing tags. In this step, you should append a string to the default tag names, such as "_2", as shown in the following illustration.



- 12 Click **OK** to confirm.



The setup procedure is now complete. Save the project and download the application to your ControlLogix processor.

3 Ladder Logic

In This Chapter

- ❖ Module Data Object (DFCMModuleDef) 33
- ❖ Configuration Objects (DFCMCONFIG) 33
- ❖ User Data Objects (DFMC.DATA) 37
- ❖ Status Object (DFCM.STATUS) 37
- ❖ Special Functions and Controls (DFCM.CONTROL) 39
- ❖ Special Functions and Controls (DFCM.CONTROL) 41
- ❖ Adding the Module to an Existing Project 41

Ladder logic is required for the MVI56-DFCM module to work. 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 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 MVI56-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.

Name	Data Type	Description
CONFIG	DFCMCONFIG	Module and port configuration
DATA	DFCMDATA	DFCM data transferred between module and processor
STATUS	DFCMSTATUS	Status information in each Read Block
CONTROL	DFCMCONTROL	Optional requests from the processor to the module
UTIL	DFCMUTIL	Variables for internal ladder usage - should not be accessed by user application.
P1DATAFILEMAP	DFCMDataFileMap[50]	Port 1 Data File Mappings
P2DATAFILEMAP	DFCMDataFileMap[50]	Port 2 Data File Mappings

This object contains objects that define the configuration, user data, status and command control data related to the module.

3.2 Configuration Objects (DFCMCONFIG)

Configuration of the module is performed by filling in the values in the module object defined in the Controller Tags Edit Tags dialog box. Each parameter required by the module has a defined location in the object. The tables and discussions below describe the parameters set in the dialog box. You can view these tables by opening the data type under the User Defined Data Type option in the Controller Organization window.

3.2.1 Module Definition (DFCMModule)

Name	Data Type	Description
WriteStartReg	INT	Start reg to transfer from PLC to module
WriteRegCnt	INT	Number of registers to write from PLC
ReadStartReg	INT	Start reg to transfer from module to PLC
ReadRegCnt	INT	Number of registers to transfer from module
BPFail	INT	Determines module operation if BP fails 0=continue,>0=number of retries before comm shutdown
ErrStatPtr	INT	Internal DB start register for status data (-1=Ignore)

This object defines the parameters for data movement between the module and the processor. Values entered determine the ladder logic and data size required in the application. The ReadData and WriteData arrays must be sized to or larger than the count values entered. The ladder logic must process the number of blocks of data to be transferred. The number of blocks is computed as follows:

$$\text{BlockCnt} = \text{INT}(\text{RegCnt}/200) + \text{if}(\text{MOD}(\text{RegCnt},200), 1,0)$$

If the register count is evenly divisible by 200, the number of blocks is easy to compute and the ladder is much simpler to write. If the number is not evenly divisible by 200, special handling of the last block of data must developed, as it must transfer less than 200 words. **It is recommended that the count values always be set to values evenly divisible by 200.**

The BPFail parameter determines if the module should continue communicating on the DF1 network when the backplane transfer operation fails. A value of zero indicates that the module should continue communicating when the backplane is not operational. If the value is greater than zero, the backplane will be retried the entered number of times before a failure will be reported and communication will cease on the ports. When backplane communication is restored, the module will start communicating on the network. For example, if you enter a value of 10 for the parameter, the module will stop all DF1 communications if 10 successive backplane errors are recognized. When a successful transfer is recognized, the module will resume communications on the network.

The ErrStatPtr parameter defines the location in the module's database where the error/status data will be stored. If the value is set to -1, the data will not be stored in the user data area. A value between 0 and 4939 will cause the module's program to store the data at the specified location.

3.2.2 DF1 Port Parameters (DFCMPort)

Name	Data Type	Description
Enabled	INT	0=Port Disabled,1=Port Enabled
Type	INT	0=Master, 1=Slave
StationID	INT	0-254 local station id for device emulated on this port
Protocol	INT	0=Full-duplex, 1=half-duplex
TerminationType	INT	0=BCC, 1=CRC
Baudrate	INT	Baudrate for port (110 to 115.2K)
Parity	INT	0=None, 1=Odd, 2=Even, 3=Mark, 4=Space
DataBits	INT	5 to 8 data bits
StopBits	INT	1 or 2 stop bits
MinResp	INT	0-65535 mSec minimum time before response to request
RTSON	INT	0-65535 mSec delay before data
RTSOFF	INT	0-65535 mSec delay after data
UseCTS	INT	0=No, 1=Yes to use CTS modem line
ENQDelay	INT	0-65535 delay after ACK before sending ENQ request (half-duplex)
CmdCount	INT	Command list count (Master)
MinCmdDelay	INT	0-65535 mSec minimum time between each command (Master)
CmdErrPtr	INT	Internal DB location to place command error list (Master)
RespTO	INT	0-65535 mSec response timeout for command (Master)
Retry_Count	INT	Retry count for failed request (Master)
ErrorDelayCntr	INT	0-65535 Command cycle count if error (Master)
SlaveListPtr	INT	-1=No slave list in database, 0-4743=Reg start location for data (Master)
SlaveListFreq	INT	0=No slave list in database, 1-65535=cycle count for update (Master)
FirstFile	INT	0-100 First file to emulate in database (Slave)
FileSize	INT	0-1000 size of each file to emulate (Slave)
FileOffset	INT	0-4999 register offset into database where file emulation starts (Slave)
DataFileMapCnt	INT	0-50 Data File Mapping Count to Re-Direct data Reads and Writes (Slave)

This object defines the parameters for the operation of each of the DF1 ports on the module. Refer to Configuration Data Definition for the definition of each parameter.

3.2.3 DF1 Master Commands (DFCMCmd)

This object defines the parameters for each command in the Master command list. The **DFCMModuleDef** object contains an array of these objects that define the complete command list for each port. The definition of each parameter required for each command is given below:

Name	Data Type	Description
Enable	INT	0=Disable, 1=Continuous, 2=Write on Data Change only
IntAddress	INT	Module's internal address associated with the command as destination for data read or source for data written
PollInt	INT	Minimum number of seconds between issuance of command (0-65535 Sec)
Count	INT	Number of registers associated with the command
Swap	INT	Swap code used with command
Node	INT	Address of device to interface with on the DF1 network
Func	INT	DF1 function code as defined for MVI56-DFCM module
Parameter_1	INT	First parameter for function
Parameter_2	INT	Second parameter for function
Parameter_3	INT	Third parameter for function
Parameter_4	INT	Fourth parameter for function

Enable: This parameter defines if the command will be executed or will be disregarded. The following values are valid: 0=Disables the command and it will not execute. 1=The command will be considered for execution each scan of the command list and will be controlled by the **PollInt** parameter. And 2=The command must be a write command and will only execute if the data associated with the command has changed since the command was last issued. This option is only available for write commands.

IntAddress: This parameter specifies the starting internal register address to be associated with the command. Valid entry for this parameter is 0 to 4999.

PollInt: This parameter defines the minimum number of seconds to wait between the execution of continuous commands (Enable=1). This poll interval command can be used to lighten the communications load on a busy network. Valid entry for this parameter is 0 to 65535.

Count: This parameter defines the number of registers to be considered by the command. Valid entry for this parameter is 1 to 100.

Swap: This parameter specifies if the data used in the command must be altered when reading data from a node on the network. Values that can be assigned are as follows: 0=no swapping of data, 1=swap word values, 2=swap word and byte values and 3=swap byte values. This option is used when interfacing the module with ASCII and floating-point data on other devices.

Note: This parameter assigns the DF1 slave node address for the module to reach with the command on the network. This parameter can be assigned values from 0 to 255.

Func: This parameter specifies the function to be performed by the command. Valid entries are those defined in the **DF1 Command Set for ProSoft Technology Communication Modules** found at the end of this manual.

Parameter_1 to Parameter_4: These are the parameters required for the selected function. Each command has its own unique set of one or more parameters. Refer to the DF1 Command Set document for a complete list of command parameters.

3.3 User Data Objects (DFMC.DATA)

These objects hold data to be transferred between the processor and the MVI56-DFCM module. User data is read and write data, transferred between the processor and the module as "pages" of data up to 200, 16-bit words long.

Name	Data Type	Description
ReadData	INT[600]	Data read from module
WriteData	INT[600]	Data to write to module

ReadData is an array set to match the value entered in the **ReadRegCnt** parameter of the **DFCMModule** object. For ease of use, this array should be dimensioned as an even increment of 200 words. This data is paged up to 200 words at a time from the module to the processor. 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.

WriteData is an array set to match the value entered in the **WriteRegCnt** parameter of the **DFCMModule** object. For ease of use, this array should be dimensioned as even increments of 200 words. This data is paged up to 200 words at a time from the processor to the module. The *WriteData* task is responsible for placing the write data into the output image for transfer to the module. This data is passed from the processor to the module for status and control use in other nodes on the network.

3.4 Status Object (DFCM.STATUS)

This object views the status of the module. The **DFCMInStat** 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
PassCnt	INT	Program cycle counter
Product	INT[2]	Product Name
Rev	INT[2]	Revision Level Number
OP	INT[2]	Operating Level Number
Run	INT[2]	Run Number
PrtErrs	DFCMPortErrors	Port error statistics
CmdReq	INT	Total number of command list requests sent
CmdResp	INT	Total number of command list responses received
CmdErr	INT	Total number of command list errors
Requests	INT	Total number of requests for port
Responses	INT	Total number of responses for port
ErrSent	INT	Total number of errors sent
ErrRec	INT	Total number of errors received
BlkErrs	DFCMBlkStat	Block transfer statistics
Read	INT	Total number of read block transfers
Write	INT	Total number of write block transfers
Parse	INT	Total number of blocks parsed
Event	INT	Total number of event blocks received
Cmd	INT	Total number of command blocks received
Err	INT	Total number of block transfer errors
P1CErr	INT	Port 1 current error/index
P1LErr	INT	Port 1 last error/index
P2CErr	INT	Port 2 current error/index
P2LErr	INT	Port 2 current error/index
SlavePollStatus	DFCMSlavePollStatus	This object contains all of the Slave Polling status (used when the port is used as a Master)
P1Slaves	INT[256]	Slave status after polling for slave status information (each array element is the status for one slave.)
P2Slaves	INT[256]	Slave status after polling for slave status information (each array element is the status for one slave.)

Refer to Status Data Definition for a complete listing of the data stored in this object.

3.5 Special Functions and Controls (DFCM.CONTROL)

The DFCMCONTROL object encapsulates and organizes variables and parameters for several special module functions, as well as slave polling control.

Name	Data Type	Description
WarmBoot	BOOL	Warm Boot Trigger Bit
ColdBoot	BOOL	Cold Boot Trigger Bit
BootTimer	TIMER	Boot Timer
CmdControl	DFCMCmdControl	Allows a command that is disabled in the command list to be sent to a slave device
TriggerCmdControl	BOOL	Trigger Command Control. User application will activate this trigger.
NumberOfCommands	INT	Number of Command Indexes per CmdCtrl block, (1 to 6)
PortNumber	INT	MIV56-DFCM Port Number of Master Port, 1 or 2
CommandIndex	INT[6]	Stores the command indexes for command control
CmdsAddedToQueue	INT	Number of commands added to the queue
CmdControlBlockID	UNT	Temporary variable to calculate Control Block ID
CmdCtrlPending	BOOL	Auxiliary Control Command - prevents a second request before acknowledge is sent
EventCmd	DFCMEventCmd	Allows a Command defined in ladder to be sent to a device
EventCmdTrigger	BOOL	Trigger for event command. User ladder must set this bit to initiate event command
EventCmdPending	BOOL	Set after the ladder has sent an event cmd to the module and is waiting for the status to be returned
PortNumber	INT	Module Master port number associated to this request (1 or 2)
InternalDBAddress	INT	Internal database address
PointCount	INT	Number of points for this command
SwapCode	INT	Swap code (0= no swap, 1=swap words, 2=swap words and bytes, 3=swap bytes)
SlaveAddress	INT	Slave target modbus address
FunctionCode	INT	Function Code
Parameter1	INT	First Parameter for the function
Parameter2	INT	Second Parameter for the function
Parameter3	INT	Third Parameter for the function
Parameter4	INT	Fourth Parameter for the function
EventCmdStatusReturned	INT	(0=Fail, 1=Success)
EventBlockID	INT	Temporay variable to calculate event block ID

Name	Data Type	Description
SlaveControl	DFCMSlaveControl	Control Slave Polling for Slave Status
TriggerSlaveControl	BOOL	Trigger to Enable or Disable Slaves
PortNumber	INT	Slave Address
NumberOfSlaves	INT	Number of Slaves
BlockNumber	INT	Block ID number
SlaveIndexes	INT[200]	Slave Indexes
NumberOfSlavesProcessed	INT	Number of Slaves processed
EnableSlaves	BOOL	Set 0 to Disable or 1 to Enable Slaves
Port1Slave0Read	BOOL	Trigger to poll slave status
Port1Slave128Read	BOOL	Trigger to poll slave status
Port2Slave0Read	BOOL	Trigger to poll slave status
Port2Slave128Read	BOOL	Trigger to poll slave status
ModuleTime	DFCMModuleTime	Allows setting Module Time
Year	INT	Year (0 to 9999)
Month	INT	Month (1 to 12)
Day	INT	Day (1 to 31)
Hour	INT	Hour (0 to 23)
Minutes	INT	Minute (0 to 59)
Seconds	INT	Second (0 to 59)
GetModuleTime	BOOL	Trigger to read module time
SetModuleTime	BOOL	Trigger to set module time

Two arrays are allocated in the DFCM.STATUS object 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, which are in communication error, or which 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, using the SlaveControl object in this user-defined data type.

Using special blocks, the processor can request current polling status for the slaves. Through the use of other blocks, the processor can enable or disable the polling of selected slaves.

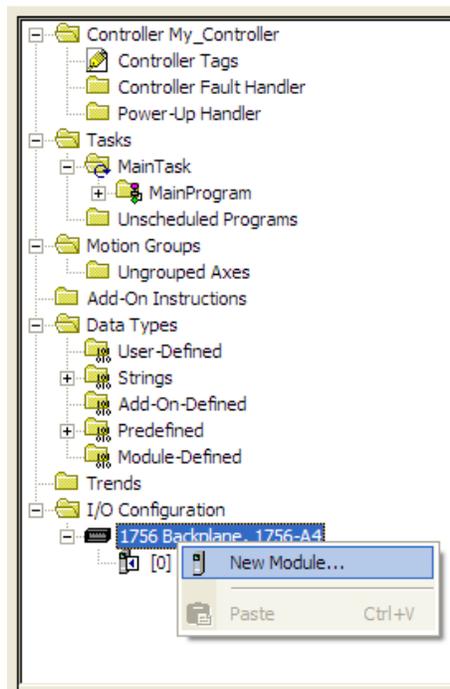
3.6 Special Functions and Controls (DFCM.CONTROL)

The DFCMUTIL object encapsulates and organizes variables and parameters used by the ladder logic and Add-On Instruction (AOI) to manage logic flow and optimize backplane block transfers based on user-selected ReadData and WriteData register counts.

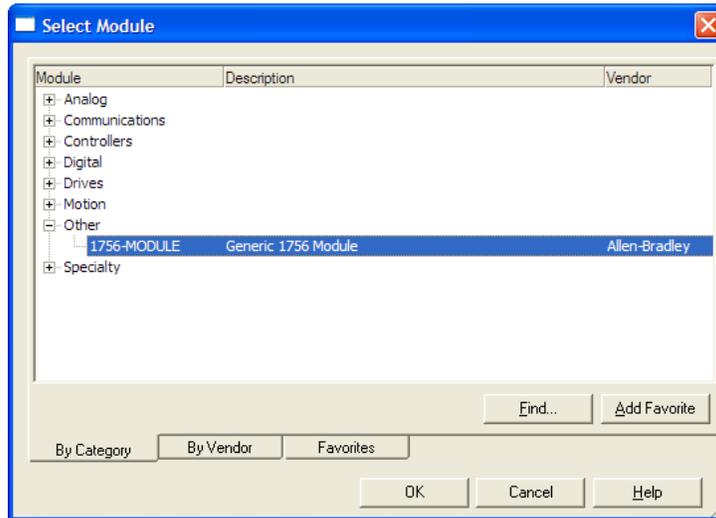
Name	Data Type	Description
BPLastRead	INT	Index of last read block
BPLastWrite	INT	Index of last write block
BlockIndex	INT	Computed block offset for data table
ReadDataSize	DINT	Size of Read Data Array
MaxReadBlock	DINT	Maximum Read Block
WriteDataSize	DINT	Size of Write Data Array
MaxWriteBlock	DINT	Maximum Write Block
RBTSremainder	INT	Contains remainder from Read Data array size divided by the block size
WBTSremainder	INT	Contains remainder from Write Data array size divided by the block size

3.7 Adding the Module to an Existing Project

- 1 Select the *I/O Configuration* folder in the *Controller Organization* window of RSLogix 5000, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW MODULE**.



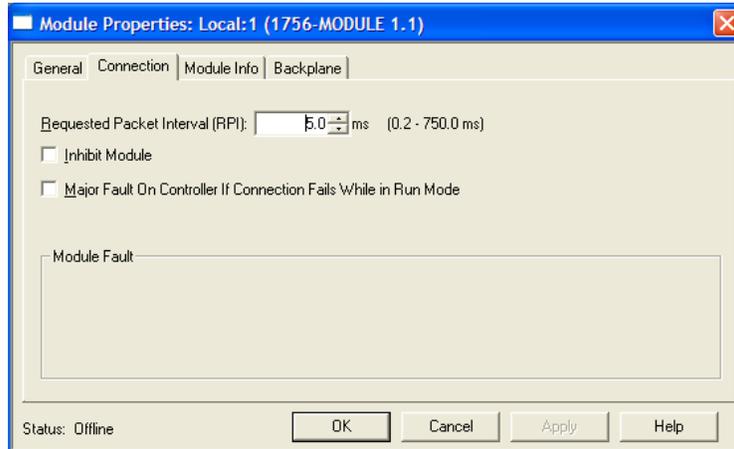
This action opens the *Select Module* dialog box:



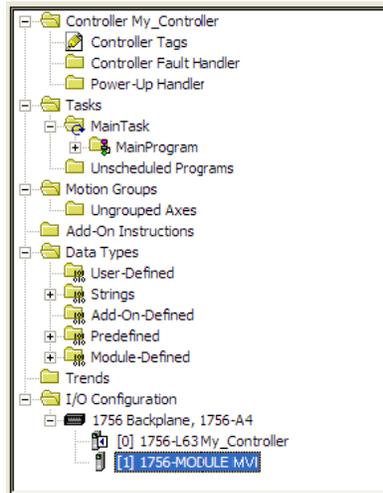
- 2 Select the **1756-MODULE (GENERIC 1756 MODULE)** from the list and click **OK**. This action opens the *New Module* dialog box.
- 3 Enter the *Name*, *Description* and *Slot* options for your application. You must select the *Comm Format* as **DATA - INT** in the dialog box, otherwise the module will not communicate. Click **OK** to continue.

Parameter	Value
Name	Enter a module identification string. Example: DFCM_2
Description	Enter a description for the module. Example: DF1 HALF/FULL DUPLEX MASTER/SLAVE SERIAL COMMUNICATION MODULE
Comm Format	Select DATA-INT .
Slot	Enter the slot number in the rack where the MVI56-DFCM module is located.
Input Assembly Instance	1
Input Size	250
Output Assembly Instance	2
Output Size	248
Configuration Assembly Instance	4
Configuration Size	0

- 4 Select the *Requested Packet Interval* value for scanning the I/O on the module. This value represents the minimum frequency that the module will handle scheduled events. This value should not be set to less than 1 millisecond. The default value is 5 milliseconds. Values between 1 and 10 milliseconds should work with most applications.



- 5 Save the module. Click **OK** to dismiss the dialog box. The *Controller Organization* window now displays the module's presence.



- 6 Copy the *User-Defined Data Types* from the sample program into your existing RSLogix 5000 project.
- 7 Copy the *Controller Tags* from the sample program into your project.
- 8 Copy the *Ladder Rungs* from the sample program into your project.

4 Diagnostics and Troubleshooting

In This Chapter

❖ LED Status Indicators.....	44
❖ Reading Status Data from the Module.....	49

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

- LED status indicators on the front of the module provide general information on the module's status.
- You can view status data contained in the module through the Configuration/Debug port or the Ethernet port, using the troubleshooting and diagnostic capabilities of *ProSoft Configuration Builder (PCB)*.
- You can transfer status data values from the module to processor memory and can monitor them in the processor manually or by customer-created logic. For details on Status Data values, see Error Status Table.

4.1 LED Status Indicators

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

LED	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 DF1 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 DF1 network on Port 2.
		Off	No data is being transferred on the port.
APP	Amber	On	The MVI56-DFCM module program has recognized a communication error on one of its ports.
		Off	The MVI56-DFCM is working normally.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module’s program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

During module configuration, the OK LED will be red and the APP and BP ACT LEDs will be on. If the LEDs are latched in this mode for a long period of time, look at the configuration error words in the configuration request block. The structure of the block is as shown below:

Offset	Description	Length
0	Reserved	1
1	9000	1
2	Module Configuration Errors	1
3	Port 1 Configuration Errors	1
4	Port 2 Configuration Errors	1
5 to 248	Spare	244
249	-2 or -3	1

The bits in each configuration word are shown below. The module configuration error word has the following definition:

Bit	Description	Value
0	Read block start value is greater than the database size.	0x0001
1	Read block start value is less than zero.	0x0002
2	Read block count value is less than zero.	0x0004
3	Read block count + start is greater than the database size.	0x0008
4	Write block start value is greater than the database size.	0x0010
5	Write block start value is less than zero.	0x0020
6	Write block count value is less than zero.	0x0040
7	Write block count + start is greater than the database size.	0x0080
8		0x0100
9		0x0200
10		0x0400
11		0x0800
12		0x1000
13		0x2000
14		0x4000
15		0x8000

The port configuration error words have the following definitions:

Bit	Description	Value
0	Type code is not valid. Enter a value from 0 (Master) to 1 (slave).	0x0001
1	Protocol parameter is not valid.	0x0002
2	Termination type parameter is not valid.	0x0004
3	Baud rate parameter is not valid.	0x0008
4	Parity parameter is not valid.	0x0010
5	Data bits parameter is not valid.	0x0020
6	Stop bits parameter is not valid.	0x0040
7	Command count parameter is not valid.	0x0080
8	Retry count parameter is not valid.	0x0100
9	Invalid Override File Map Count	0x0200
10	Spare	0x0400
11	Spare	0x0800
12	Spare	0x1000
13	Spare	0x2000
14	Spare	0x4000
15	Spare	0x8000

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

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

4.1.1 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 correct configuration data is being transferred to the module from the ControlLogix controller.

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

4.1.2 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 securely plugged into the slot that has been configured for the module in the I/O Configuration in RSLogix. Verify that the slot location in the rack has been configured correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. A problem could exist between the processor and any installed I/O module, not just the MVI56-DFCM. Verify that all modules in the rack are correctly configured.

Module Errors

Problem description	Steps to take
BP ACT LED (not present on MVI56E modules) remains OFF or blinks slowly MVI56 modules with scrolling LED display: <i><Backplane Status></i> condition reads ERR	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: <ul style="list-style-type: none"> ▪ The processor is in RUN or REM RUN mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write data block transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is properly configured in the processor I/O configuration and ladder logic.
OK LED remains RED	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert it, and then restore power to the rack.

4.2 Reading Status Data from the Module

The MVI56-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 (N45:0) to 7628 (N45:28) and at the location specified in the configuration. This data is transferred to the ControlLogix processor continuously with each read block. For a complete listing of the status data object, refer to the **Module Set Up** section.

4.2.1 The Configuration/Debug Menu

The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu. Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in *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 the *Prosoft Configuration Builder (PCB)* application.

Using the Configuration/Debug Port

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

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

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

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

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

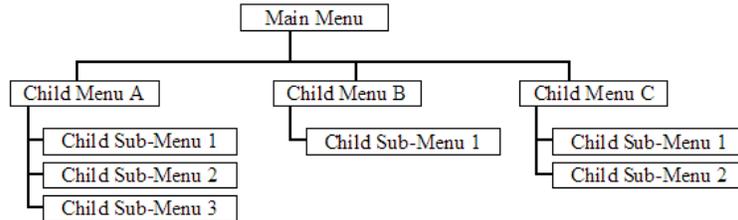
- 1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
- 2 Verify that RSLinx is not controlling the COM port. Refer to *Disabling the RSLinx Driver for the Com Port on the PC*.
- 3 Verify that your communication software is using the correct settings for baud rate, parity and handshaking.
- 4 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

Navigation

All of the submenus in *ProSoft Configuration Builder* for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a submenu 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 available commands.

Keystrokes

The keyboard commands on these menus are usually not case sensitive. You can enter most commands in lowercase or uppercase letters.

The menus use a few special characters (**[?]**, **[-]**, **[+]**, **[@]**) that must be entered exactly as shown. Some of these characters 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]** and **[/]**.

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

4.2.2 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.

```
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
U=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. Use these commands only if you fully understand their potential effects, or 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 (page 54) 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 number of seconds passed between the two readings.

Viewing Module Configuration

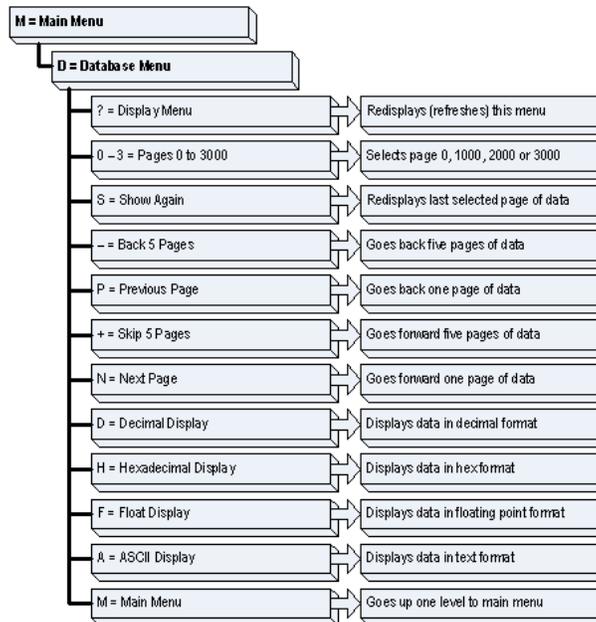
Press **[C]** to view the *Module Configuration* screen.

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

Opening the Database View Menu

Press **[D]** to open the *Database View* menu.

Use this menu command to view the current contents of the module's database. For more information about this submenu, see Database View Menu.



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 firmware version of the software (Software Revision Level) 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. Use these commands only if you fully understand their potential effects, or 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 causes 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 reboot.

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. Use these commands only if you fully understand their potential effects, or 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 memory to configure the module.

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.

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

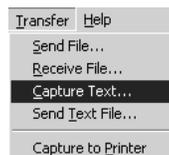
Port = 1, Format=HEX, Tick=10
```

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

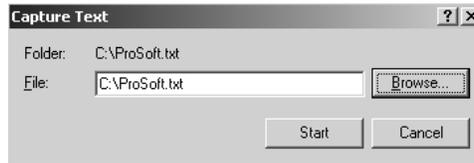
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

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 your 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_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][00][01]
[00][02]_TT_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_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_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_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_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.

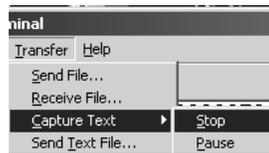
All characters shown in <> brackets are characters being sent out by the module.

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.

Returning to the Main Menu

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

4.2.4 Database View Menu

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

```
DB Menu Selected
DATABASE VIEW MENU
?=Display Menu
0-3=Display data page 0-3000
S=Show Again
-=Back 5 Pages
P=Previous Page
+=Skip 5 Pages
N=Next Page
D=Decimal Display
H=Hexadecimal Display
F=Float Display
A=ASCII Display
M=Main Menu
```

Viewing Register Pages

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

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

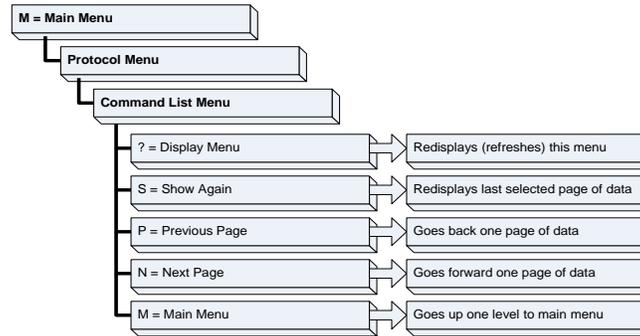
And so on. The total number of register pages available to view depends on your module's configuration.

Returning to the Main Menu

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

4.2.5 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.

Moving Back Through 5 Pages of Commands

Press **[-]** to display data for last 5 page commands.

Viewing the Previous Page of Commands

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

Moving Forward (Skipping) Through 5 Pages of Commands

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

Viewing the Next Page of Commands

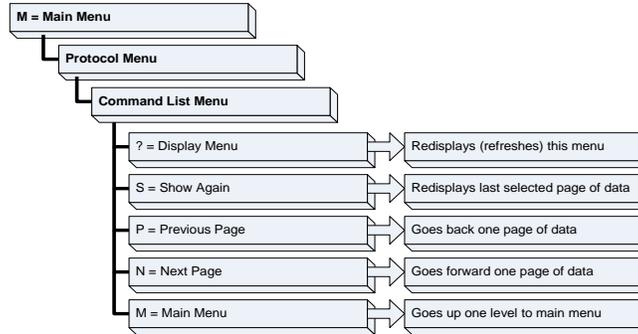
Press **[N]** to display the next page of commands.

Returning to the Main Menu

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

4.2.6 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.

5 Reference

In This Chapter

❖ Product Specifications	63
❖ Functional Overview	66
❖ DFCM Database Definition	86
❖ Error Codes	90
❖ Cable Connections	92

5.1 Product Specifications

The MVI56 DF1 Master/Slave Communication Module allows Rockwell Automation ControlLogix compatible processors to interface easily with DF1 protocol compatible devices and hosts. Devices commonly supporting the protocol include Rockwell Automation PLCs and power monitoring equipment, as well as several other third party devices in the marketplace.

The MVI56-DFCM module has two Application 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 ControlLogix 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 - 1756 backplane compatible
- Local or remote rack
- 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.
- Configuration data obtained through user-defined ladder. Sample ladder file included

5.1.2 Hardware Specifications

Specification	Description
Backplane Current Load	800 mA @ 5 VDC 3 mA @ 24 VDC
Operating Temperature	0°C to 60°C (32°F to 140°F)
Storage Temperature	-40°C to 85°C (-40°F to 185°F)
Shock	30 g operational 50 g non-operational Vibration: 5 g from 10 to 150 Hz
Relative Humidity	5% to 95% (without condensation)
LED Indicators	Module Status Backplane Transfer Status Application Status Serial Activity
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only
Application ports (PRT1 & PRT2)	
Full hardware handshaking control, providing radio, modem and multi-drop support	
Software configurable communication parameters	Baud rate: 110 to 115,200 baud, depending on protocol RS-232, 485 and 422 Parity: none, odd or even Data bits: 5, 6, 7, or 8 Stop bits: 1 or 2 RTS on/off delay: 0 to 65535 milliseconds
App Ports (P1, P2) (Serial modules)	RJ45 (DB-9M 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 5000 registers to/from the control processor
- Up to 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 processor 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 Technology 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

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding module operation.

Module Power Up

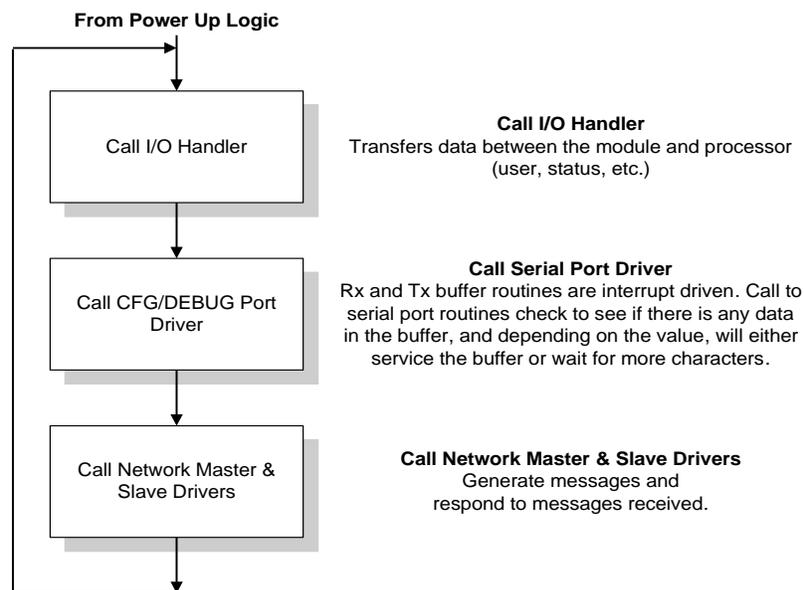
On power up the module begins performing the following logical functions:

- 1 Initialize hardware components
 - Initialize ControlLogix backplane driver
 - Test and Clear all RAM
 - Initialize the serial communication ports
- 2 Wait for Module Configuration from ControlLogix processor
- 3 Initialize Module Register space
- 4 Enable Slave Driver on selected ports
- 5 Enable Master Driver on selected ports

After the module has received the Module Configuration Block from the processor, 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:



ControlLogix Processor Not in Run

Whenever the module detects that the processor has gone out of the Run mode (that is, Fault or PGM), the DF1 ports can be shut down as prescribed in the user configuration. When the processor is returned to a running state, the module will resume communications on the network.

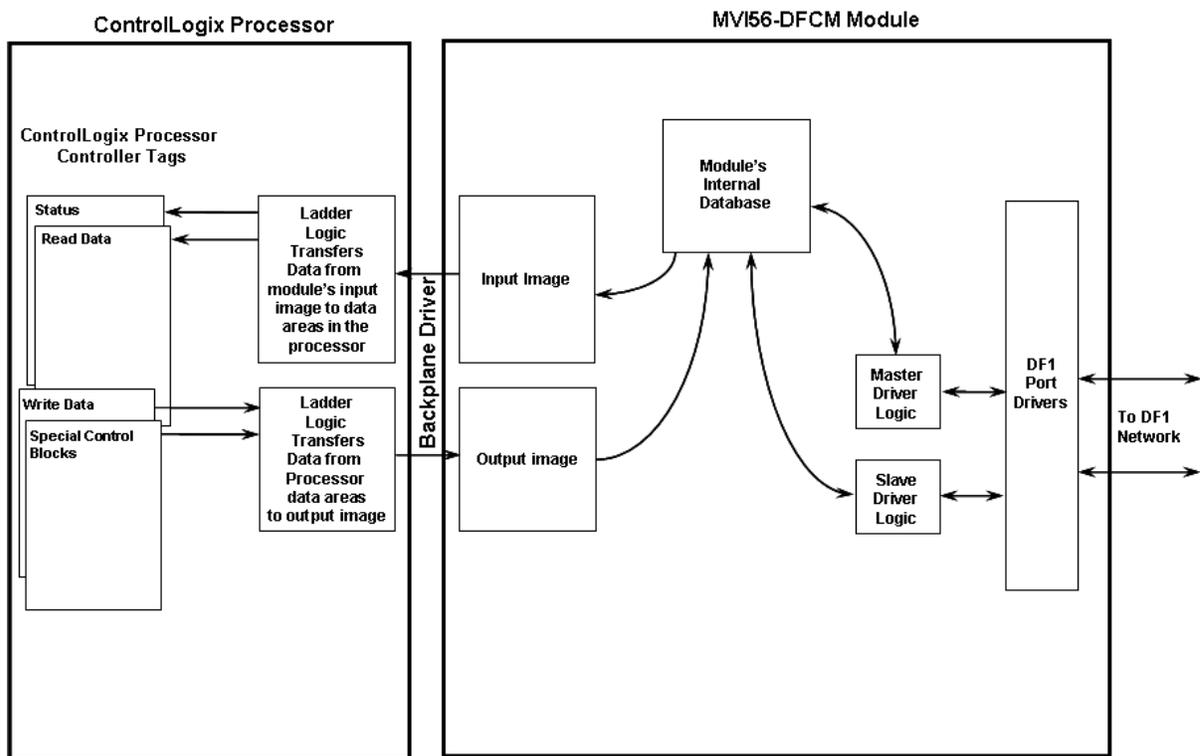
Backplane Data Transfer

The MVI56-DFCM module communicates directly over the ControlLogix backplane. Data is paged between the module and the ControlLogix processor across the backplane using the module's input and output images. The update frequency of the images is determined by the scheduled scan rate defined by the user for the module and the communication load on the module. Typical updates are in the range of 2 to 10 milliseconds.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the Controller Tags in the processor by the ladder logic. The input image for the module is set to 250 words. This large data area permits fast throughput of data between the module and the processor.

The processor inserts data to the module's output image to transfer to the module. The module's program extracts the data and places it in the module's internal database. The output image for the module is set to 248 words. This large data area permits fast throughput of data from the processor to the module.

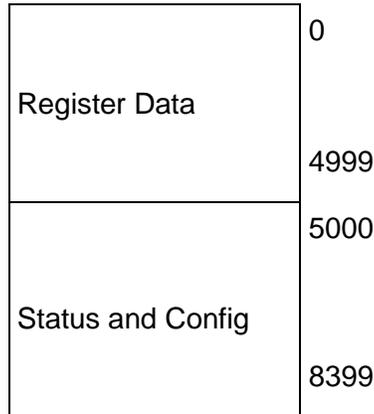
The following illustration shows the data transfer method used to move data between the ControlLogix processor, the MVI56-DFCM module and the DF1 network.



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 ControlLogix processor to interface the input and output image data with data defined in the Controller Tags. All data used by the module is stored in its internal database. The following illustration shows the layout of the database:

Module's Internal Database Structure

5000 registers for user data



3400 words of configuration and status data

Data contained in this database is paged through the input and output images by coordination of the ControlLogix ladder logic and the MVI56-DFCM module's program. Up to 248 words of data can be transferred from the module to the processor at a time. Up to 247 words of data can be transferred from the processor to the module. 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
-9000	Configuration data (sent from module)
-7100	Port 2 Override File Maps (sent from module)
-7000	Port 1 Override File Maps (sent from module)
-6100 to -6104	Port 2 commands (sent from module)
-6000 to -6004	Port 1 commands (sent from module)
-1	Null block
0	Null 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 3003	Port 1 slave status
3100 to 3101	Port 2 slave polling control
3102 to 3103	Port 2 slave status
5000 to 5006	Port 1 command control
5100 to 5106	Port 2 command control
6000 to 6004	Port 1 commands (sent to module)
6100 to 6104	Port 2 commands (sent to module)
7000	Port 1 Override File Maps
7100	Port 2 Override File Maps
9000	Configuration data (sent to module)
9972	Set module time using processor time
9973	Get module time to set processor time
9998	Warm-boot control block
9999	Cold-boot control block

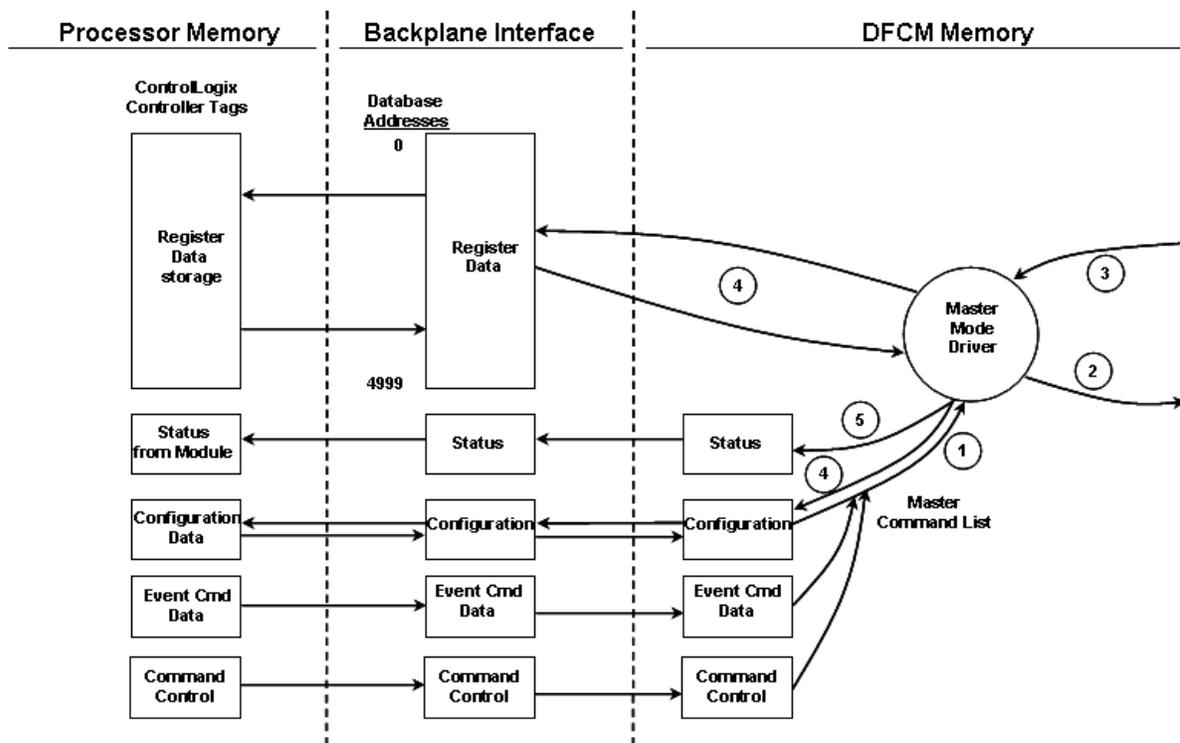
Each image has a defined structure depending on the data content and the function of the data transfer.

5.2.2 Data Flow between MVI56-DFCM Module and ControlLogix Processor

The following topics describe the flow of data between the two pieces of hardware (ControlLogix processor and MVI56-DFCM module) and other nodes on the DF1 network under the module's different operating modes. Each port on the module is configured to emulate a DF1 Master device or a DF1 slave device. The operation of each port depends on this configuration. The sections below discuss the operation of each mode.

Master Driver Mode

In Master Mode, the MVI56-DFCM module is responsible for issuing read or write commands to slave devices on the DF1 network. These commands are user configured in the module via the Master Command List received from the ControlLogix processor or issued directly from the ControlLogix 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 ControlLogix processor. 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 DF1 network (Refer to the Configuring the Module section).
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 ControlLogix processor for each command in the Master Command List.

Refer to the **Configuring the Module** section for a complete description of the parameters required to define the virtual DF1 Master port. Refer to the **DF1 Command Set** documentation for a complete discussion of 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
Func Code	Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	Node Address	Func 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 150	Code	Register	1/10 Sec	Count	Code	Node	150	File String	N/A	N/A	N/A
FC 151	Code	Register	1/10 Sec	Count	Code	Node	151	File String	N/A	N/A	N/A
FC 152	Code	Register	1/10 Sec	Count	0	Node	152	File String	N/A	N/A	N/A
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

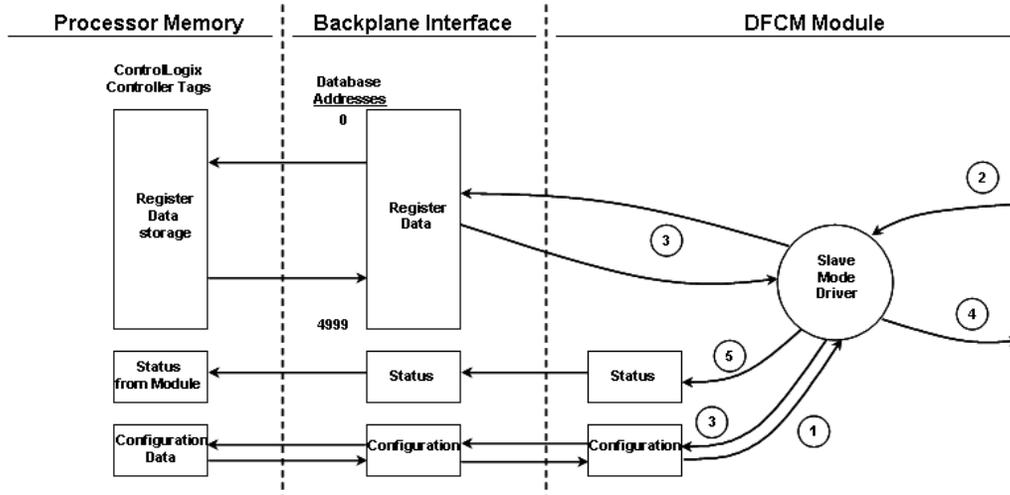
Node Address = Destination Address for Message

If the DF1 Master port is configured to support the DF1 half-duplex protocol, the Master port can be used to route messages between slaves. Peer-to-peer communication is accomplished by the Master constantly polling all the slaves on the network and relaying the messages received. The slaves must contain ladder logic with MSG commands to generate and accept messages. This routing can be used in conjunction with the normal command processing discussed above.

If the slave node to be polled is not included in the command list, a special command is required in the command list. Enter a 999 in the **ENABLE CODE** and the slave's node address in the **NODE ADDRESS** field of the command. No other parameters are required for a device poll message. This command will force the Master port to issue an inquiry request to the slave device without first issuing a command. Any messages held in the slave's message queue will be sent to the Master and the Master will route the messages that do not contain the Master's station code.

Slave Driver Mode

The Slave Driver Mode allows the MVI56-DFCM module to respond to data read and write commands issued by a Master on the DF1 network. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The DF1 slave port driver receives the configuration information from the ControlLogix processor. 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 fixed length of 200-word registers.
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 Configuring the Module section for a complete list of the parameters that must be defined for a slave port.

5.2.3 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. Refer to **Configuring the Module** for a description of the data objects used with the blocks and the ladder logic required. The following topics describe the function and structure of each block.

Read Block

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

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2 to 201	Read Data	200
202 to 234	Module Status Data	33
235 to 248	Spare	14
249	Read Block ID	1

The Read Block ID is an index value used to determine the location of where the data will be placed in the ControlLogix processor controller tag array of module read data. Each transfer can move up to 200 words (block offsets 2 to 201) of data. In addition to moving user data, the block also contains status data for the module. This last set of data is transferred with each new block of data and is used for high-speed data movement.

The Write Block ID associated with the block requests data from the ControlLogix processor. Under normal program operation, the module sequentially sends read blocks and requests write blocks. For example, if the application uses three read and two write blocks, 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 DF1 network or operator control through the module's Configuration/Debug port.

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.

DFCMSTATUS Data Structure

Offset	Content	Description
202	Program Scan Count	This value is incremented each time a complete program cycle occurs in the module.
203 to 204	Product Code	These two registers contain the product code of "DFCM"
205 to 206	Product Version	These two registers contain the product version for the current running software.
207 to 208	Operating System	These two registers contain the month and year values for the program operating system.
209 to 210	Run Number	These two registers contain the run number value for the currently running software.
211	Port 1 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
212	Port 1 Command List Response	This field contains the number of slave response messages received on the port.
213	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.
214	Port 1 Requests	This field contains the total number of messages sent out of the port.
215	Port 1 Responses	This field contains the total number of messages received on the port.
216	Port 1 Errors Sent	This field contains the total number of message errors sent out of the port.
217	Port 1 Errors Received	This field contains the total number of message errors received on the port.
218	Port 2 Command List Requests	This field contains the number of requests made from this port to slave devices on the network.
219	Port 2 Command List Response	This field contains the number of slave response messages received on the port.
220	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.
221	Port 2 Requests	This field contains the total number of messages sent out the port.
222	Port 2 Responses	This field contains the total number of messages received on the port.
223	Port 2 Errors Sent	This field contains the total number of message errors sent out of the port.
224	Port 2 Errors Received	This field contains the total number of message errors received on the port.
225	Read Block Count	This field contains the total number of read blocks transferred from the module to the processor.
226	Write Block Count	This field contains the total number of write blocks transferred from the processor to the module.

Offset	Content	Description
227	Parse Block Count	This field contains the total number of blocks successfully parsed that were received from the processor.
228	Command Event Block Count	This field contains the total number of command event blocks received from the processor.
229	Command Block Count	This field contains the total number of command blocks received from the processor.
230	Error Block Count	This field contains the total number of block errors recognized by the module.
231	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. When the port is configured as a master and the value is negative one (-1), none of the configured commands have had an error yet.
232	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. If the value shown is negative one (-1), no error has yet happened.
233	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.
234	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.

Write Block

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

Offset	Description	Length
0	Write Block ID	1
1 to 200	Write Data	200
201 to 247	Spare	47

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.

5.2.4 Configuration Data Transfer Blocks (-9000, -6000 to -6004, -6100 to -6104, 6000 to 6004, 6100- to 6104, and 9000)

When the module performs a restart operation, it will request configuration information from the ControlLogix processor. This data is transferred to the module in specially formatted write blocks (output image). The module will poll for each block by setting the required write block number in a read block (input image). Refer to Configuring the Module for a description of the data objects used with the blocks and the ladder logic required.

Write Configuration Blocks (-9000, -6000 to -6004, and -6100 to -6104)

This block is sent from the ControlLogix processor, and causes the module to write its current configuration back to the processor. This function is used when the module's configuration has been altered remotely using database write operations. The write block contains a value of -9000 in the first word. The module will respond with blocks containing the module configuration data. Ladder logic must handle the receipt of these blocks. The blocks transferred from the module are as follows:

Block -9000, General Configuration Data:

Offset	Description	Length
0	Reserved	1
1	-9000	1
2 to 7	Backplane Set Up	6
8 to 33	Port 1 Configuration	26
34 to 59	Port 2 Configuration	26
60 to 248	Spare	189
249	-9000	1

Blocks -6000 to -6004 and -6100 to 6104, Master Command List Data for ports 1 and 2, respectively:

Offset	Description	Length
0	Reserved	1
1	-6000 to -6004 and -6100 to -6104	1
2 to 13	Command Definition	12
14 to 25	Command Definition	12
26 to 37	Command Definition	12
38 to 49	Command Definition	12
50 to 61	Command Definition	12
62 to 73	Command Definition	12
74 to 85	Command Definition	12
86 to 97	Command Definition	12
98 to 109	Command Definition	12
110 to 121	Command Definition	12
122 to 133	Command Definition	12
134 to 145	Command Definition	12
146 to 157	Command Definition	12
158 to 169	Command Definition	12
170 to 181	Command Definition	12
182 to 193	Command Definition	12

Offset	Description	Length
194 to 205	Command Definition	12
206 to 217	Command Definition	12
218 to 229	Command Definition	12
230 to 241	Command Definition	12
242 to 248	Spare	7
249	-6000 to -6004 and -6100 to -6104	1

Each of these blocks must be handled by the ladder logic for proper module operation.

Module Configuration Data Block (9000)

This block sends general configuration information from the processor to the module. The data is transferred in a block with an identification code of 9000.

Module Configuration (Write Block)

Offset	Description	Length
0	Configuration Block ID number (value 9000)	1
1 to 6	Backplane Set Up	6
7 to 32	Port 1 Configuration	26
33 to 58	Port 2 Configuration	26
59 to 247	Spare	189

The read block used to request the configuration has the following structure:

Offset	Description	Length
0	Reserved	1
1	Configuration Block ID number (value 9000)	1
2	Module Configuration Errors	1
3	Port 1 Configuration Errors	1
4	Port 2 Configuration Errors	1
5 to 248	Spare	244
249	-2 or -3	1

If there are any errors in the configuration, the bit associated with the error will be set in one of the three configuration error words. The error must be corrected before the module starts operating.

Master Command List Data Blocks (6000 to 6004 and 6100 to 6104)

Each port on the module can be configured as a DF1 Master device containing its own list of one hundred commands. The commands are sent from the processor using Write Block ID's 6000 to 6004 for DF1 Port 1 and 6100 to 6104 for DF1 Port 2. The module will sequentially request each block from the processor. Ladder logic handles the data transfers. The structure of each block is shown below:

Offset	Description	Length
0	6000 to 6004 and 6100 to 6104	1
1 to 12	Command Definition	12
13 to 24	Command Definition	12
25 to 36	Command Definition	12
37 to 48	Command Definition	12
49 to 60	Command Definition	12
61 to 72	Command Definition	12
73 to 84	Command Definition	12
85 to 96	Command Definition	12
97 to 108	Command Definition	12
109 to 120	Command Definition	12
121 to 132	Command Definition	12
133 to 144	Command Definition	12
145 to 156	Command Definition	12
157 to 168	Command Definition	12
169 to 180	Command Definition	12
181 to 192	Command Definition	12
193 to 204	Command Definition	12
205 to 216	Command Definition	12
217 to 228	Command Definition	12
229 to 240	Command Definition	12
241 to 247	Spare	7

5.2.5 Event Command Blocks (1000 or 2000)

Event Command Blocks send DF1 commands directly from the ladder logic to one of the Master ports. The Event Command Write Block format is:

Offset	Description	Length
0	1000 (Send from Port 1) or 2000 (Send from Port 2)	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 247	Spare	242

The block number defines the DF1 port to be considered. Block 1000 commands are directed to Port 1, and block 2000 commands are directed to Port 2. Use the parameters passed with the block to 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 DF1 network to consider.

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 command block with a Read Block. The following table describes the format of this block.

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2	0=Fail, 1=Success	1
3 to 248	Spare	246
249	1000 or 2000	1

Word Two (2) of the block can be used by the ladder logic to determine whether or not the command was successfully added to the module's command queue. The Event Command attempt will fail if the command queue for the port is full (100 commands for each queue) or the command requested is invalid.

5.2.6 Slave Status Blocks (3002, 3003, 3102, and 3103)

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:

Status Value	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 (ErrorDelayCnt value in the DFCM.config.PortX object for each port). 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. This will enable polling of the slave.

In order to read the slave status table, ladder logic must be written. 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 for first 128 slave status values for Port 1
3003	Request for last 128 slave status values for Port 1
3102	Request for first 128 slave status values for Port 2
3103	Request for last 128 slave status values for Port 2

The format of these blocks is as shown below:

Offset	Description	Length
0	3002 to 3003 or 3102 to 3103	1
1 to 247	Spare	246

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

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2 to 129	Slave Poll Status Data	128
130 to 248	Spare	119
249	3002 to 3003 or 3102 to 3103	1

Ladder logic can be written 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:

Offset	Description	Length
0	3000 or 3100	1
1	Number of Slaves in Block	1
2 to 201	Slave indexes	200
202 to 247	Spare	46

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:

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2	Number of slaves processed	1
3 to 248	Spare	246
249	3000 or 3100	1

Ladder logic can be written 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:

Offset	Description	Length
0	3001 or 3101	1
1	Number of Slaves in Block	1
2 to 201	Slave indexes	200
202 to 247	Spare	46

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:

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2	Number of slaves processed	1
3 to 248	Spare	246
249	3001 or 3101	1

5.2.7 Command Control Blocks (5000 to 5006 or 5100 to 5106)

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 format of the block is displayed below:

Offset	Description	Length
0	5001 to 5006 or 5101 to 5106	1
1	Command index	1
2	Command index	1
3	Command index	1
4	Command index	1
5	Command index	1
6	Command index	1
7 to 247	Spare	241

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 for 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 format of the block is displayed below:

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2	Number of commands added to command queue	1
3 to 248	Spare	246
249	5000 to 5006 or 5100 to 5106	1

5.2.8 Override File Map Configuration Blocks (7000 and 7100, 50 entries per port)

The Override File Mapping feature allows users to assign PCCC-type data table memory addresses (like N10, F8, B3, etc.) to specific areas of the module's user data memory. This allows older DF1 applications and drivers to access the module's memory as if it were a PLC5 or SLC processor. Override File Mapping also allows users to change the module's default PCCC-type data table addressing by assigning specific data file numbers to specific areas of module memory.

Offset	Description	Length
0	7000 or 7100	1
1 to 4	Map #1 Definition	4
5 to 8	Map #2 Definition	4
9 to 12	Map #3 Definition	4
...	and so on	
197 to 200	Map #50 Definition	4
201 to 247	Spare	4

5.2.9 Module Time Blocks (9972 and 9973)

Set Module Time Using Processor Time Block (9972)

If the module receives a Block ID of 9972 from the processor, it will set the time and date in the module to what is contained in the block. This is the Write block structure:

Word Offset in Block	Data Field(s)	Description
0	9972	Write Block ID (for the Set Time function)
1	Year	This field contains the four-digit value for the new time value.
2	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
3	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
4	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
5	Minute	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
6	Seconds	This field contains the second value for the new time. Valid entry for this field is in the range of 0 to 59.
7 to 247	Not Used	Reserved

The module will respond to a valid 9972 Write block by returning the following Read Block:

Word Offset in Block	Data Field(s)	Description
0	Not used	Reserved
1	Block Number	Write Block ID
2 to 248	Spare	Reserved
249	9972	Read Block ID (Set Time acknowledgment)

Get Module Time (Block 9973)

Block Identification Code 9973 is used to requests the module's date and time. You can use this data to set the ControlLogix clock to match the module's data and time.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Write Block ID	This field contains the value of 9973 to request the module's current date and time.
1 to 247	Not Used	Reserved

The module responds to a valid block 9973 request with a block containing the requested date and time.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Write Block ID	This is the next Write Block requested by the module.
2	Year	This field contains the four-digit year for the new time value.
3	Month	This field contains the month value for the new time. Valid entries for this field are in the range of 1 to 12.
4	Day	This field contains the day value for the new time. Valid entries for this field are in the range of 1 to 31.
5	Hour	This field contains the hour value for the new time. Valid entries for this field are in the range of 0 to 23.
6	Minute	This field contains the minute value for the new time. Valid entries for this field are in the range of 0 to 59.
7	Seconds	This field contains the second value for the new time. Valid entries for this field are in the range of 0 to 59.
8	Milliseconds	This field contains the millisecond value for the new time. Valid entries for this field are in the range of 0 to 999.
9 to 248	Not Used	Not Used
249	Read Block ID	This field contains the block identification code of 9973 to let the ladder logic know this block contains the module's date and time information.

5.2.10 Block 9999: Cold Boot

This Block ID is sent from the ControlLogix processor to the module (output image) when the user wants to perform a cold boot (hardware reset) operation. This Block ID can be sent to the module when a problem is detected by ladder logic that may be corrected by a hardware reset.

Offset	Description	Length
0	9999	1
1 to 247	Spare	247

5.2.11 Warm Boot Block (9998) MVI56-DFCM

This Block ID is sent from the ControlLogix processor to the module when the user wants to perform a warm-boot (software reset) operation. Ladder logic can be created to send this block to the module any time configuration data modifications are made in the controller tags data area. This block will cause the module to read the new configuration information and to restart.

Offset	Description	Length
0	9998	1
1 to 247	Spare	247

5.3 DFCM Database Definition

This section contains a listing of the internal database of the MVI56-DFCM module. This information can be used to interface other devices to the data contained in 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 7699	N45:0	N45:199	Misc. Status Data	200
7800 to 7999	N46:0	N46:199	Command Control	200
8000 to 8199	N47:0	N47:199	Port 1 Override File Maps	200
8200 to 8399	N48:0	N48:199	Port 2 Override File Maps	200
8400 to 9999	N49:0	N56:199	Reserved	1600

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).

Detailed definition of the miscellaneous status data area can be found in Status Data Definition (page 75)

Definition of the configuration data areas can be found in the data definition section of this document and in Configuration Data Definition

DFCM Command Control contains a discussion of the command control section of the database.

Col #	Module Information Data ←					→ Device Information Data					
	1	2	3	4	5	6	7	8	9	10	11
Func Code	Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	Node Address	Func 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 150	Code	Register	1/10 Sec	Count	Code	Node	150	File String	N/A	N/A	N/A
FC 151	Code	Register	1/10 Sec	Count	Code	Node	151	File String	N/A	N/A	N/A
FC 152	Code	Register	1/10 Sec	Count	0	Node	152	File String	N/A	N/A	N/A
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

5.3.1 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

5.3.2 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

5.3.3 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
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 module's 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. 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.

5.4 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.

If a 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:

5.4.1 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

5.4.2 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
-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)

5.4.3 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

5.4.4 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
-8	0xFFF8	DLE-ENQ received
-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
-200	0xFF38	DLE-NAK received after request

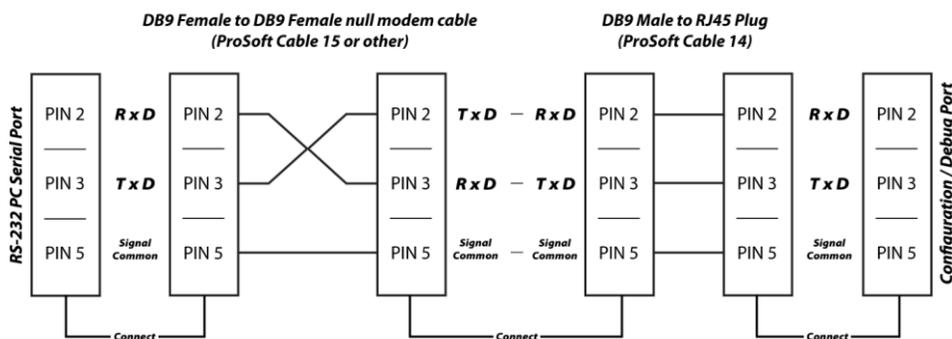
5.5 Cable Connections

The application ports on the MVI56-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.

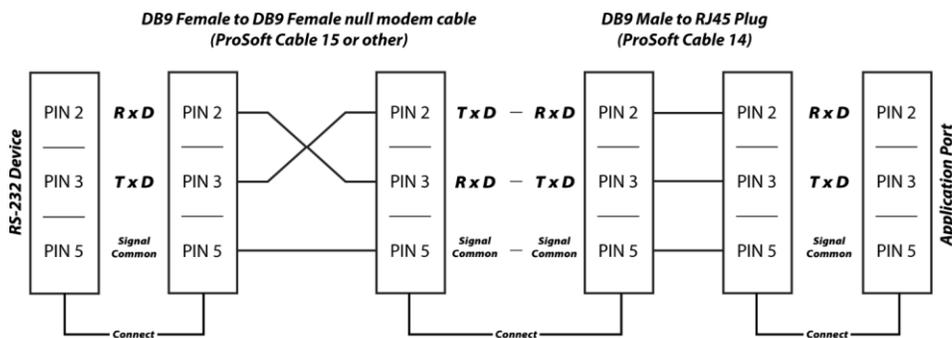
5.5.1 RS-232 Configuration/Debug Port

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 pinout for communications on this port is shown in the following diagram.



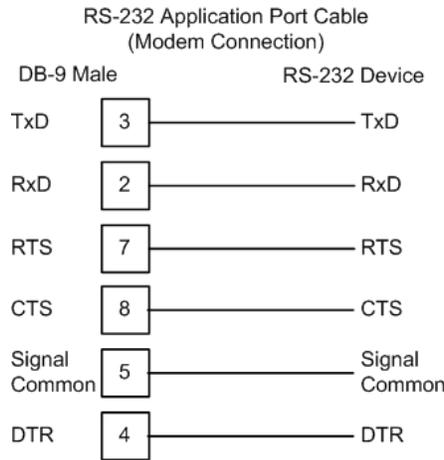
5.5.2 RS-232 Application Port(s)

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, here are the cable pinouts to connect to the port.



RS-232: Modem Connection (Hardware Handshaking Required)

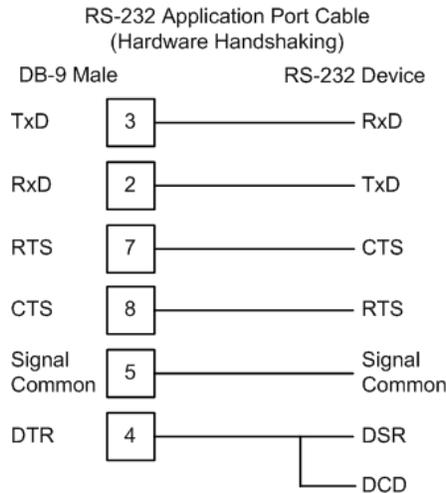
This type of connection is required between the module and a modem or other communication device.



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

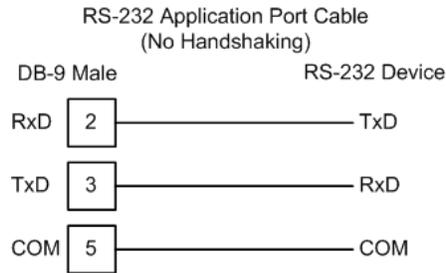
RS-232: Null Modem Connection (Hardware Handshaking)

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

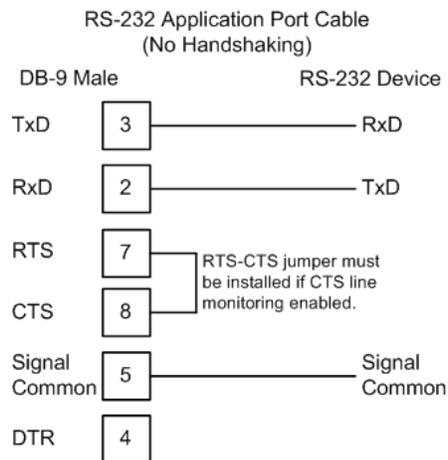


RS-232: Null Modem Connection (No Hardware Handshaking)

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

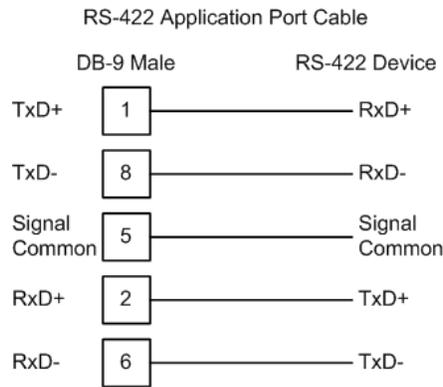


Note: For most null modem connections where hardware handshaking is not required, the *Use CTS Line* parameter should be set to **N** and no jumper will be required between Pins 7 (RTS) and 8 (CTS) on the connector. If the port is configured with the *Use CTS Line* set to **Y**, then a jumper is required between the RTS and the CTS lines on the port connection.



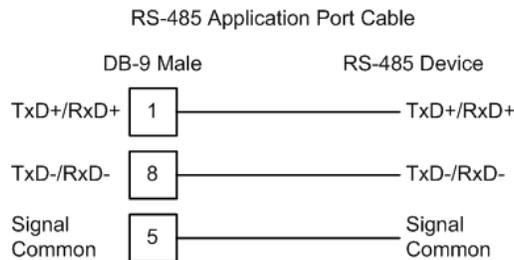
5.5.3 RS-422

The RS-422 interface requires a single four or five wire cable. The Common connection is optional, depending on the RS-422 network devices used. The cable required for this interface is shown below:



5.5.4 RS-485 Application Port(s)

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

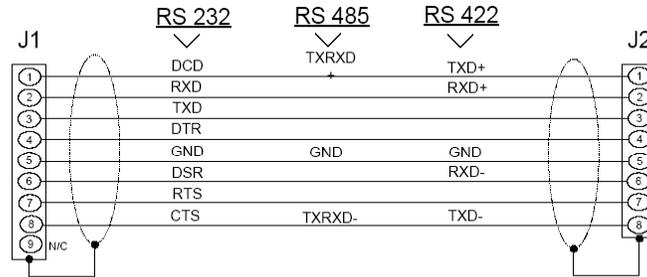
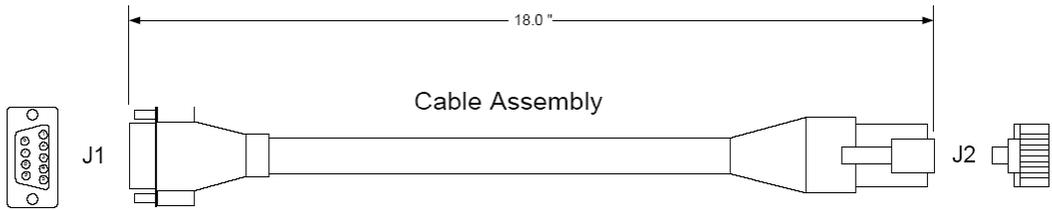


Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In these cases, installing a 120-ohm terminating resistor between pins 1 and 8 on the module connector end of the RS-485 line may improve communication quality.

RS-485 and RS-422 Tip

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

5.5.5 DB9 to RJ45 Adaptor (Cable 14)



Wiring Diagram

6 Support, Service & Warranty

6.1 Contacting Technical Support

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

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

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

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

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

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

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

6.2 Warranty Information

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

www.prosoft-technology/legal