Where Automation Connects.

MVI56E-SIE

ControlLogix Platform

Siemens Industrial Ethernet
Client Communication Module

February 7, 2023

USER MANUAL

Your Feedback Please

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

How to Contact Us

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

MVI56E-SIE User Manual
For Public Use.

February 7, 2023

Content Disclaimer

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

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

For professional users in the European Union

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

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

Agency Approvals & Certifications

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

Open Source Information

Open Source Software used in the product

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

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

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

Warranty regarding further use of the Open Source Software

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

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

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

Important Safety Information

North America Warnings

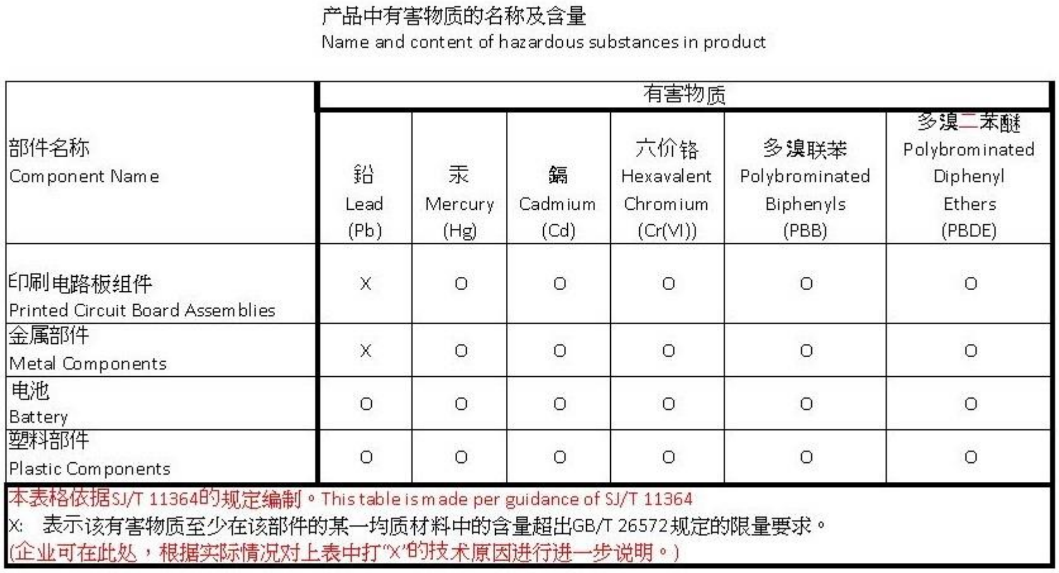
A Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.
B Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or rewiring modules.
Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be nonhazardous.
C Suitable for use in Class I, Division 2 Groups A, B, C, and D, T5 Hazardous Locations or Non-Hazardous Locations.
D Class 2 Power

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.

China RoHS Hazardous Material Declaration Table



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

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

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

Electrical Ratings

- 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 Hz to 150 Hz
- Relative Humidity 5% to 95% (without condensation)
- All phase conductor sizes must be at least 1.3 mm^2 and all earth ground conductors must be at least 4mm^2.

Contents

1 Start Here

To get the most benefit from this User Manual, the following skills will be needed:

- **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 Siemens Industrial Ethernet and ControlLogix devices to a power source and to the MVI56E-SIE module's application port(s)

1.1 Features

- **ProSoft Configuration Builder (PCB):** Windows-based software for diagnostics, connecting via the module's Ethernet port or CIPconnect®, to upload/download module configuration information and access troubleshooting features and functions.
- **ProSoft Discovery Service (PDS):** Utility software to find and display a list of MVI56E modules on the network and to temporarily change an IP address to connect with a module's web page.
- **CIPconnect-enabled:** Allows PC-to-module configuration and diagnostics from the Ethernet network through a ControlLogix 1756-ENBT EtherNet/IP™ module.
- **Personality Module:** An industrial compact flash memory card storing the module's complete configuration and Ethernet settings, allowing quick and easy replacement.
- **LED Scrolling Diagnostic Display:** 4-character, alphanumeric display, providing messages for status and alarm data, and for processor and network communication status.

1.2 System Requirements

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

- Rockwell Automation ControlLogix® processor (firmware version 16 or higher), with compatible power supply, and one free slot in the rack for the MVI56E-SIE module. The module requires 800 mA of available 5 Vdc power
- Rockwell Automation RSLogix 5000 programming software
  - Version 16 or higher required for Add-On Instruction
- Rockwell Automation RSLinx® communication software version 2.51 or higher
- ProSoft Configuration Builder (PCB) (included)
- ProSoft Discovery Service (PDS) (included in PCB)
- 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 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)

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

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

1.3 Package Contents

The following components are included with your MVI56E-SIE module, and are all required for installation and configuration.

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

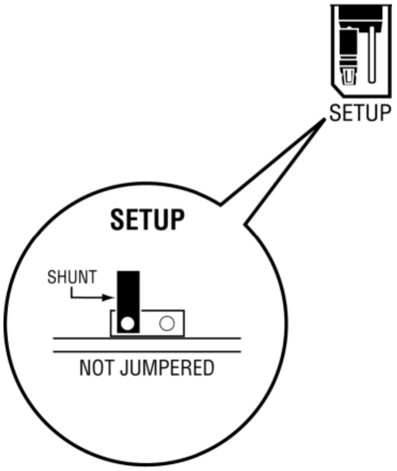
Qty.	Part Name	Part Number	Part Description
1	MVI56E-SIE Module	MVI56E-SIE	Siemens Industrial Ethernet Client Communication Module

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

1.4 Setting Jumpers

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

The following illustration shows the jumper configuration with the Setup Jumper OFF.



Note: If you are installing the module in a remote rack, you may prefer to leave the Setup pins jumpered. That way, you can update the module’s firmware without requiring physical access to the module.

Security considerations:

Leaving the Setup pin jumpered leaves the module open to unexpected firmware updates.

You should consider segmenting the data flow for security reasons. Per IEC 62443-1-1, you should align with IEC 62443 and implement segmentation of the control system. Relevant capabilities are firewalls, unidirectional communication, DMZ. Oil and Gas customers should also see DNVGL-RP-G108 for guidance on partitioning.

You should practice security by design, per IEC 62443-4-1, including layers of security and detection. The module relies on overall network security design, as it is only one component of what should be a defined zone or subnet.

1.5 Installing the Module in the Rack

If the ControlLogix processor and power supply have not been installed and configured, please do so before installing the MVI56E-SIE module. Refer to the Rockwell Automation product documentation for installation instructions.

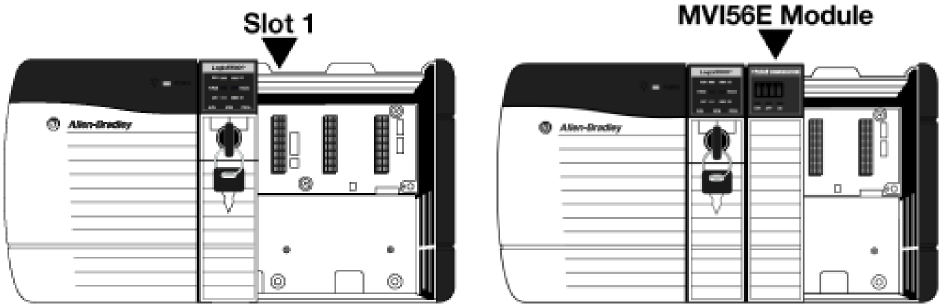
> **Warning:** All safety instructions must be followed when installing this or any other electronic devices. Failure to follow safety procedures could result in damage to hardware or data, or even serious injury or death to personnel. Refer to the documentation for each device that will be connected to verify that suitable safety procedures are in place before installing or servicing the device.

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

ControlLogix system components can be installed or removed while chassis power is applied and the system is operating. However, please note the following warning.

> **Warning:** When the module is inserted or removed while backplane power is on, an electrical arc can occur. An electrical arc can cause personal injury or property damage by sending an erroneous signal to the system's actuators. This can cause unintended machine motion or loss of process control. Electrical arcs may also cause an explosion when they happen in a hazardous environment. Verify that power is removed or the area is non-hazardous before proceeding.
>
> Repeated electrical arcing causes excessive wear to contacts on both the module and its mating connector. Worn contacts may create electrical resistance that can affect module operation.

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



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

1.6 Importing the Sample Add-On Instruction

Note: This section only applies if the processor is using RSLogix 5000 version 16 or higher. If an earlier version is installed, please contact ProSoft Technology for more information.

Before You Begin

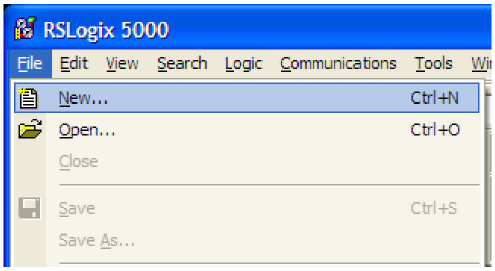
Two Add-On Instructions are provided for the MVI56E-SIE module. The first is required for setting up the module; the second is optional.

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

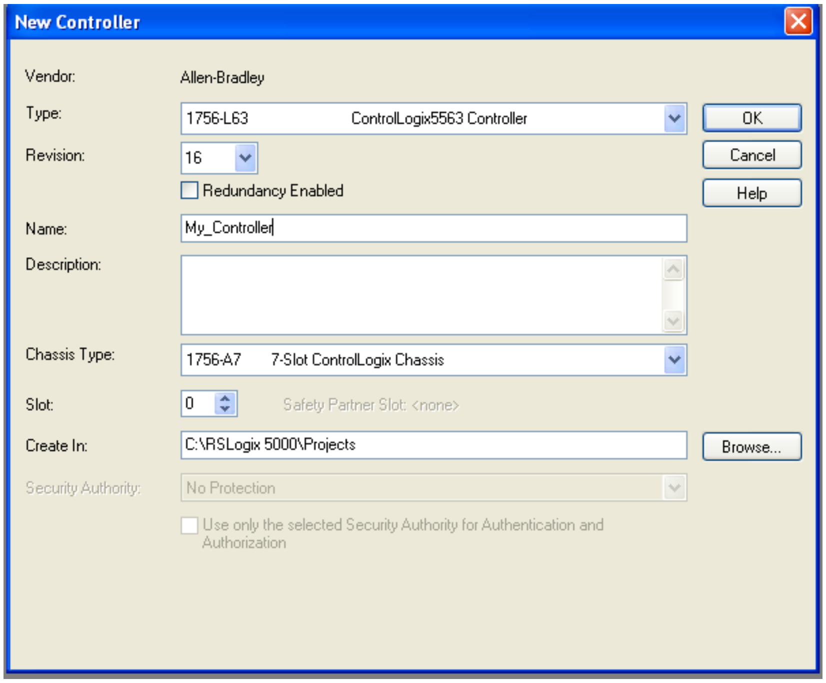
File Name	Description
MVI56ESIE_AddOn_Rung_vXXX.L5X	L5X file containing Add-On Instruction, user defined data types, controller tags and ladder logic required to configure the MVI56E-SIE module
MVI56ESIE_Optional_Rung_vXXX.L5X	Optional L5X file containing additional Add-On Instruction with logic for changing Ethernet configuration and clock settings.

1.7 Creating a New RSLogix 5000 Project

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



2 Select the ControlLogix controller model.
3 Select **REVISION 16**.
4 Enter a name for the controller, such as *My_Controller*.
5 Select the ControlLogix chassis type.
6 Select **SLOT 0** for the controller.

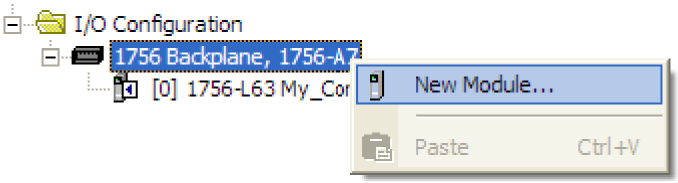

New Controller

Vendor: Allen-Bradley
Type: 1756-L63 ControlLogix5563 Controller
Revision: 16
☐ Redundancy Enabled
Name: My_Controller
Description:
Chassis Type: 1756-A7 7-Slot ControlLogix Chassis
Slot: 0 Safety Partner Slot: <none>
Create In: C:\RSLogix 5000\Projects
Security Authority: No Protection
☐ Use only the selected Security Authority for Authentication and Authorization

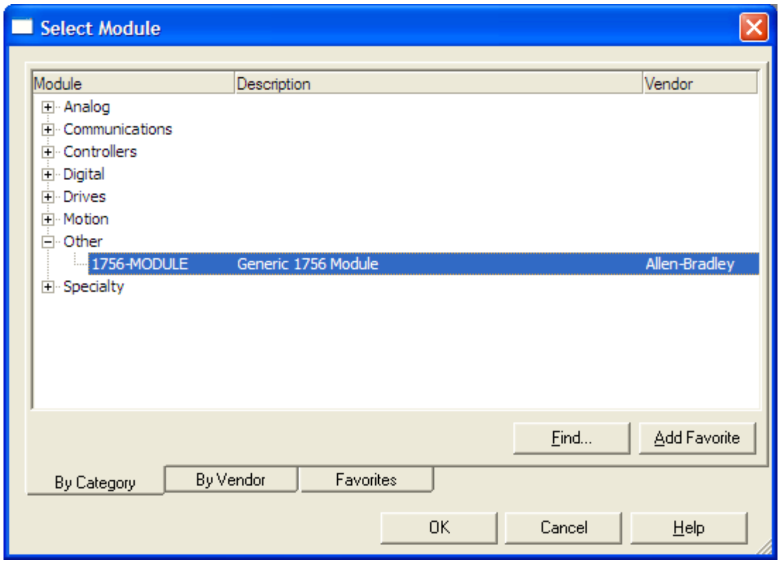
1.7.1 *Creating the Module*

1 Add the MVI56E-SIE 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.



Module	Description	Vendor
Analog		
Communications		
Controllers		
Digital		
Drives		
Motion		
Other		
1756-MODULE	Generic 1756 Module	Allen-Bradley
Specialty		

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

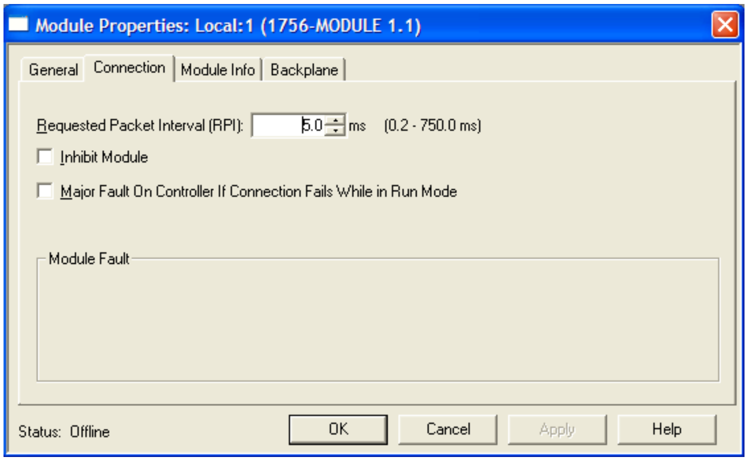
3 In the *New Module* dialog box, enter the following values.

Parameter	Value
Name	Enter a module identification string. Example: SIE
Description	Enter a description for the module. Example: **SIEMENS INDUSTRIAL ETHERNET CLIENT COMMUNICATION MODULE**
Comm Format	Select **DATA-INT**.
Slot	Enter the slot number in the rack where the MVI56E-SIE 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: The *Comm Format* **DATA – INT** must be selected in the dialog box, otherwise the module will not communicate over the backplane of the ControlLogix rack.

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.



Requested Packet Interval (RPI): 5.0 ms (0.2 - 750.0 ms)

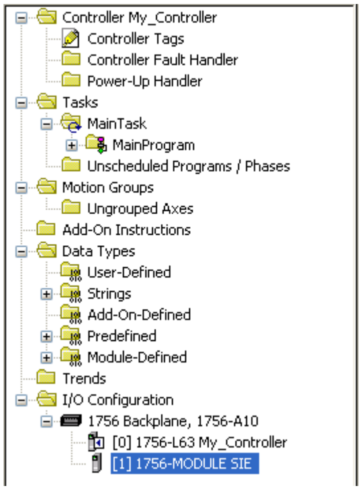
□ Inhibit Module

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

Status: Offline

6 Save the module.

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

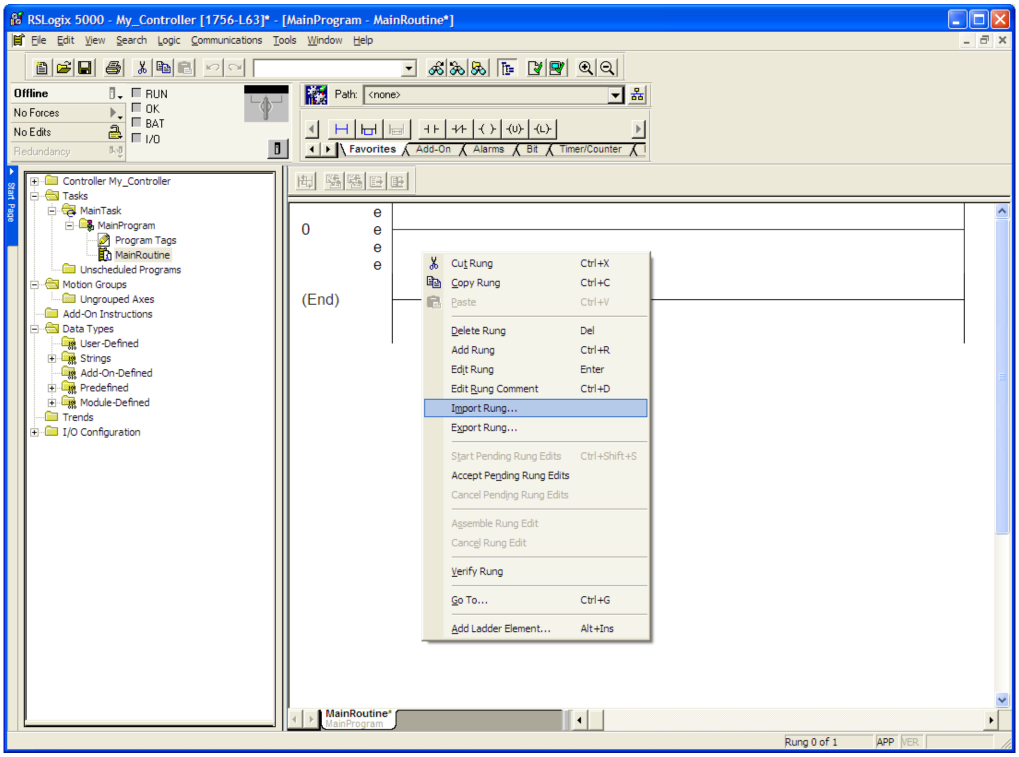


1.7.2 Importing the Add-On Instruction

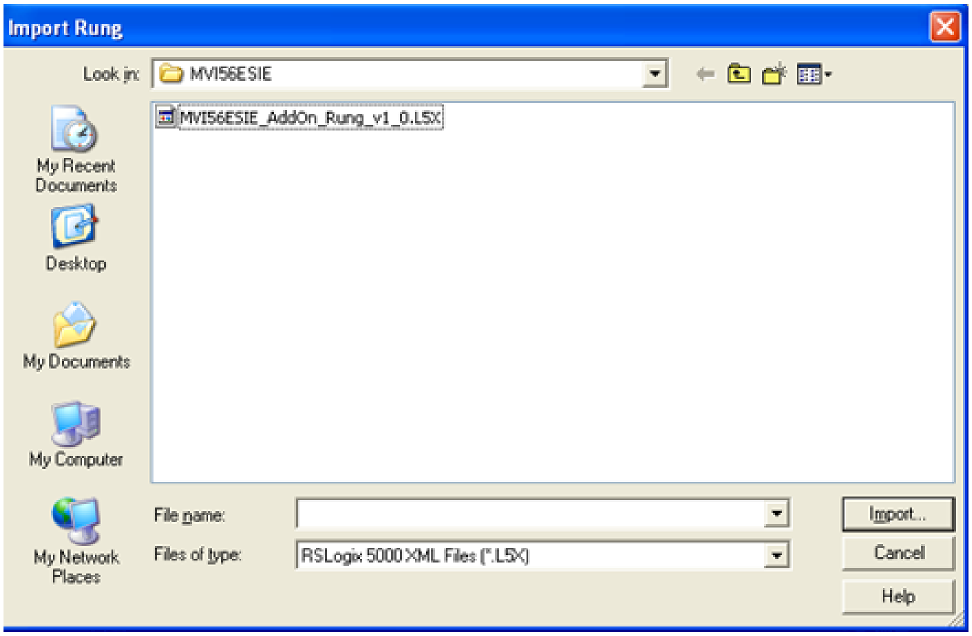
1 In the *Controller Organization* window, expand the **TASKS** folder and subfolder until the **MAINPROGRAM** folder is reached.

2 In the **MAINPROGRAM** folder, double-click to open the **MAINROUTINE** ladder.

3 Select an empty rung in the new routine, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **IMPORT RUNG**.



4 Navigate to the location on the PC where the Add-On Instruction was saved (for example, *My Documents* or *Desktop*). Select the **MVI56ESIE_ADDON_RUNG_V1_X.L5X** file.



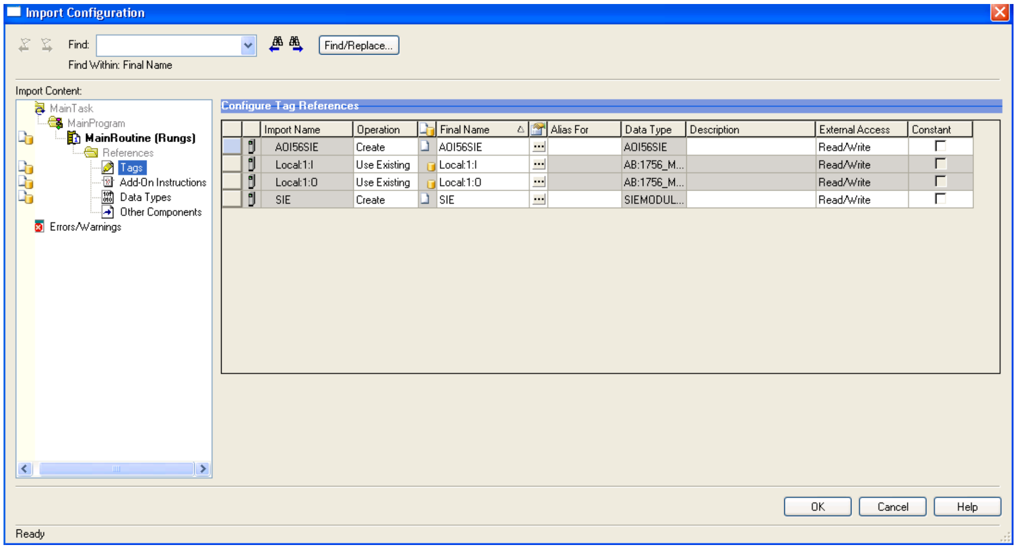
Look in: MVI56ESIE

MVI56ESIE_AddOn_Rung_v1_0.L5X

File name:

Files of type: RSLogix 5000 XML Files (*.L5X)

This action opens the *Import Configuration* dialog box. Clicking on **Tags** Reference will show the controller tags that will be created.



Find:

Find Within: Final Name

Import Content:

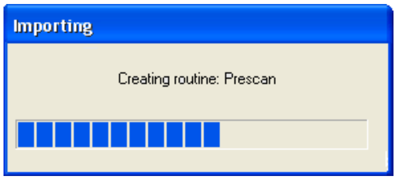
Configure Tag References

Import Name	Operation	Final Name	Alias For	Data Type	Description	External Access	Constant
AOI56SIE	Create	AOI56SIE		AOI56SIE		Read/Write	□
Local:1:I	Use Existing	Local:1:I		AB:1756_M...		Read/Write	□
Local:1:O	Use Existing	Local:1:O		AB:1756_M...		Read/Write	□
SIE	Create	SIE		SIEMODUL...		Read/Write	□

Ready

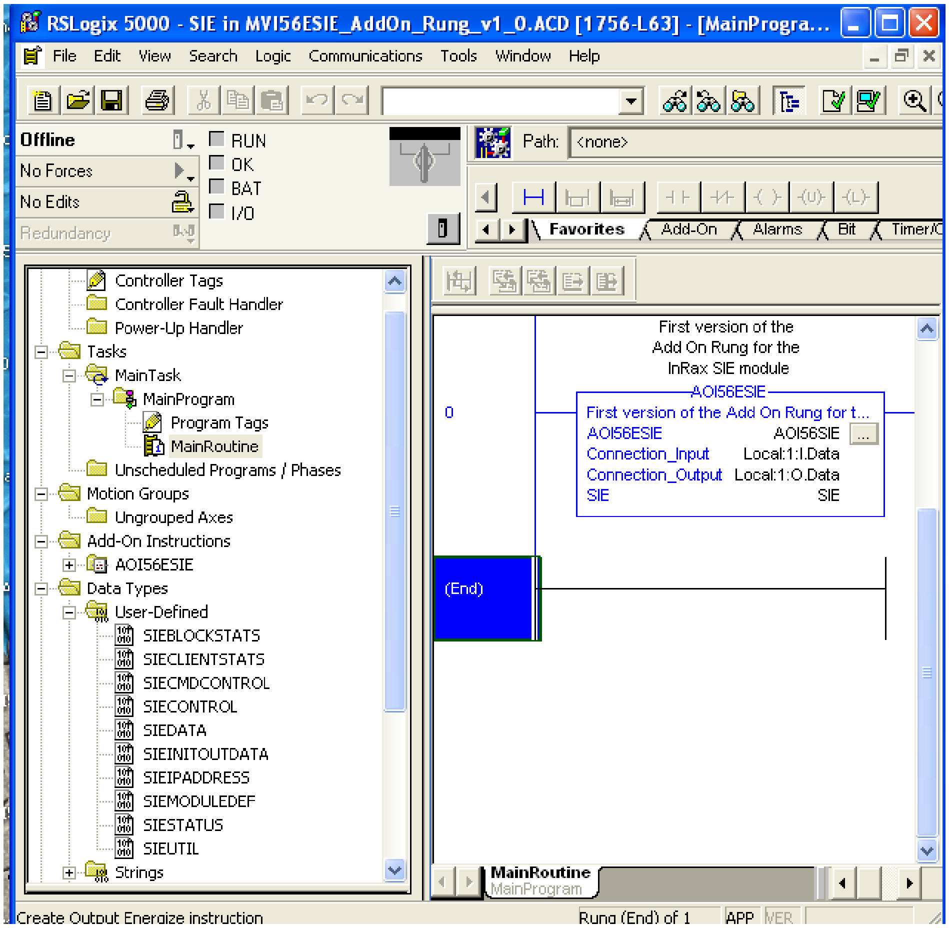
5 If the module is being used in a different slot (or remote rack), select the correct connection input and output variables that define the path to the module. If the module is located in Slot 1 of the local rack, this step is not required.

6 Click **OK** to confirm the import. RSLogix will indicate that the import is in progress:



Creating routine: Prescan

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



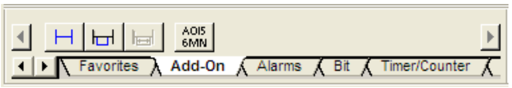
First version of the
Add On Rung for the
InRax SIE module

0

AOI56ESIE
First version of the Add On Rung for t...
AOI56ESIE AOI56SIE
Connection_Input Local:1:I.Data
Connection_Output Local:1:O.Data
SIE SIE

(End)

The procedure has also imported new user-defined data types, data objects and the Add-On Instruction for your project.

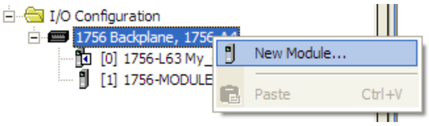


7 Save the application and then download the sample ladder logic to the processor.

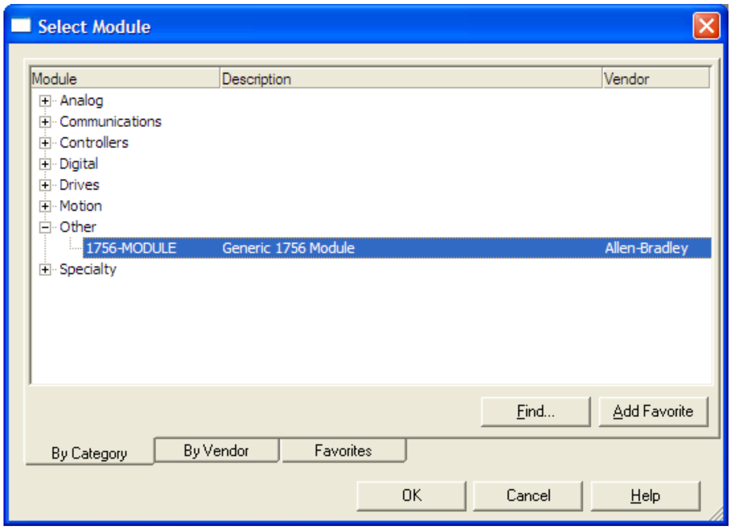
Adding Multiple Modules (Optional)

Important: If the application requires more than one MVI56-SIE 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**.

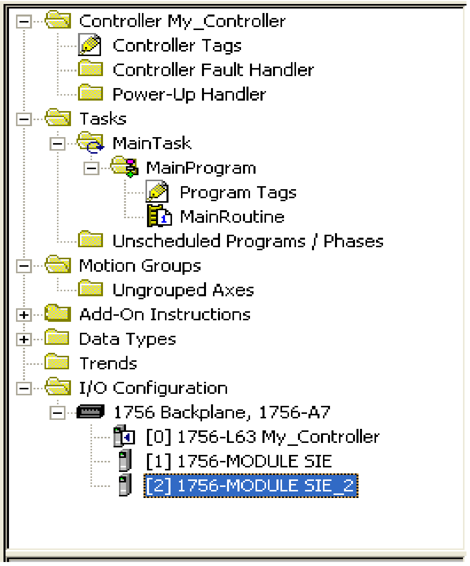


Module	Description	Vendor
1756-MODULE	Generic 1756 Module	Allen-Bradley

3 Fill the module properties as follows:

Parameter	Value
Name	Enter a module identification string. Example: SIE_**2**.
Description	Enter a description for the module. Example: **SIEMENS INDUSTRIAL ETHERNET CLIENT COMMUNICATION MODULE**
Comm Format	Select **DATA-INT.**
Slot	Enter the slot number in the rack where the MVI56E-SIE 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:

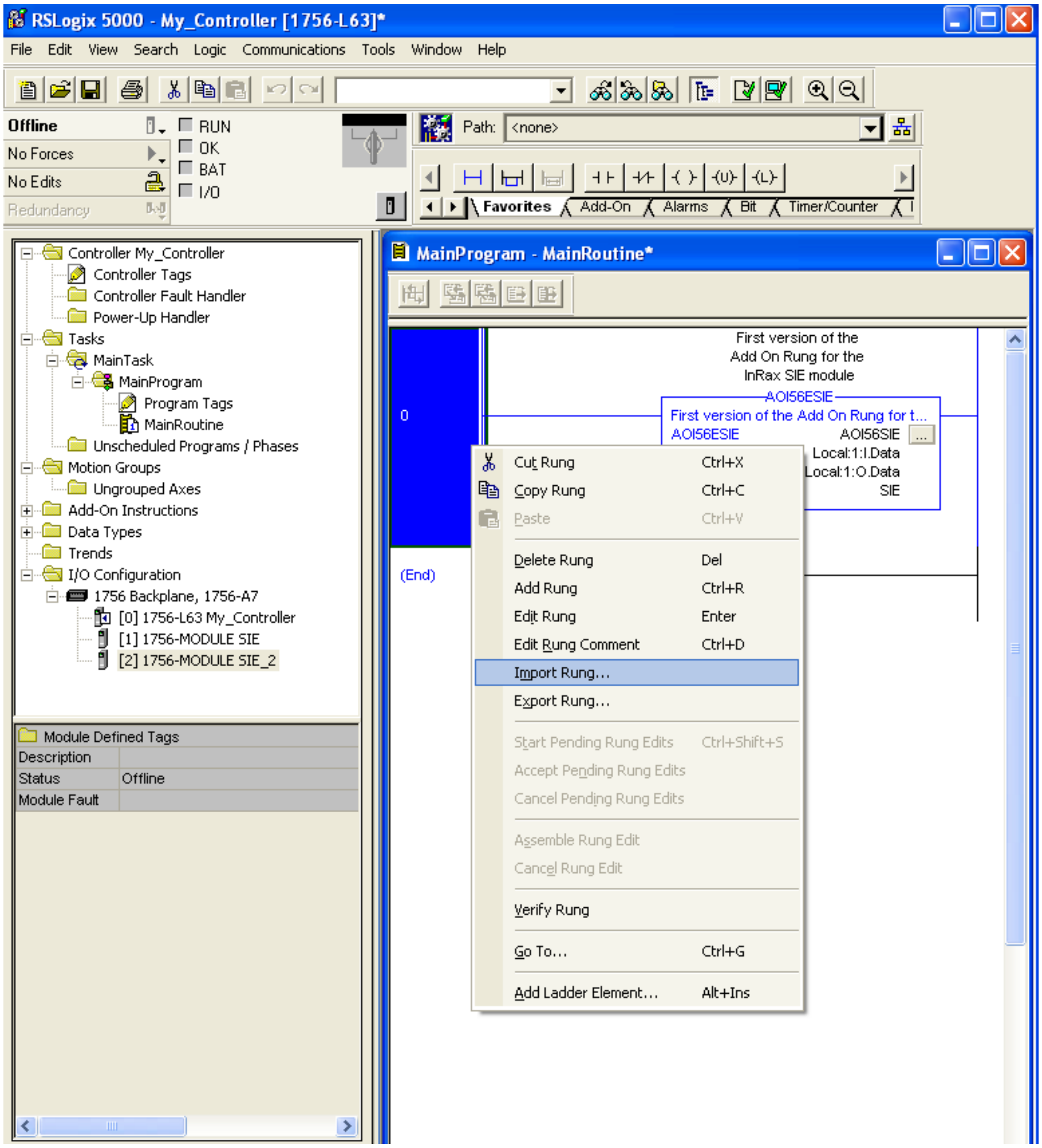


```
Controller My_Controller
    Controller Tags
    Controller Fault Handler
    Power-Up Handler
Tasks
    MainTask
        MainProgram
            Program Tags
            MainRoutine
    Unscheduled Programs / Phases
Motion Groups
    Ungrouped Axes
Add-On Instructions
Data Types
Trends
I/O Configuration
    1756 Backplane, 1756-A7
        [0] 1756-L63 My_Controller
        [1] 1756-MODULE SIE
        [2] 1756-MODULE SIE_2
```

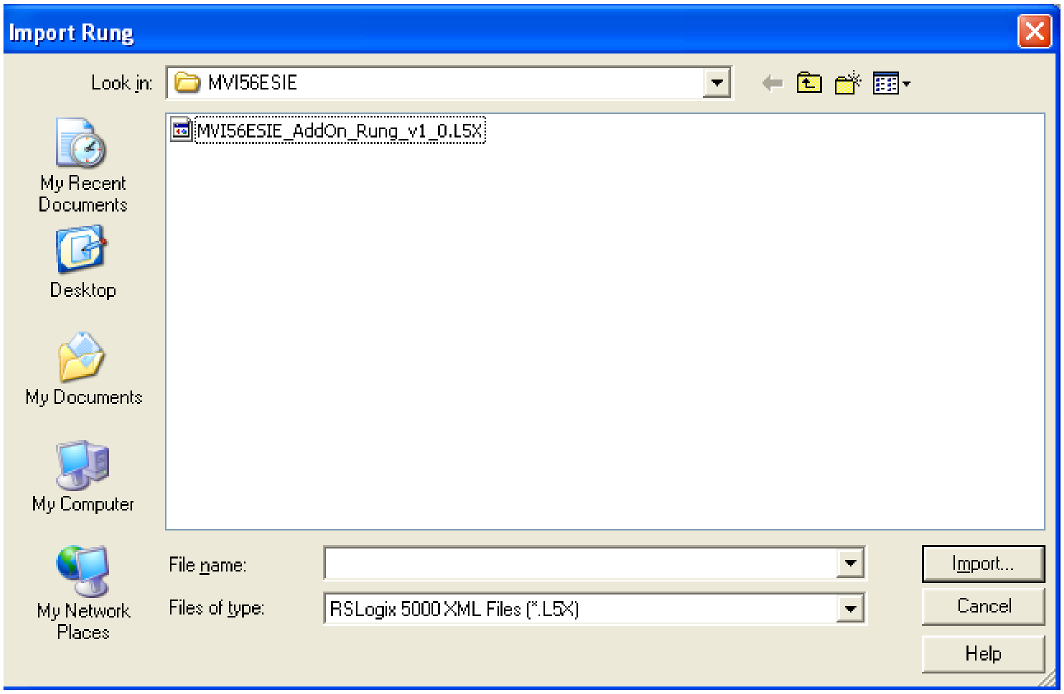
5 Expand the **TASKS** folder, and then expand the **MAINTASK** folder.

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 **MVI56E**SIE_**ADDON_RUNG_V1_0.L5X** file, and then click **IMPORT**.



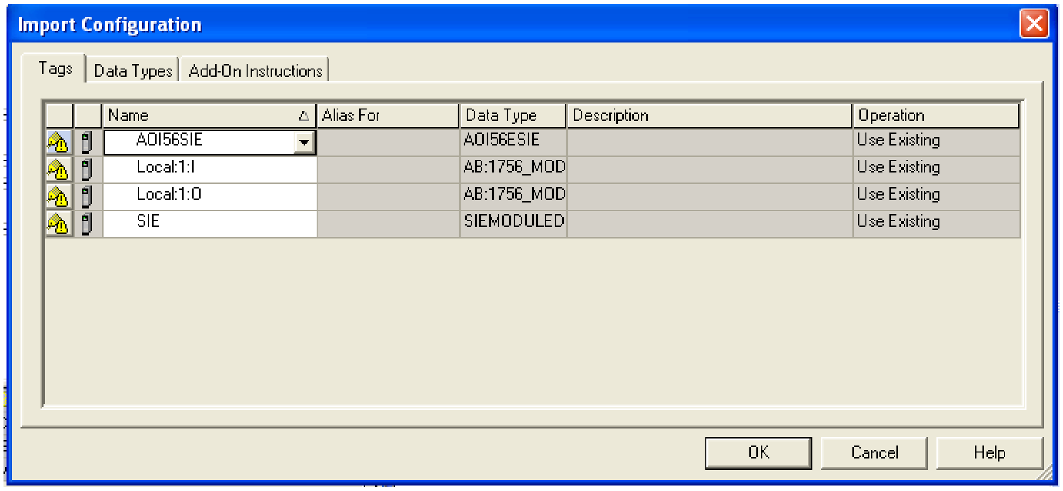
Look in: MVI56ESIE

MVI56ESIE_AddOn_Rung_v1_0.L5X

File name:

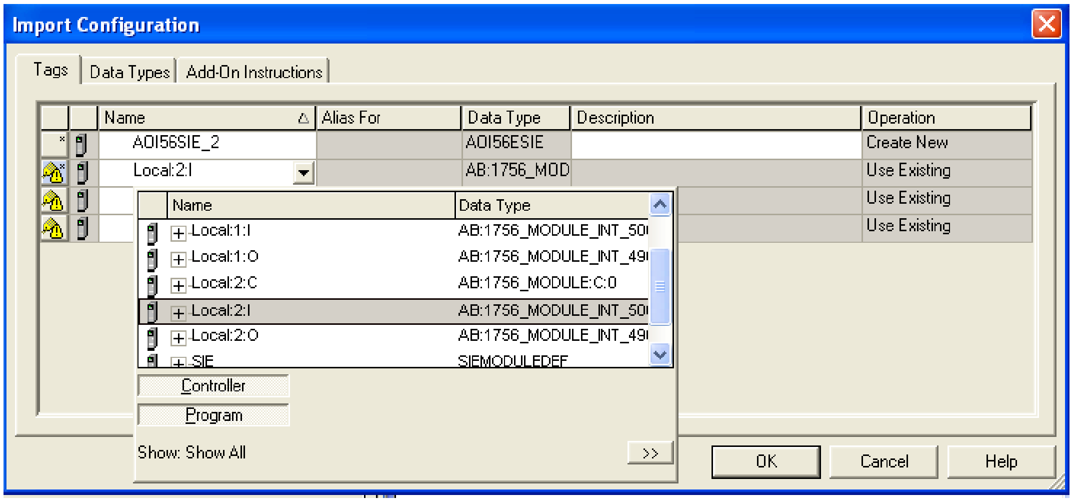
Files of type: RSLogix 5000 XML Files (*.L5X)

9 This action opens the *Import Configuration* window, which shows the tags that will be imported.



Name	Alias For	Data Type	Description	Operation
AOI56SIE		AOI56ESIE		Use Existing
Local:1:I		AB:1756_MOD		Use Existing
Local:1:O		AB:1756_MOD		Use Existing
SIE		SIEMODULED		Use Existing

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.

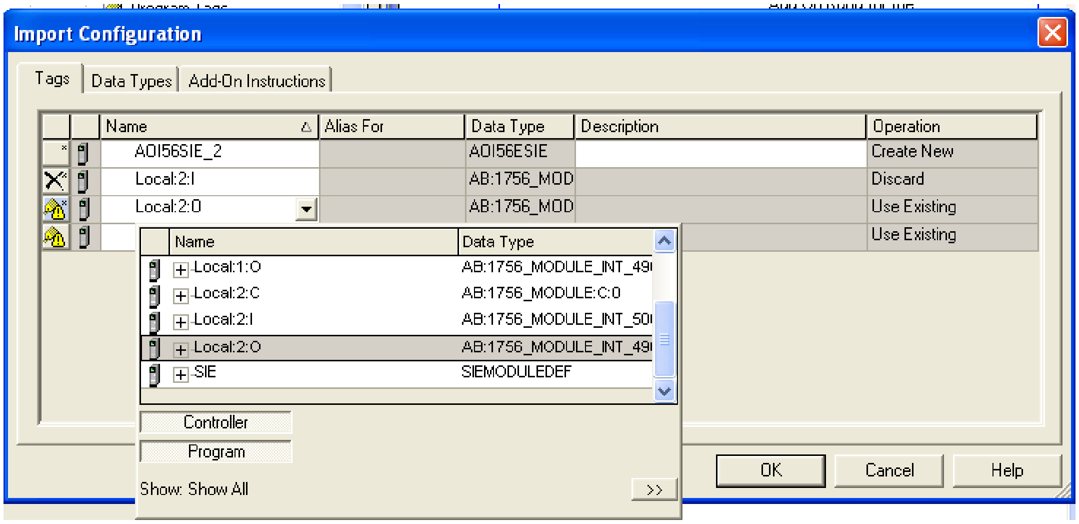


Import Configuration

Tags | Data Types | Add-On Instructions

Name	Alias For	Data Type	Description	Operation
AOI56SIE_2		AOI56ESIE		Create New
Local:2:I		AB:1756_MOD		Use Existing
				Use Existing
				Use Existing

Name	Data Type
Local:1:I	AB:1756_MODULE_INT_50
Local:1:O	AB:1756_MODULE_INT_49
Local:2:C	AB:1756_MODULE:C:0
Local:2:I	AB:1756_MODULE_INT_50
Local:2:O	AB:1756_MODULE_INT_49
SIE	SIEMODULEDEF



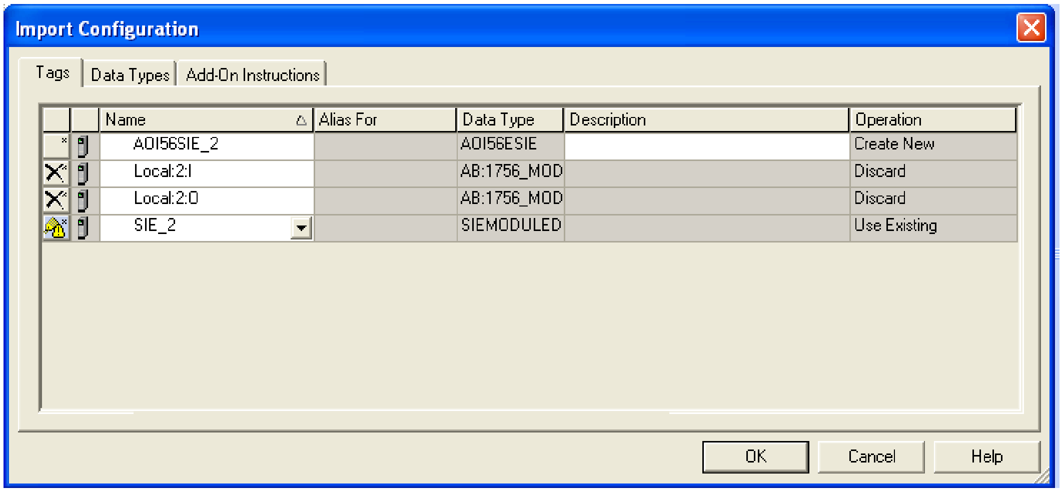
Import Configuration

Tags | Data Types | Add-On Instructions

Name	Alias For	Data Type	Description	Operation
AOI56SIE_2		AOI56ESIE		Create New
Local:2:I		AB:1756_MOD		Discard
Local:2:O		AB:1756_MOD		Use Existing
				Use Existing

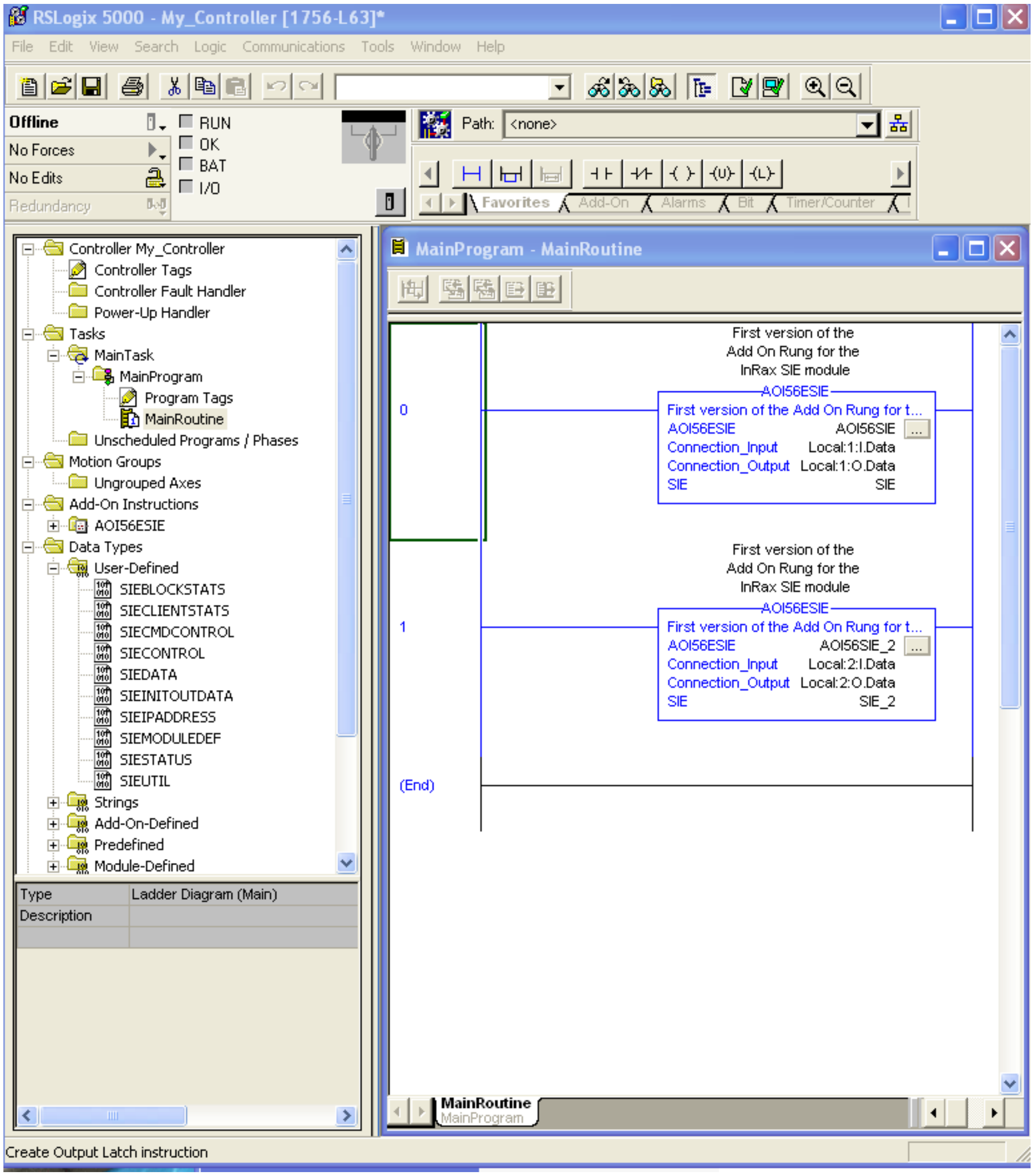
Name	Data Type
Local:1:O	AB:1756_MODULE_INT_49
Local:2:C	AB:1756_MODULE:C:0
Local:2:I	AB:1756_MODULE_INT_50
Local:2:O	AB:1756_MODULE_INT_49
SIE	SIEMODULEDEF

11 Change the default tags *SIE* and *AOI56SIE* 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.



Name	Alias For	Data Type	Description	Operation
AOI56SIE_2		AOI56ESIE		Create New
Local:2:I		AB:1756_MOD		Discard
Local:2:O		AB:1756_MOD		Discard
SIE_2		SIEMODULED		Use Existing

12 Click **OK** to confirm.


RSLogix 5000 - My_Controller [1756-L63]*
File Edit View Search Logic Communications Tools Window Help
Offline
No Forces
No Edits
Redundancy
RUN
OK
BAT
I/O
Path: <none>
Favorites
Add-On
Alarms
Bit
Timer/Counter
MainProgram - MainRoutine
MainRoutine
MainProgram

Controller My_Controller
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Program Tags
MainRoutine
Unscheduled Programs / Phases
Motion Groups
Ungrouped Axes
Add-On Instructions
AOI56ESIE
Data Types
User-Defined
SIEBLOCKSTATS
SIECLIENTSTATS
SIECMDCONTROL
SIECONTROL
SIEDATA
SIEINITOUTDATA
SIEIPADDRESS
SIEMODULEDEF
SIESTATUS
SIEUTIL
Strings
Add-On-Defined
Predefined
Module-Defined

Type	Ladder Diagram (Main)
Description	

0

First version of the
Add On Rung for the
InRax SIE module

AOI56ESIE

First version of the Add On Rung for t...	
AOI56ESIE	AOI56SIE
Connection_Input	Local:1:I.Data
Connection_Output	Local:1:O.Data
SIE	SIE

1

First version of the
Add On Rung for the
InRax SIE module

AOI56ESIE

First version of the Add On Rung for t...	
AOI56ESIE	AOI56SIE_2
Connection_Input	Local:2:I.Data
Connection_Output	Local:2:O.Data
SIE	SIE_2

(End)

Create Output Latch instruction

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

Adjusting the Input and Output Array Sizes

The module's internal database is divided into two user-configurable areas:

- Read Data
- Write Data

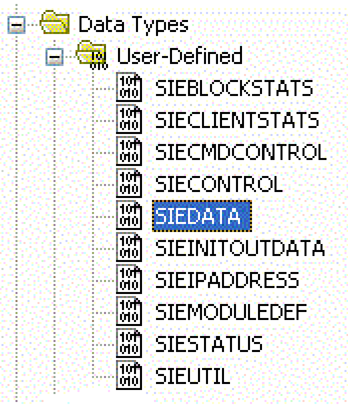
The Read Data is moved from the module to the processor, while the Write Data is moved from the processor to the module.

The MVI56E-SIE Add-On Instruction rung is configured for 600 registers of Read Data and 600 registers of Write Data, which is sufficient for most applications. However, you can configure the sizes of these data areas to meet the needs of your application.

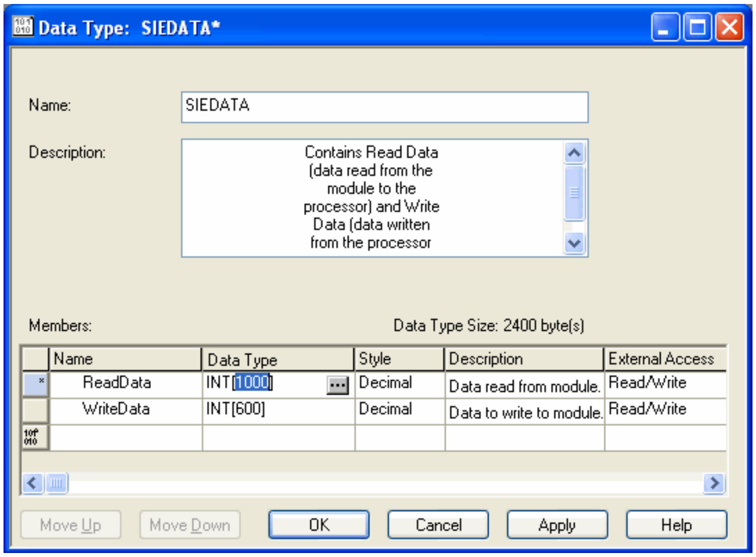
Important: Because the module pages data in blocks of 200 registers at a time, you should configure your user data areas in multiples of 200 registers.

Caution: When the array size is changed, RSLogix may reset the SIE tags to zero. To avoid data loss, be sure to save the settings before continuing.

1 In the *Controller Organizer* pane, expand the *Data Types* and *User-Defined* folders, then double-click **SIEDATA**.



2 In the *Data Type: SIEDATA* dialog box, change the data type designation of the *ReadData* array to the desired value. In the example below, it was changed from **INT[600]** to **INT[1000]**. Click **APPLY**.



Name: SIEDATA

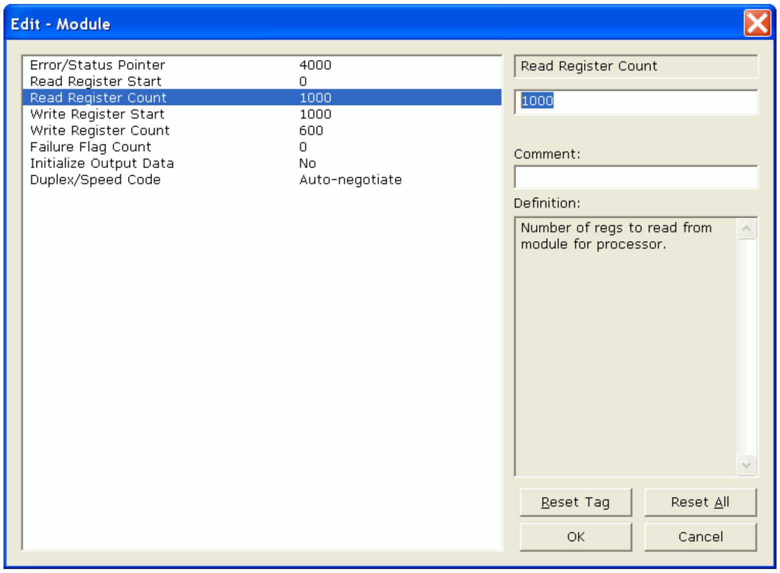
Description: Contains Read Data (data read from the module to the processor) and Write Data (data written from the processor

Members: Data Type Size: 2400 byte(s)

Name	Data Type	Style	Description	External Access
ReadData	INT[1000]	Decimal	Data read from module.	Read/Write
WriteData	INT[600]	Decimal	Data to write to module.	Read/Write

Note: If RSLogix resets your data values, refer to the backup copy of your program to re-enter your configuration parameters.

3 In *ProSoft Configuration Builder*, expand the *Module* icon in the tree view and double-click **MODULE** to open an *Edit* window. Change the **READ REGISTER COUNT** to contain the number of words for your Read Data area.



Error/Status Pointer	4000
Read Register Start	0
Read Register Count	1000
Write Register Start	1000
Write Register Count	600
Failure Flag Count	0
Initialize Output Data	No
Duplex/Speed Code	Auto-negotiate

Read Register Count: 1000

Comment:

Definition: Number of regs to read from module for processor.

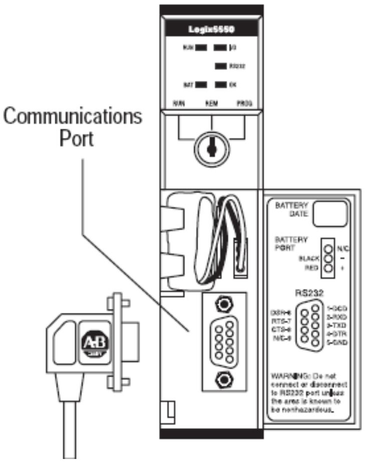
4 To modify the *WriteData* array, follow the above steps, substituting *Write* for *Read*.
5 Save and download the configuration to the module and reboot.

Make sure that the *ReadData* and *WriteData* arrays do not overlap in the module memory. For example, if your application requires 2000 words of Write Data starting at register 0, then your *Read Register Start* parameter must be set to a value of 2000 or greater.

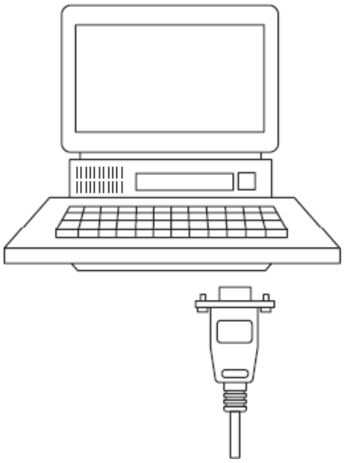
1.8 Connecting Your PC to the ControlLogix Processor

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. It is not mandatory to use the processor's serial interface. The processor may be accessed through whatever network interface is available on the system. Refer to the Rockwell Automation documentation for information on other connection methods.

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



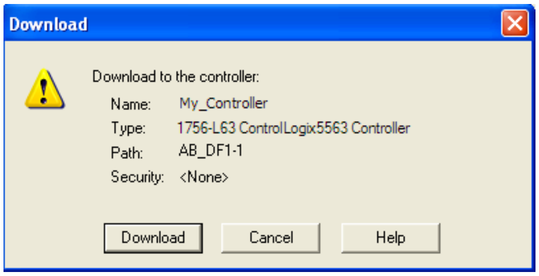
2 Connect the straight connector end of the cable to the serial port on the computer.



1.9 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.

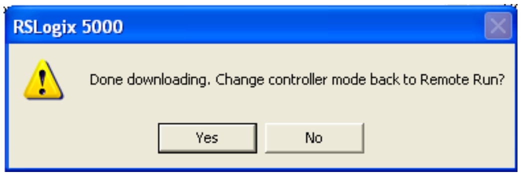
1. If the processor is not already online, open the *Communications* menu, and then choose **DOWNLOAD**. RSLogix 5000 will establish communication with the processor. Download does not have to occur using the processor's serial port, as shown here. Download may be achieved through any available network 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.


Download

Download to the controller:

Name: My_Controller
Type: 1756-L63 ControlLogix5563 Controller
Path: AB_DF1-1
Security: <None>

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.


RSLogix 5000

Done downloading. Change controller mode back to Remote Run?

Note: If an error message is recieved during these steps, refer to the RSLogix documentation to interpret and correct the error.

2 Configuring the MVI56E-SIE Module

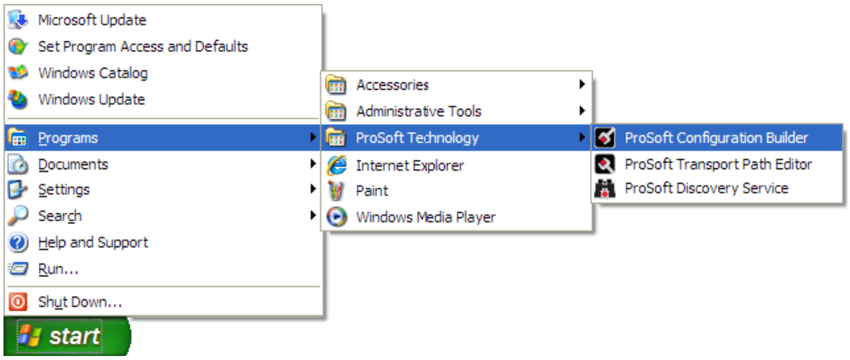
2.1 Using ProSoft Configuration Builder Software

ProSoft Configuration Builder (PCB) provides a convenient way to manage module configuration files customized to meet application needs. *PCB* is not only a powerful solution for new configuration files, but also allows importing of information from previously installed (known working) configurations to new projects.

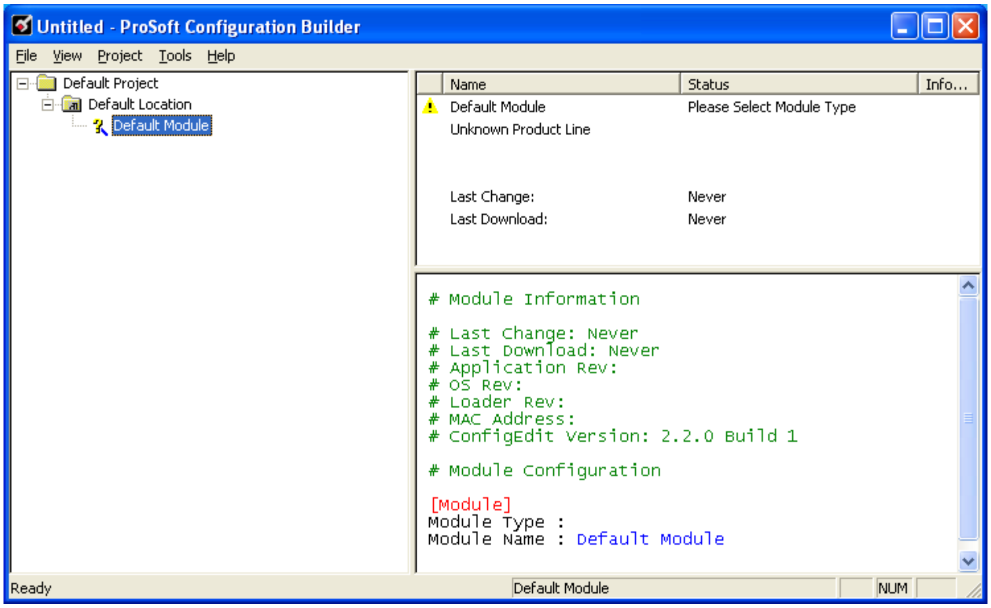
> **Note:** During startup and initialization, the MVI56E-SIE module receives its protocol and backplane configuration information from the installed Personality Module (Compact Flash). Use *ProSoft Configuration Builder* to configure module settings and to download changes to the Personality Module.

2.1.1 Setting Up the Project

To begin, start **PROSOFT CONFIGURATION BUILDER** (PCB).



If other Windows configuration tools have been used before, the screen layout will be found to be familiar. *PCB*'s window consists of a tree view on the left, and an information pane and a configuration pane on the right side of the window. When first starting *PCB*, the tree view consists of folders for *Default Project* and *Default Location*, with a *Default Module* in the *Default Location* folder. The following illustration shows the *PCB* window with a new project.

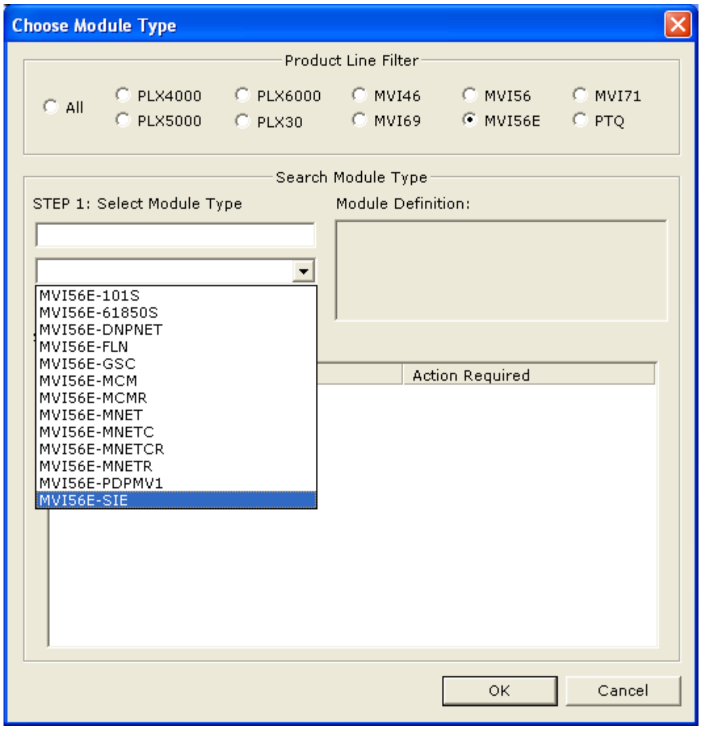

Untitled - ProSoft Configuration Builder
File View Project Tools Help
Default Project
Default Location
Default Module
Ready
Default Module
NUM

Name	Status	Info...
Default Module	Please Select Module Type	
Unknown Product Line		
Last Change:	Never	
Last Download:	Never	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 2.2.0 Build 1

# Module Configuration

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

The first task is to add the MVI56E-SIE module to the project.

1. Use the mouse to select **DEFAULT MODULE** in the tree view, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, select **CHOOSE MODULE TYPE**. This action opens the *Choose Module Type* dialog box.



Choose Module Type

Product Line Filter

○ All ○ PLX4000 ○ PLX5000 ○ PLX6000 ○ PLX30 ○ MVI46 ○ MVI69 ○ MVI56 ● MVI56E ○ MVI71 ○ PTQ

Search Module Type

STEP 1: Select Module Type

Module Definition:

MVI56E-101S
MVI56E-61850S
MVI56E-DNPNET
MVI56E-FLN
MVI56E-GSC
MVI56E-MCM
MVI56E-MCMR
MVI56E-MNET
MVI56E-MNETC
MVI56E-MNETCR
MVI56E-MNETR
MVI56E-PDPMV1
MVI56E-SIE

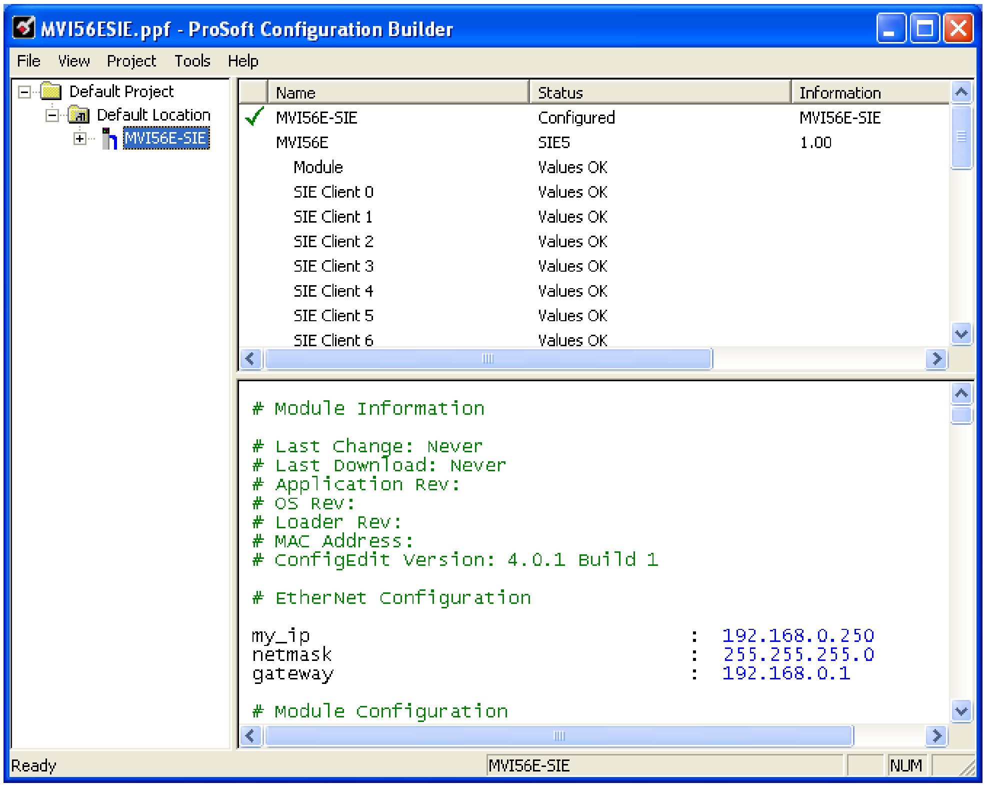
Action Required

OK Cancel

3. In the *Product Line Filter* area of the dialog box, select **MVI56E**. In the *Select Module Type* dropdown list, select **MVI56E-SIE**, and then click **OK** to save the settings and return to the *ProSoft Configuration Builder* window.

2.1.2 Setting Module Parameters

Notice that the contents of the information pane and the configuration pane changed when the MVI56E-SIE module was added to the project.



Name	Status	Information
✓ MVI56E-SIE	Configured	MVI56E-SIE
MVI56E	SIE5	1.00
Module	Values OK	
SIE Client 0	Values OK	
SIE Client 1	Values OK	
SIE Client 2	Values OK	
SIE Client 3	Values OK	
SIE Client 4	Values OK	
SIE Client 5	Values OK	
SIE Client 6	Values OK	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 4.0.1 Build 1

# EtherNet Configuration

my_ip                                      :  192.168.0.250
netmask                                    :  255.255.255.0
gateway                                    :  192.168.0.1

# Module Configuration
```

At this time, the *Default Project* and *Default Location* folders may be renamed in the tree view.

<u>Renaming an Object</u>

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

<u>Configuring Module Parameters</u>

1. Click the **[+]** sign next to the module icon to expand module information.
2. Click the **[+]** sign next to any icon to view module information and configuration options.
3. Double-click any icon to open an *Edit* dialog box.
4. To edit a parameter, select the parameter in the left pane and make the changes in the right pane.
5. Click **OK** to save the changes.

Printing a Configuration File

1 Select the module icon, and then click the right mouse button to open a shortcut menu.

2 On the shortcut menu, choose **VIEW CONFIGURATION.** This action opens the *View Configuration* window.

3 In the *View Configuration* window, open the **FILE** menu, and choose **PRINT**. This action opens the *Print* dialog box.

4 In the *Print* dialog box, choose the printer to use from the drop-down list, select printing options, and then click **OK**.

2.1.3 Module

This section of the configuration describes the database setup and module-level parameters.

Backplane Error/Status Pointer

1 to **4955**

This parameter sets the address in the internal database where the backplane error/status data will be placed. If you want the error/status data to be moved to the processor and placed into the *ReadData* array, the value entered should be a module memory address in the Read Data area. If the value is set to **-1**, the error/status data will not be stored in the module's internal database and will not be transferred to the processor's *ReadData* array.

Enabling the *Error/Status Pointer* is optional. The error/status data is routinely returned as part of the input image, which is continually being transferred from the module to the processor. For more information, see Normal Data Transfer Blocks (page 127).

Read Register Start

0 to **4999**

This parameter specifies the start of the Read Data area in module memory. Data in this area will be transferred from the module to the processor.

Note: Total user database memory space is limited to the first 5000 registers of module memory, addresses 0 through 4999. Therefore, the practical limit for this parameter is 4999 minus the value entered for *Read Register Count*, so that the Read Data Area does not try to extend above address 4999. Read Data and Write Data Areas must be configured to occupy separate address ranges in module memory and should not be allowed to overlap.

Read Register Count

0 to **5000**

This parameter specifies the size of the Read Data area of module memory and the number of registers to transfer from this area to the processor, up to a maximum of 5000 words.

Note: Total *Read Register Count* and *Write Register Count* cannot exceed 5000 total registers. Read Data and Write Data Areas must be configured to occupy separate address ranges in module memory and should not be allowed to overlap.

Write Register Start

0 to **4999**

This parameter specifies the start of the Write Data area in module memory. Data in this area will be transferred in from the processor.

Note: Total user database memory space is limited to the first 5000 registers of module memory, addresses 0 through 4999. Therefore, the practical limit for this parameter is 4999 minus the value entered for *Write Register Count*, so that the Write Data Area does not try to extend above address 4999. Read Data and Write Data Areas must be configured to occupy separate address ranges in module memory and should not be allowed to overlap.

Write Register Count

0 to **5000**

This parameter specifies the size of the Write Data area of module memory and the number of registers to transfer from the processor to this memory area, up to a maximum value of 5000 words.

Note: Total *Read Register Count* and *Write Register Count* cannot exceed 5000 total registers. Read Data and Write Data Areas must be configured to occupy separate address ranges in module memory and should not be allowed to overlap.

Failure Flag Count

If this value is greater than zero the protocol communication will be interrupted once a backplane failure is detected, or communication with the processor fails. A value of zero will disable this feature.

Initialize Output Data

0 = No, **1** = Yes

This parameter is used to determine if the output data for the module should be initialized with values from the processor. If the value is set to **0**, the output data will be initialized to 0. If the value is set to **1**, the data will be initialized with data from the processor. Use of this option requires associated ladder logic to pass the data from the processor to the module.

Duplex/Speed Code

0, **1**, **2**, **3** or **4**

This parameter allows the module to use a specific duplex and speed setting.

- Value = **1**: Half duplex, 10 MB speed
- Value = **2**: Full duplex, 10 MB speed
- Value = **3**: Half duplex, 100 MB speed
- Value = **4**: Full duplex, 100 MB speed
- Value = **0**: Auto-negotiate

Auto-negotiate is the default value.

2.1.4 SIE Client x

This section defines general configuration for the SIE Client (Master).

Client Error/Status Pointer

-1 to **4990**

This parameter sets the address in the internal database where the Client error/status data will be placed. If the error/status data needs to be moved to the processor and placed into the *ReadData* array, the value entered should be a module memory address in the Read Data area. If the value is set to **-1**, the error/status data will not be stored in the module's internal database and will not be transferred to the processor's *ReadData* array.

Enabling the *Error/Status Pointer* is optional. Alternatively, the error/status data for a specific Client can be requested by the processor and returned in a special Client Status block. For more information, see Client Status Blocks (page 132).

Command Error Pointer

-1 to **4999**

This parameter sets the address in the internal database where the Command Error List data will be placed. If you want the Command Error List data to be moved to the processor and placed into the *ReadData* array, the value entered should be a module memory address in the Read Data area. If the value is set to **-1**, the Command Error List data will not be stored in the module's internal database and will not be transferred to the processor's *ReadData* array.

Enabling the *Command Error Pointer* is optional. Alternatively, the Command Error List data for a specific Client can be requested by the processor and returned in a special Client Status block. For more information, see Client Status Blocks (page 132).

Minimum Command Delay

0 to **65535** milliseconds

This parameter specifies the number of milliseconds to wait between the initial issuances of a command. This parameter can be used to delay all commands sent to servers to avoid "flooding" commands on the network. This parameter does not affect retries of a command as they will be issued when failure is recognized.

Response Timeout

0 to **65535** milliseconds

This parameter specifies the time in milliseconds that a Client will wait before re-transmitting a command if no response is received from the addressed server. The value to use depends on the type of communication network used, and the expected response time of the slowest device on the network.

Retry Count

0 to **10**

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

Command Error Delay

0 to **300**

This parameter specifies the number of 100 millisecond intervals to turn off a command in the error list after an error is recognized for the command. If this parameter is set to **0**, there will be no delay.

2.1.5 SIE Client x Commands

The *SIE Client x Commands* section of the configuration sets the Siemens Industrial Ethernet Client command list. This command list polls Siemens Industrial Ethernet server devices attached to the Siemens Industrial Ethernet Client port. The module supports numerous commands. This permits the module to interface with a wide variety of Siemens Industrial Ethernet protocol devices.

The function codes used for each command are those specified in the Siemens Industrial Ethernet protocol. Each command list record has the same format. The first part of the record contains the information relating to the MVI56E-SIE communication module, and the second part contains information required to interface to the Siemens Industrial Ethernet server device.

Command List Overview

In order to interface the module with Siemens Industrial Ethernet server devices, a command list must be constructed. The commands in the list specify the server device to be addressed, the function to be performed (read or write), the data area in the device to interface with, and the registers in the internal database to be associated with the device data. The Client command list supports up to 16 commands.

The command list is processed from top (command #1) to bottom. A poll interval parameter is associated with each command to specify a minimum delay time in tenths of a second between the issuances of a command. If the user specifies a value of **10** for the parameter, the command will be executed no more frequently than every 1 second.

Commands Supported by the Module

The format of each command in the list depends on the Siemens Industrial Ethernet Function Code being executed.

The following table lists the functions supported by the module. The type of functions that will be supported will also depend on the server device and what it can support. Below are examples of S7-200, S7-300 and S7-1200 functions that are supported using MVI56E-SIE module.

S7-300:

Data Block:

Address Type	Function	Data Type
DB	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	COUNT
	Write	COUNT

Timer:

Address Type	Function	Data Type
Timer	READ	TIME

Counter:

Address Type	Function	Data Type
Counter	READ	Count

Flag:

Address Type	Function	Data Type
Flag	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

Output:

Address Type	**Function**	**Data Type**
Output	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

Input:

Address Type	**Function**	**Data Type**
Input	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

S7-200:

Data Block:

Address Type	Function	Data Type
DB	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT

Flag:

Address Type	Function	Data Type
Flag	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT

Output:

Address Type	Function	Data Type
Output	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT

Input:

Address Type	Function	Data Type
Input	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT

S7-1200:

Data Block:

Address Type	Function	Data Type
DB	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	COUNT
	Write	COUNT

Flag:

Address Type	Function	Data Type
Flag	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

Output:

Address Type	Function	Data Type
Output	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

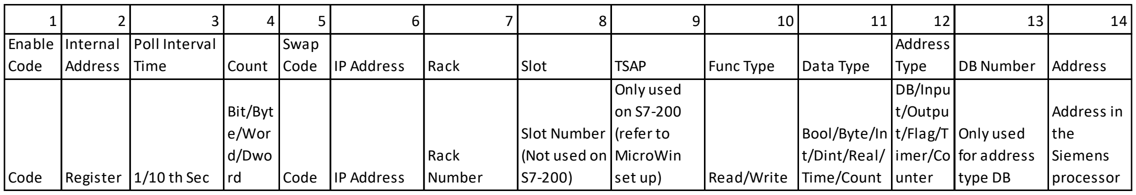
Input:

Address Type	Function	Data Type
Input	READ	BOOL
	Write	BOOL
	READ	BYTE
	Write	BYTE
	READ	DINT
	Write	DINT
	READ	REAL
	Write	REAL
	READ	INT
	Write	INT
	READ	TIME
	Write	TIME
	READ	Count
	Write	Count

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 Siemens Industrial Ethernet server device.

Command Entry Formats

The following table shows the structure of the configuration data necessary for each of the supported commands.



1	2	3	4	5	6	7	8	9	10	11	12	13	14
Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	IP Address	Rack	Slot	TSAP	Func Type	Data Type	Address Type	DB Number	Address
Code	Register	1/10 th Sec	Bit/Byte/Word/Dword	Code	IP Address	Rack Number	Slot Number (Not used on S7-200)	Only used on S7-200 (refer to MicroWin set up)	Read/Write	Bool/Byte/Int/Dint/Real/Time/Count	DB/Input/Output/Flag/Timer/Counter	Only used for address type DB	Address in the Siemens processor

The first part of the record is the module information, which relates to the MVI56E module, and the second part contains information required to interface to the server device.

Command list example:

Enable

NO (0) or **YES** (1)

This parameter specifies whether or not the command is to be executed.

Value	Description
NO (0)	The command is disabled and will not be executed in the normal polling sequence.
YES (1)	The command is executed each scan of the command list if the Poll Interval Time is set to zero (**0**). If the Poll Interval time is set, the command will be executed when the interval timer expires.

Important: The commands must also be enabled in the ladder logic in order for them to be executed. The *SIE.CONTROL.CmdControl.WriteCmdBits[x]* controller tag array holds 16-command bit arrays for each Client. If a bit for a specific command is set to zero (**0**) in the *WriteCmdBits[x]* controller tag, the command will not be executed, regardless of its enabled or disabled state in the configuration. For more information, see Command Control Blocks (page 133).

Internal Address

0 to **65535** (for bit-level addressing) or

0 to **4999** (for word-level addressing)

This parameter specifies the database address in the module's internal database to use as the destination for data brought in by a read command or as the source for data to be sent out by a write command. The database address is interpreted as a bit address or a 16-bit word (register) address, depending on the Siemens Industrial Ethernet’s Data Type used in the command. If Data Type – Bool is used in the command list, then the database address will be interpreted as a bit address. When any other data types are used, then the database address is interpreted as 16-bit word (register) address.

Poll Interval

0 to **65535**

This parameter specifies the minimum interval between issuances of a command during continuous command execution (*Enable* code of **1**). The parameter is entered in tenths of a second. Therefore, if a value of **100** is entered for a command, the command executes no more frequently than every 10 seconds.

Reg Count

Regs: This Count will depend on the Siemens processor type

Coils: This Count will depend on the Siemens processor type

Below are the tables of the max reg count limit for three of the Siemens processors:

CPU315-2 DP

Data Block:

Address Type	**Function**	**Data Type**	**Max Reg Cnt**	**Max Reg Cnt**
DB	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	164	
	Write	BYTE		164
	READ	DINT	41	
	Write	DINT		41
	READ	REAL	41	
	Write	REAL		41
	READ	INT	82	
	Write	INT		82
	READ	TIME	82	
	Write	TIME		41
	READ	COUNT	82	
	Write	COUNT		82

Timer:

Address Type	Function	Data Type	Max Reg Cnt
Timer	READ	TIME	1

Counter:

Address Type	Function	Data Type	Max Reg Cnt
Counter	READ	Count	111

Flag:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Flag	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	222	
	Write	BYTE		212
	READ	DINT	55	
	Write	DINT		53
	READ	REAL	55	
	Write	REAL		53
	READ	INT	111	
	Write	INT		106
	READ	TIME	111	
	Write	TIME		53
	READ	Count	111	
	Write	Count		106

Output:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Output	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	128	
	Write	BYTE		128
	READ	DINT	32	
	Write	DINT		32
	READ	REAL	32	
	Write	REAL		32
	READ	INT	64	
	Write	INT		64
	READ	TIME	64	
	Write	TIME		32
	READ	Count	64	
	Write	Count		64

Input:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Input	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	128	
	Write	BYTE		128
	READ	DINT	32	
	Write	DINT		32
	READ	REAL	32	
	Write	REAL		32
	READ	INT	64	
	Write	INT		64
	READ	TIME	64	
	Write	TIME		32
	READ	Count	64	
	Write	Count		64

CPU1212C:

Data Block:

Address Type	Function	Data Type	Max Read	Max Write
DB	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	30	
	Write	BYTE		30
	READ	DINT	7	
	Write	DINT		7
	READ	REAL	7	
	Write	REAL		7
	READ	INT	15	
	Write	INT		15
	READ	TIME	15	
	Write	TIME		15
	READ	COUNT	15	
	Write	COUNT		15

Flag:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Flag	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	212	
	Write	BYTE		212
	READ	DINT	53	
	Write	DINT		53
	READ	REAL	53	
	Write	REAL		53
	READ	INT	106	
	Write	INT		106
	READ	TIME	105	
	Write	TIME		105
	READ	Count	106	
	Write	Count		106

Output:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Output	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	212	
	Write	BYTE		212
	READ	DINT	53	
	Write	DINT		53
	READ	REAL	53	
	Write	REAL		53
	READ	INT	106	
	Write	INT		106
	READ	TIME	105	
	Write	TIME		105
	READ	Count	111	
	Write	Count		106

Input:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Input	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	222	
	Write	BYTE		212
	READ	DINT	55	
	Write	DINT		53
	READ	REAL	55	
	Write	REAL		53
	READ	INT	111	
	Write	INT		111
	READ	TIME	111	
	Write	TIME		106
	READ	Count	111	
	Write	Count		106

CPU224XP:

Data Block:

Address Type	Function	Data Type	Max Read	Max Write
DB	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	222	
	Write	BYTE		212
	READ	DINT	55	
	Write	DINT		53
	READ	REAL	55	
	Write	REAL		53
	READ	INT	111	
	Write	INT		106

Flag:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Flag	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	32	
	Write	BYTE		32
	READ	DINT	8	
	Write	DINT		8
	READ	REAL	8	
	Write	REAL		8
	READ	INT	16	
	Write	INT		16

Output:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Output	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	16	
	Write	BYTE		16
	READ	DINT	4	
	Write	DINT		4
	READ	REAL	4	
	Write	REAL		4
	READ	INT	8	
	Write	INT		8

Input:

Address Type	Function	Data Type	Max Reg Cnt	Max Reg Cnt
Input	READ	BOOL	1	
	Write	BOOL		1
	READ	BYTE	16	
	Write	BYTE		16
	READ	DINT	4	
	Write	DINT		4
	READ	REAL	4	
	Write	REAL		4
	READ	INT	8	
	Write	INT		8

Swap Code

NONE

SWAP WORDS

SWAP WORDS & BYTES

SWAP BYTES

This parameter specifies if and how the order of bytes in data received or sent is to be rearranged. This option exists to allow for the fact that different manufacturers store and transmit multi-byte data in different combinations. This parameter is helpful when dealing with floating-point or other multi-byte values, as there is no one standard method of storing these data types. This parameter can be set to rearrange the byte order of data received or sent into an order more useful or convenient for other applications. The following table defines the valid *Swap Code* values and the effect they have on the byte-order of the data.

Swap Code	Description
NONE	No change is made in the byte ordering (1234 = 1234)
SWAP WORDS	The words are swapped (1234=3412)
SWAP WORDS & BYTES	The words are swapped, then the bytes in each word are swapped (1234=4321)
SWAP BYTES	The bytes in each word are swapped (1234=2143)

These swap operations affect 4-byte (or 2-word) groups of data. Therefore, data swapping using these *Swap Codes* should be done only when using an even number of words, such as when 32-bit integer or floating-point data is involved.

Node IP Address

xxx.xxx.xxx.xxx

This parameter specifies the IP address of the device being addressed by the command.

Rack

Rack number of the S7-300, S7-400 or S7-1200 CPU.

Rack number is not used for the S7-200 CPU.

Slot

Slot number of the S7-300, S7-400 or S7-1200 CPU.

Slot number is not used for the S7-200 CPU.

TSAP

TSAP of the S7-200 CPU. This can be found in the Siemens STEP 7 MicroWIN software.

TSAP is not used for the S7-300, S7-400 and S7-1200.

Func Type

This parameter can either be Read or Write.

Data Type

This parameter can be: BOOL, BYTE, DINT, REAL, INT, TIME, or COUNT.

Address Type

This parameter can be: INPUT, OUTPUT, FLAG, TIMER, COUNTER or DB (Data Block).

DB Number

This parameter specifies the Data Block number to be used with the command.

DB Number is only used when the Address Type is set to DB.

Address

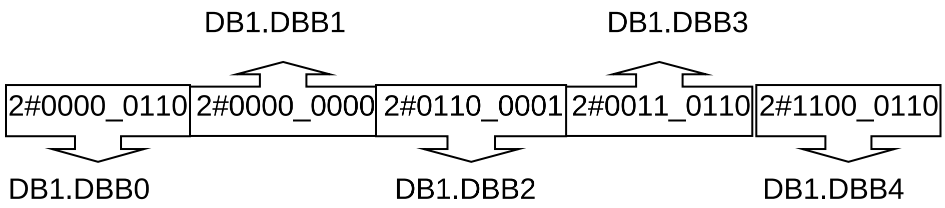
For Read or Write operations using the INT, DINT, REAL or BYTE Data Types, the address is a byte address.

For Read or Write operations using the BOOL Data Type, the address is a bit address.

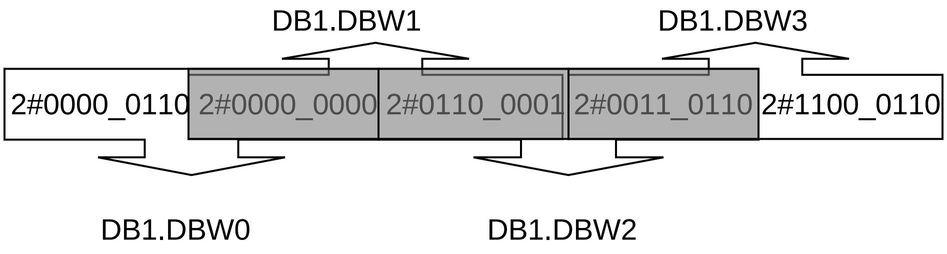
Please note below:

S7-300/S7-1200 Processor:

Byte Address in Data Block:



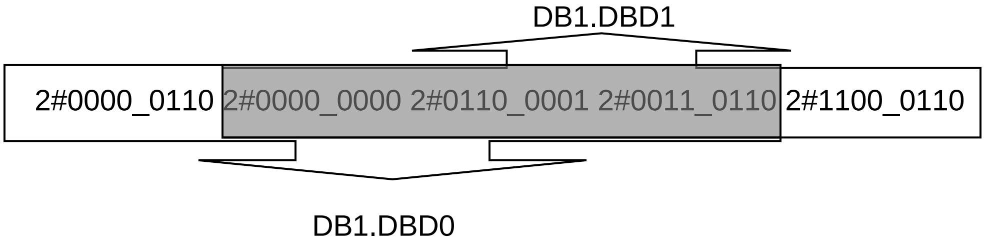
Word Address in Data Block:



The gray area above represents the byte memory locations being overlapped when word address is used consecutively (DB1.DBW0, DB1.DBW1, DB1.DBW2, etc).

If DB1.DBW0 is used as the first address in the Siemens processor, the next word address that can be used without overwriting the data would be DB1.DBW2.

Double Word Address in Data Block:



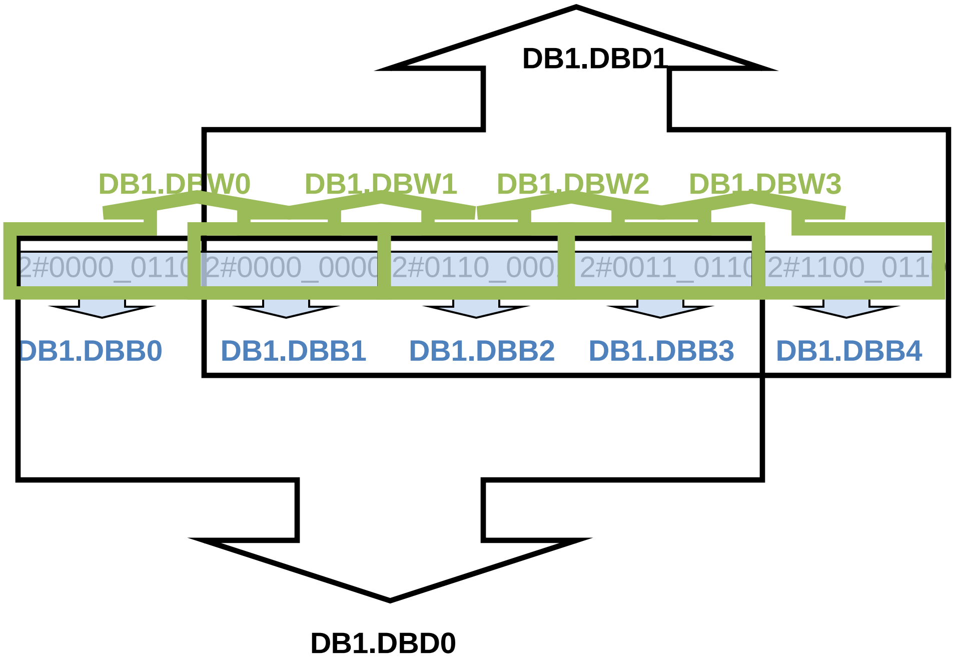
The gray area above represents the byte memory locations being overlapped when double word address is used consecutively (DB1.DBD0, DB1.DBD1, DB1.DBD2, etc).

If DB1.DBD0 is used as the first address in the Siemens processor, the next double word address that can be used without overwriting the data would be DB1.DBD4.

All of the above share the same memory locations in the processor.

Note: Incorrect memory location addressing can cause the data to be overwritten.

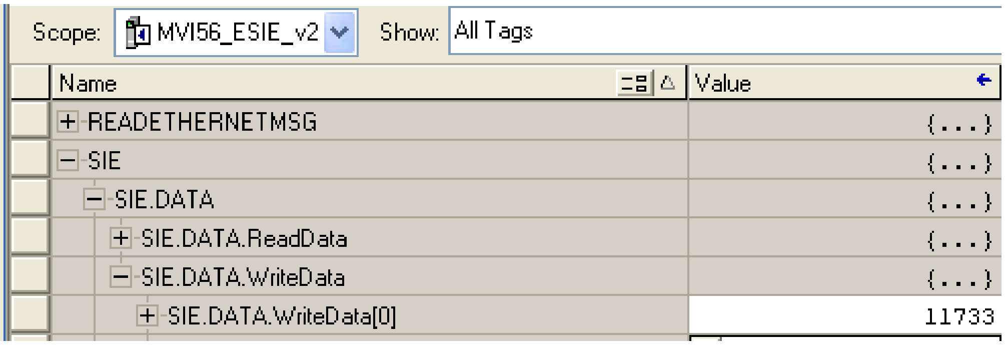
Below is a graphical representation of the addressing of the processor's memory locations.



Example:

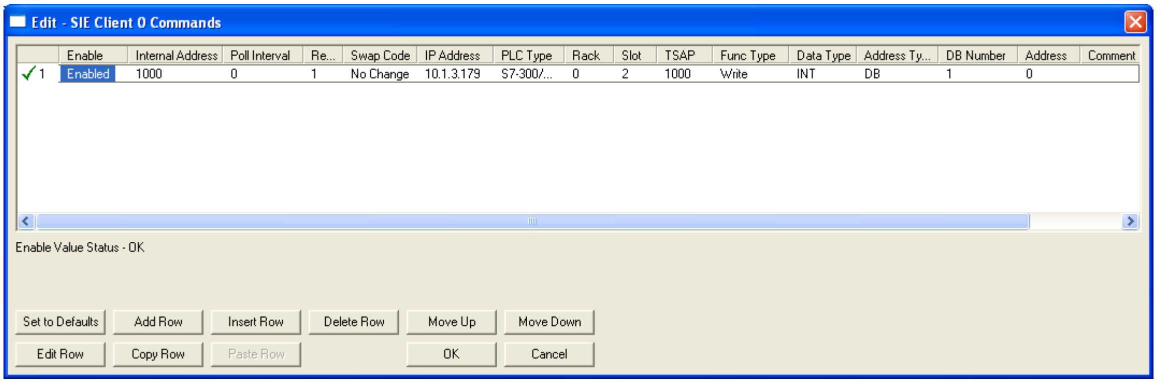
Sending an integer value of 11733 from a ControlLogix processor to a Siemens S7-300 processor demonstrates the addressing scheme in the Siemens S7-300 processor:

In RSLogix 5000:


Scope: MVI56_ESIE_v2 Show: All Tags

Name	Value
+ READETHERNETMSG	{...}
− SIE	{...}
− SIE.DATA	{...}
+ SIE.DATA.ReadData	{...}
− SIE.DATA.WriteData	{...}
+ SIE.DATA.WriteData[0]	11733

PCB screen shot showing MVI56E-SIE command to send INT data to DB1 address 0:



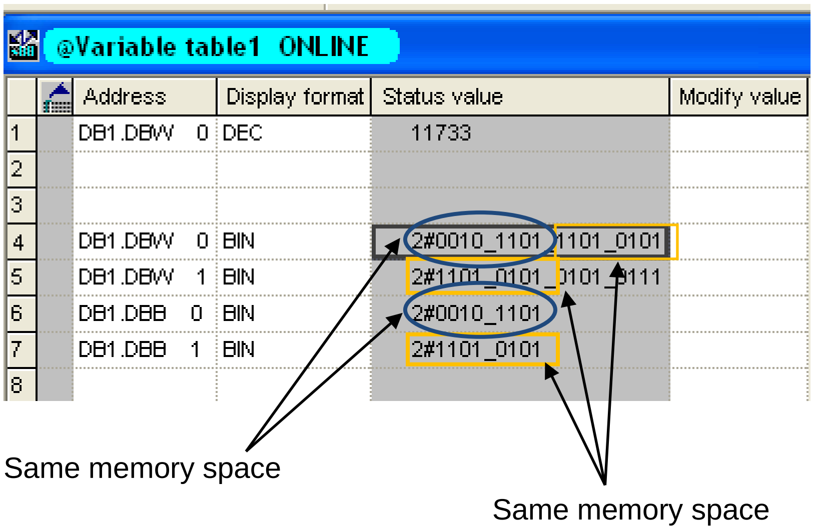
Edit - SIE Client 0 Commands

	Enable	Internal Address	Poll Interval	Re...	Swap Code	IP Address	PLC Type	Rack	Slot	TSAP	Func Type	Data Type	Address Ty...	DB Number	Address	Comment
✓ 1	Enabled	1000	0	1	No Change	10.1.3.179	S7-300/...	0	2	1000	Write	INT	DB	1	0	

Enable Value Status - OK

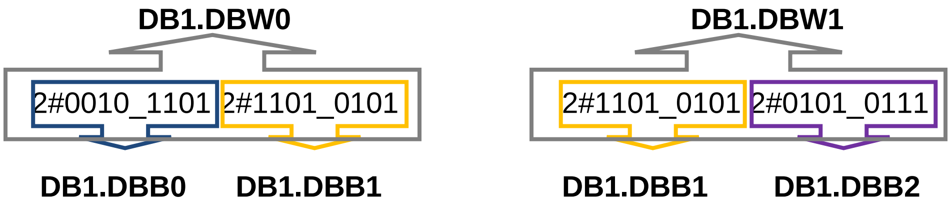
Row 1 in the SIMATICS Manager screen shot below shows the data transferred from the *SIE.DATA.WriteData[0]* controller tag in RSLogix 5000.

If the data is broken up and displayed in binary format, it can be seen that the binary data stored in the first byte of DB1.DBW 0 is identical to that stored in byte address DB1.DBB 0. This is because the memory locations referenced by the first byte of DB1.DBW 0 and by DB1.DBB0 are one and the same, as explained previously.



@Variable table1 ONLINE

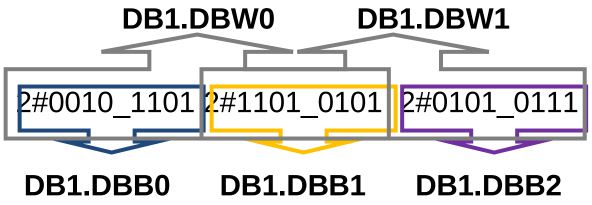
	Address	Display format	Status value	Modify value
1	DB1.DBW 0	DEC	11733	
2				
3				
4	DB1.DBW 0	BIN	2#0010_1101_1101_0101	
5	DB1.DBW 1	BIN	2#1101_0101_0101_0111	
6	DB1.DBB 0	BIN	2#0010_1101	
7	DB1.DBB 1	BIN	2#1101_0101	
8				



The first byte of DB1.DBW0 is the same as DB1.DBB0.

The second byte of DB1.DBW0 is the same as DB1.DBB1, and is the same as the first byte of DB1.DBW1.

In reality, the memory space looks like below:



Note: To access the first address of Data Block, Flag, Input, Output, Timer, and Counter memory locations in the S7-300 and S7-1200 processors, use the following address syntax.

Data Block -> DB1.DBB0, DB1.DBW0, DB1.DBD0
Flag -> MB0, MW0, MD0
Input -> IB0, IW0, ID0
Output -> QB0, QW0, QD0
Timers -> T0 – T65535
Counters -> C0 – C65535

Note: To access the first address of Data Block, Flag, Input and Output memory locations in the S7-200 processor, use the following address syntax.

Data Block -> VB0, VW0, VD0
Flag -> MB0, MW0, MD0
Input -> IB0, IW0, ID0
Output -> QB0, QW0, QD0

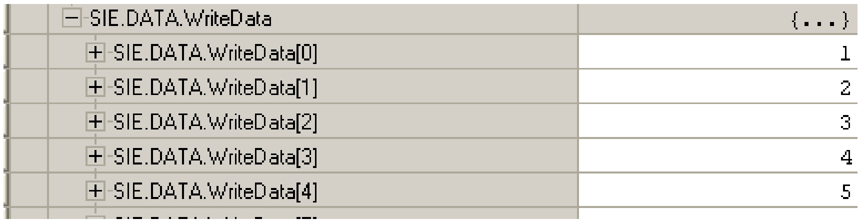
2.1.6 Configuration Examples

> **Important:** The following sections are for example purposes only, and are intended to assist a user in setting up and configuring module communications with a processor using Siemens Industrial Ethernet protocol. Before issuing control operations (write commands), please make sure that the processor is in a safe state to receive these commands. **PLEASE ENSURE THAT COMMUNICATION AND CONTROL OPERATIONS CAN BE CARRIED OUT WITHOUT DAMAGE TO PLANT AND EQUIPMENT, OR INJURY TO PERSONNEL.**

Example 1

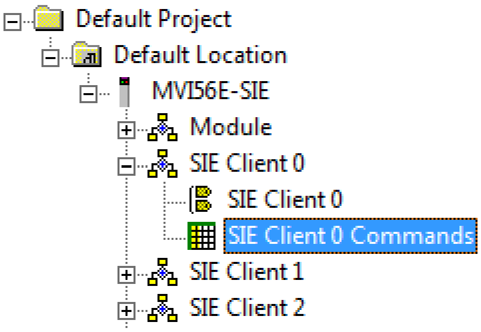
This example demonstrates configuration for data transfer from a ControlLogix processor to an S7-300 processor. The data is transferred from RSLogix controller tags *SIE.WriteData[0]* through *SIE.WriteData[4]* to Data Block 1 addresses 0 through 8 in the Siemens processor, using the INT data type.

1 Copy or move the appropriate data into controller tags *SIE.WriteData[0]* through *SIE.WriteData[4]* in RSLogix 5000.

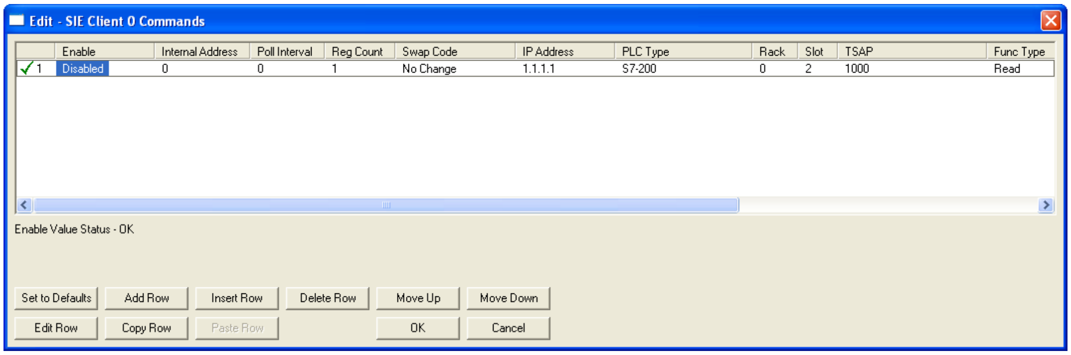


⊟ SIE.DATA.WriteData	{...}
⊞ SIE.DATA.WriteData[0]	1
⊞ SIE.DATA.WriteData[1]	2
⊞ SIE.DATA.WriteData[2]	3
⊞ SIE.DATA.WriteData[3]	4
⊞ SIE.DATA.WriteData[4]	5

2 In ProSoft Configuration Builder, expand the module tree. Double-click **SIE CLIENT 0 COMMANDS**.



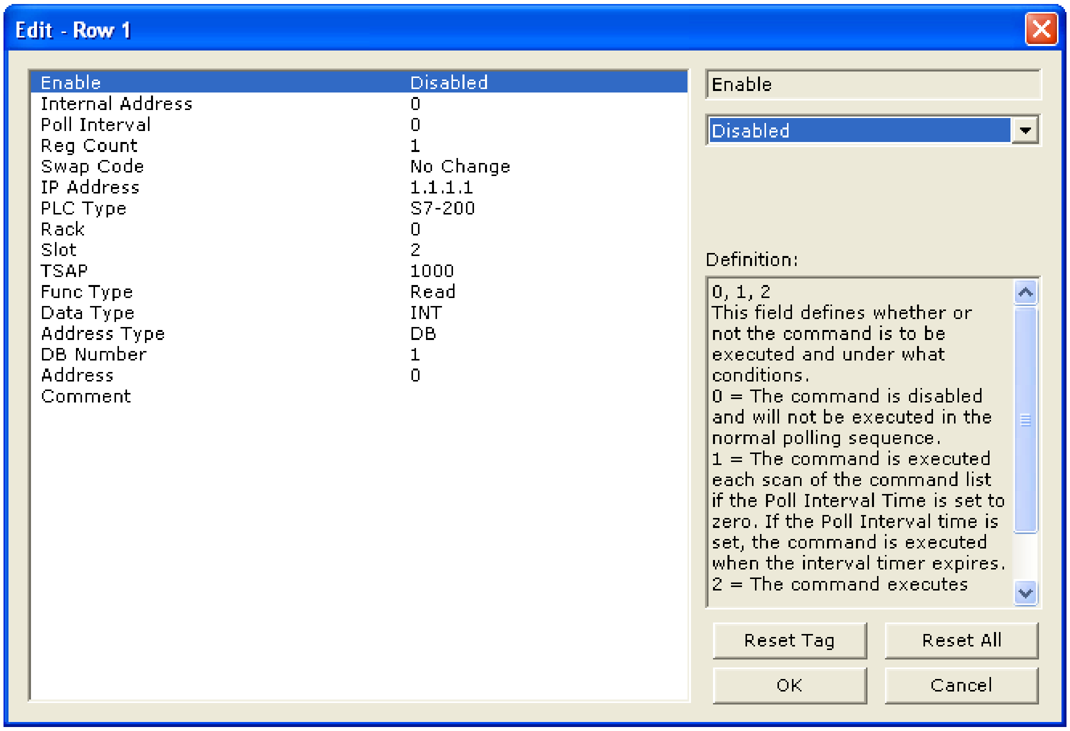
3 In the *Edit –SIE Client 0 Commands* dialog box, click the **ADD ROW** button.



	Enable	Internal Address	Poll Interval	Reg Count	Swap Code	IP Address	PLC Type	Rack	Slot	TSAP	Func Type
✓ 1	Disabled	0	0	1	No Change	1.1.1.1	S7-200	0	2	1000	Read

Enable Value Status - OK

4 Click the **EDIT ROW** button to open the *Edit – Row 1* dialog box.



Enable	Disabled
Internal Address	0
Poll Interval	0
Reg Count	1
Swap Code	No Change
IP Address	1.1.1.1
PLC Type	S7-200
Rack	0
Slot	2
TSAP	1000
Func Type	Read
Data Type	INT
Address Type	DB
DB Number	1
Address	0
Comment	

Enable

Disabled

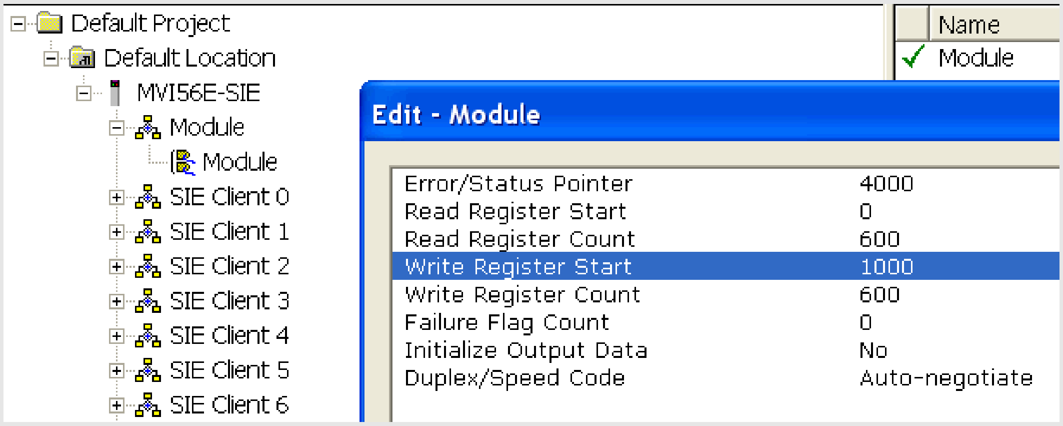
Definition:

0, 1, 2
This field defines whether or not the command is to be executed and under what conditions.
0 = The command is disabled and will not be executed in the normal polling sequence.
1 = The command is executed each scan of the command list if the Poll Interval Time is set to zero. If the Poll Interval time is set, the command is executed when the interval timer expires.
2 = The command executes

5 Select each parameter and edit its value according to the table below. (Use the IP address applicable to your network.)

Parameter	Value	Comment
Enable	Enabled	Enables the command to be executed.
Internal Address	1000	In this example, the data will be sourced from register 1000 in the module's internal database, which is by default the first register in the module's Write Data area. See note below.
Poll Interval	0	A poll interval of 0 enables the fastest polling possible.
Reg Count	5	In this example, five consecutive registers will be written, starting at Data Block 1, Address 0 (DB1.DBW0).
Swap Code	No Change	The INT data type will be used, so there is no need to consider word or byte order.
IP Address	10.1.3.179	In this example, the S7-300 Siemens processor addressed in this command is at 10.1.3.179.
PLC Type	S7-300/ S7-400/ S7-1200	This is the correct value for the S7-300 processor.
Rack	0	In this example, the rack number is 0.
Slot	2	In this example, slot 2 is being used.
TSAP	1000	This parameter is ignored in a command to an S7-300 processor, so it is left at its default value here.
Func Type	Write	In this example, data is being moved into the Siemens processor.
Data Type	INT	In this example, the INT data type is used.
Address Type	DB	In this example, the DB address type is used.
DB Number	1	In this example, DB Number 1 is used.
Address	0	In this example, the data's destination address is 0.

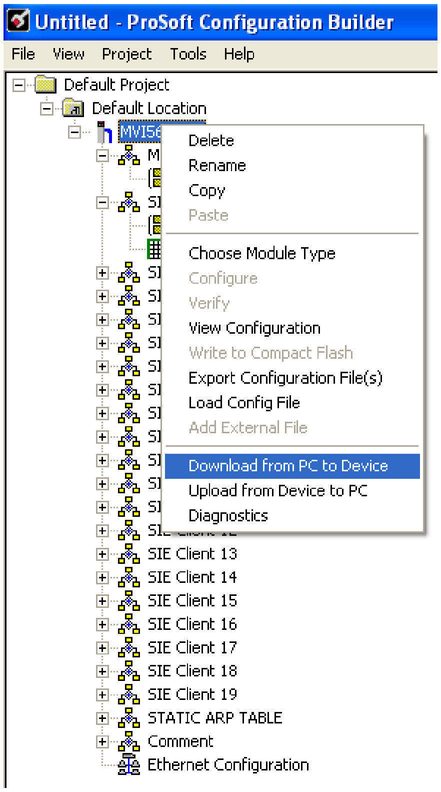
Note: In this example, the *Internal Address* parameter configures this command to source the Write data from the first register in the module's Write Data area. The start address of the module's Write Data area is determined by the *Write Register Start* parameter in the module configuration. By default, the *Write Register Start* parameter is set to 1000.



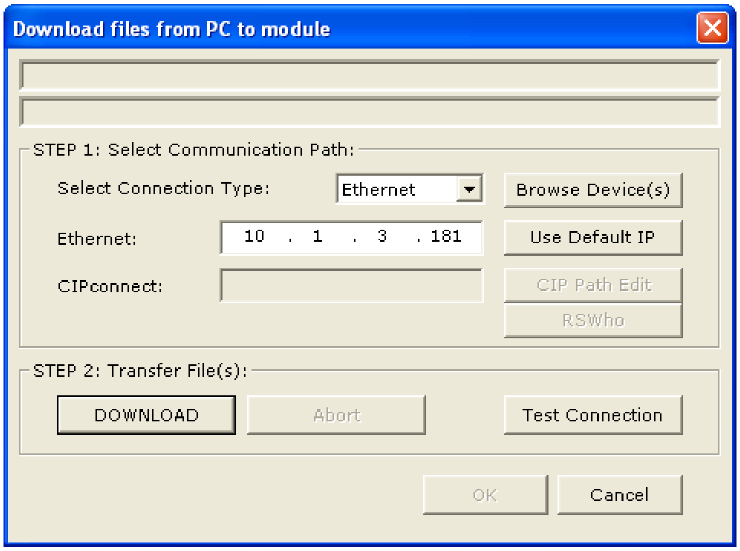
Error/Status Pointer	4000
Read Register Start	0
Read Register Count	600
Write Register Start	1000
Write Register Count	600
Failure Flag Count	0
Initialize Output Data	No
Duplex/Speed Code	Auto-negotiate

6 Click the **OK** button to save the command configuration and exit the *Edit – Row 1* dialog box. Click the **OK** button to exit the the *Edit –SIE Client 0 Commands* dialog box.

7 Right-click the **MVI56E-SIE** module icon and choose **DOWNLOAD FROM PC TO DEVICE** from the dropdown menu.



8 Click the **DOWNLOAD** button in the *Download* dialog box.

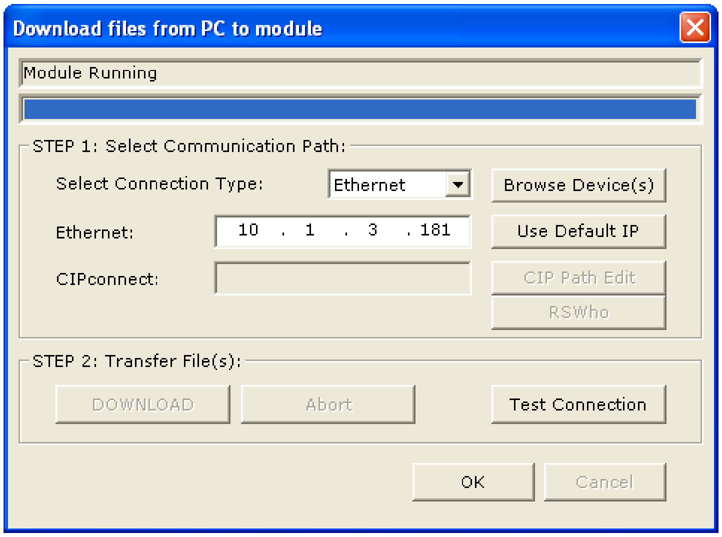

Download files from PC to module
STEP 1: Select Communication Path:
Browse Device(s)
Use Default IP
CIP Path Edit
RSWho
STEP 2: Transfer File(s):
DOWNLOAD
Abort
Test Connection
OK
Cancel

Select Connection Type: Ethernet

Ethernet: 10 . 1 . 3 . 181

CIPconnect:

9 After the download is complete, click the **OK** button.



Download files from PC to module

Module Running

STEP 1: Select Communication Path:

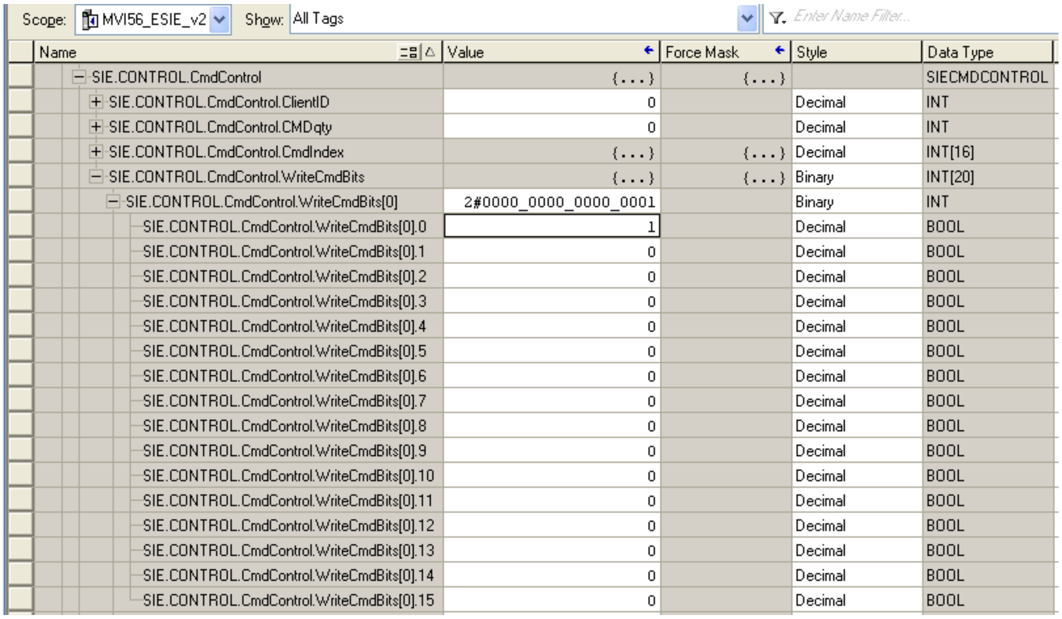
Select Connection Type: Ethernet

Ethernet: 10 . 1 . 3 . 181

CIPconnect:

STEP 2: Transfer File(s):

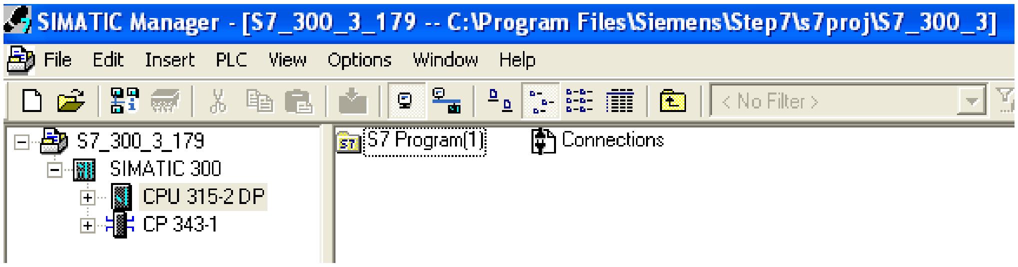
10 In the RSLogix 5000 program, expand the *SIE.CONTROL* structure. Enable the first Client 0 command by entering **1** in *SIE.CONTROL.CmdControl.WriteCmdBits[0].0*.



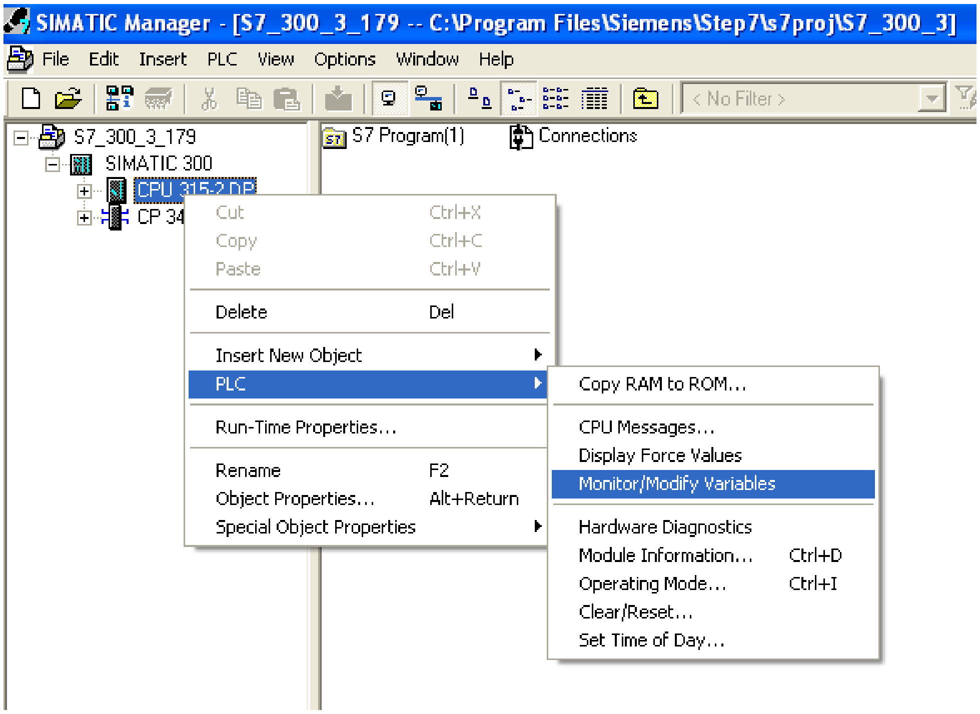
Scope: MVI56_ESIE_v2 Show: All Tags

Name	Value	Force Mask	Style	Data Type
SIE.CONTROL.CmdControl	{...}	{...}		SIECMDCONTROL
SIE.CONTROL.CmdControl.ClientID	0		Decimal	INT
SIE.CONTROL.CmdControl.CMDqty	0		Decimal	INT
SIE.CONTROL.CmdControl.CmdIndex	{...}	{...}	Decimal	INT[16]
SIE.CONTROL.CmdControl.WriteCmdBits	{...}	{...}	Binary	INT[20]
SIE.CONTROL.CmdControl.WriteCmdBits[0]	2#0000_0000_0000_0001		Binary	INT
SIE.CONTROL.CmdControl.WriteCmdBits[0].0	1		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].1	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].2	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].3	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].4	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].5	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].6	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].7	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].8	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].9	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].10	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].11	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].12	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].13	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].14	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].15	0		Decimal	BOOL

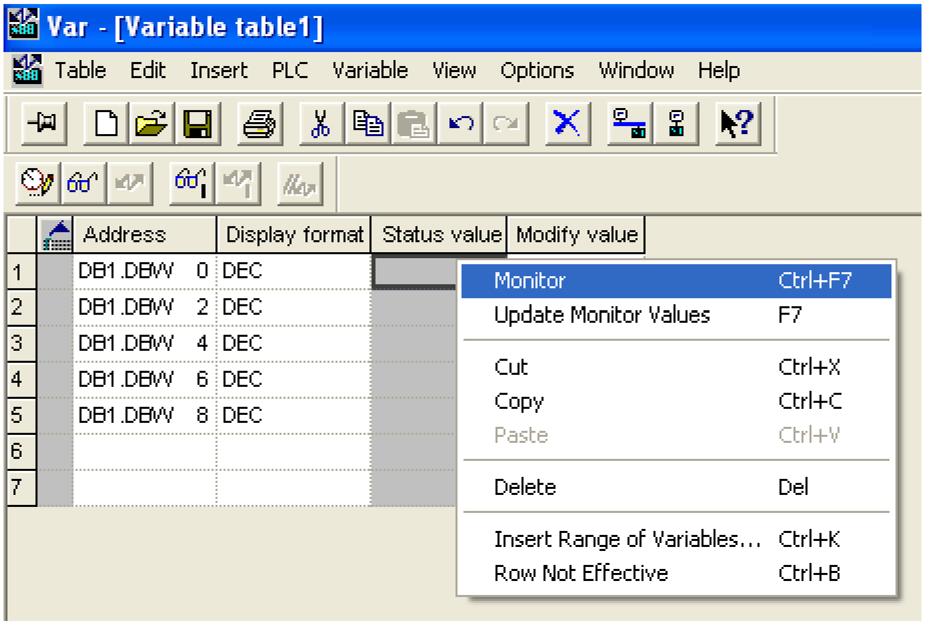
11 Check the variable table in SIMATIC Manager to see if the data transferred from the ControlLogix processor through the MVI56E-SIE module is present in the proper locations. Open SIMATIC Manager where the S7-300 processor has been set up.


SIMATIC Manager - [S7_300_3_179 -- C:\Program Files\Siemens\Step7\s7proj\S7_300_3]
File Edit Insert PLC View Options Window Help
< No Filter >
S7_300_3_179
SIMATIC 300
CPU 315-2 DP
CP 343-1
S7 Program(1)
Connections

12 Right-click the **CPU** icon, then choose **PLC -> MONITOR/MODIFY VARIABLES** .

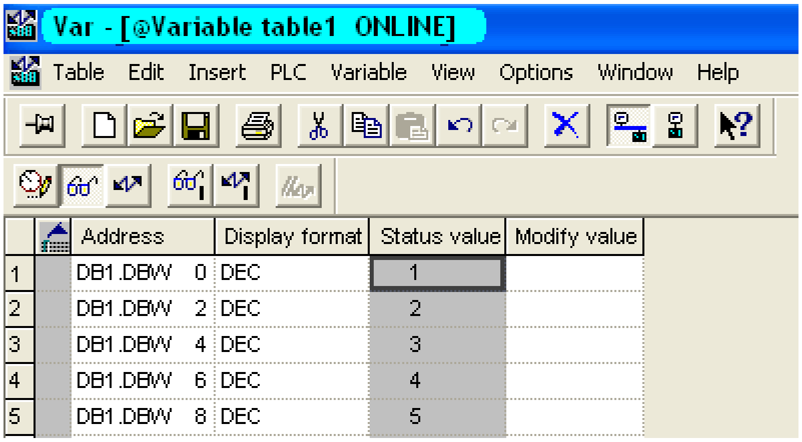


13 In the *Variable table* dialog box, type in the destination addresses as shown below. Right-click an entry in the *Status value* column and select **MONITOR** to see the variables in addresses *DB1.DBW 0* through *DB1.DBW 8*.



	Address	Display format	Status value	Modify value
1	DB1.DBW 0	DEC		
2	DB1.DBW 2	DEC		
3	DB1.DBW 4	DEC		
4	DB1.DBW 6	DEC		
5	DB1.DBW 8	DEC		
6				
7				

If the values from controller tags *SIE.WriteData[0]* through *SIE.WriteData[4]* were successfully transmitted, they will appear in the *Status value* column for addresses *DB1.DBW[0]* through *DB1.DBW[8]*.

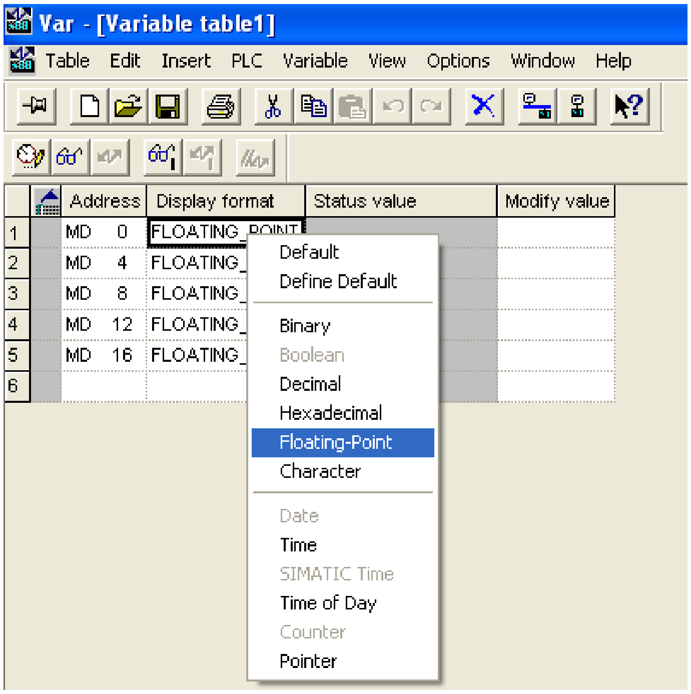

Var - [@Variable table1 ONLINE]
Table Edit Insert PLC Variable View Options Window Help

	Address	Display format	Status value	Modify value
1	DB1.DBW 0	DEC	1	
2	DB1.DBW 2	DEC	2	
3	DB1.DBW 4	DEC	3	
4	DB1.DBW 6	DEC	4	
5	DB1.DBW 8	DEC	5	

Example 2

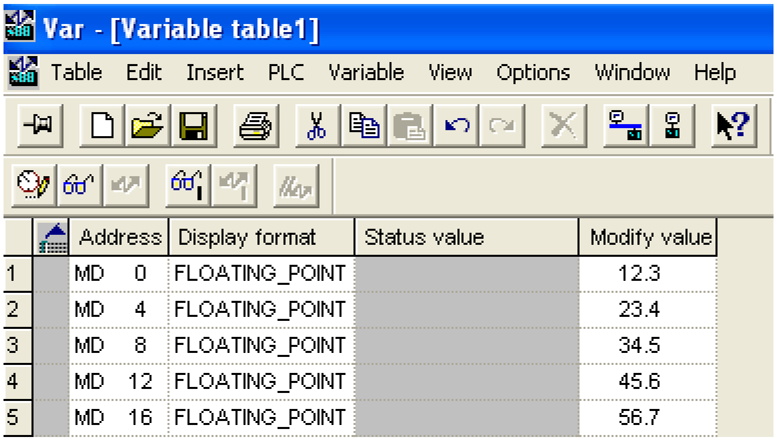
This example demonstrates data transfer from an S7-300 processor to a ControlLogix processor. The data is transferred from addresses 0 through 16 (*MD 0* through *MD 16*) to RSLogix controller tags *SIE.ReadData[0]* through *SIE.ReadData[9]*, using the REAL data type and Address Type Flag.

1 In Simatic Manager, type in Flag addresses *MD 0* through *MD 16*. Change the display format by right-clicking an entry in the *Display format* column and selecting **FLOATING-POINT** from the dropdown list.



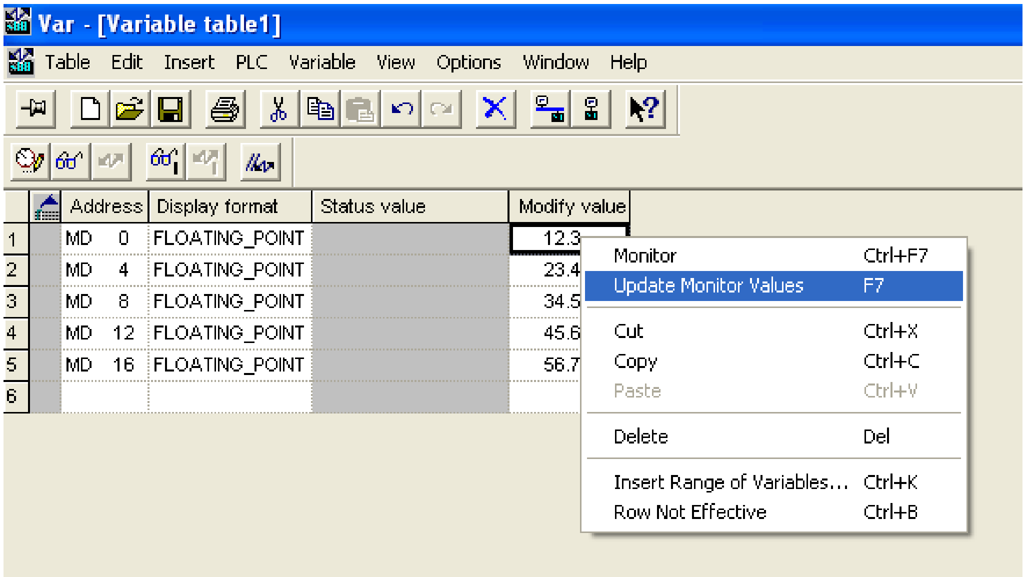
	Address	Display format	Status value	Modify value
1	MD 0	FLOATING_POINT		
2	MD 4	FLOATING_		
3	MD 8	FLOATING_		
4	MD 12	FLOATING_		
5	MD 16	FLOATING_		
6				

2 Type in the *Modify values* as shown below.



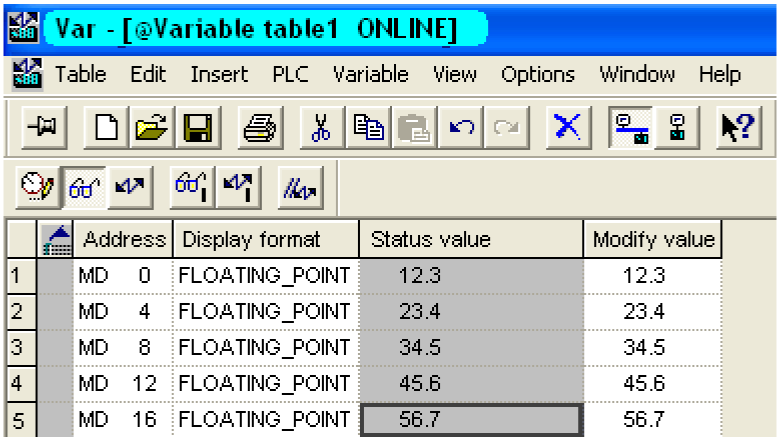
	Address	Display format	Status value	Modify value
1	MD 0	FLOATING_POINT		12.3
2	MD 4	FLOATING_POINT		23.4
3	MD 8	FLOATING_POINT		34.5
4	MD 12	FLOATING_POINT		45.6
5	MD 16	FLOATING_POINT		56.7

3 Right-click an entry in the *Modify value* column and select **UPDATE MONITOR VALUES** from the dropdown list.



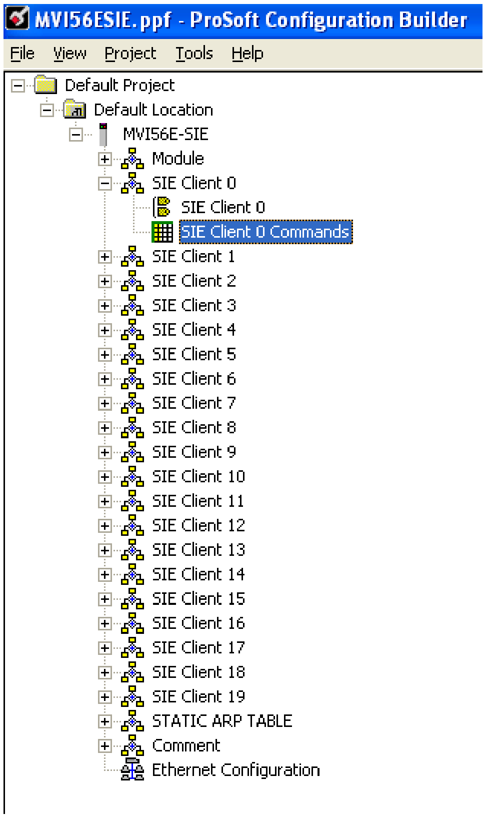
	Address	Display format	Status value	Modify value
1	MD 0	FLOATING_POINT		12.3
2	MD 4	FLOATING_POINT		23.4
3	MD 8	FLOATING_POINT		34.5
4	MD 12	FLOATING_POINT		45.6
5	MD 16	FLOATING_POINT		56.7
6				

The screen should update as shown below.

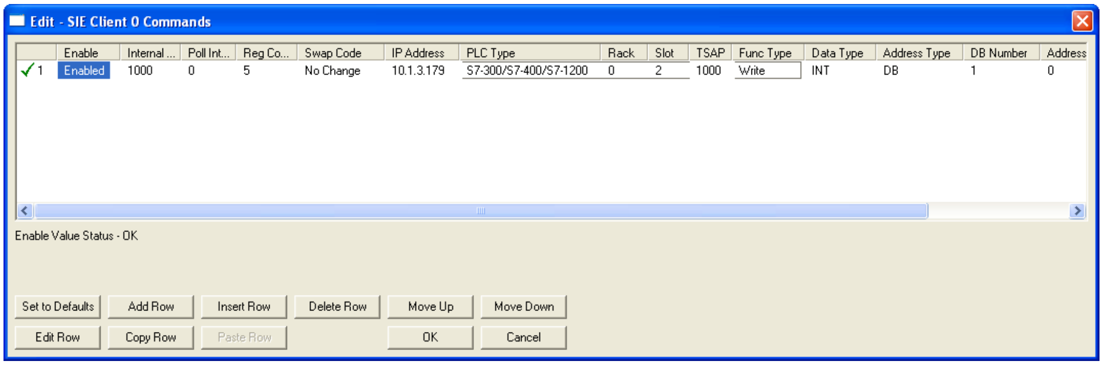

Var - [@Variable table1 ONLINE]
Table Edit Insert PLC Variable View Options Window Help

	Address	Display format	Status value	Modify value
1	MD 0	FLOATING_POINT	12.3	12.3
2	MD 4	FLOATING_POINT	23.4	23.4
3	MD 8	FLOATING_POINT	34.5	34.5
4	MD 12	FLOATING_POINT	45.6	45.6
5	MD 16	FLOATING_POINT	56.7	56.7

4 In Prosoft Configuration Builder, expand the module tree and double-click **SIE CLIENT 0 COMMANDS**.



5 In the *Edit –SIE Client 0 Commands* dialog box, click the **ADD ROW** button.

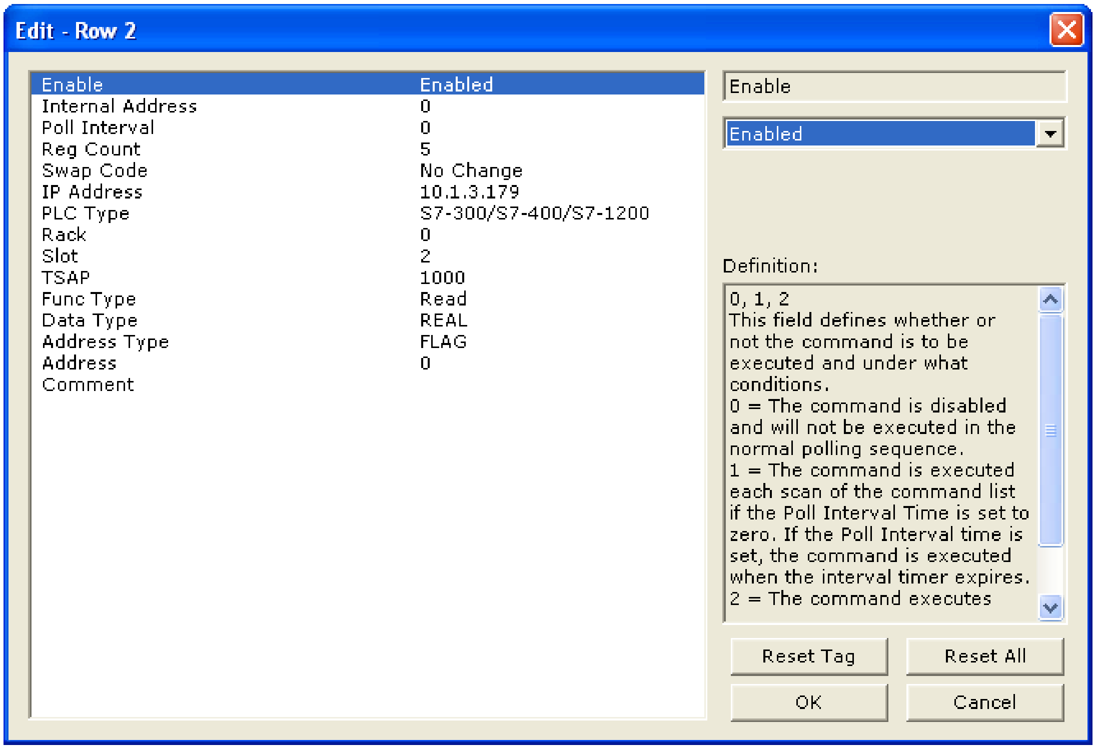


	Enable	Internal ...	Poll Int...	Reg Co...	Swap Code	IP Address	PLC Type	Rack	Slot	TSAP	Func Type	Data Type	Address Type	DB Number	Address
✓ 1	Enabled	1000	0	5	No Change	10.1.3.179	S7-300/S7-400/S7-1200	0	2	1000	Write	INT	DB	1	0

Enable Value Status - OK

6 Click the **EDIT ROW** button to open the *Edit – Row 1* dialog box.

7 Select each parameter and edit its value as shown below, except for the IP address. Use the IP address of the S7-300 processor that the command is addressing. Click the **OK** button when all the configuration parameters have been entered.



Enable	Enabled
Internal Address	0
Poll Interval	0
Reg Count	5
Swap Code	No Change
IP Address	10.1.3.179
PLC Type	S7-300/S7-400/S7-1200
Rack	0
Slot	2
TSAP	1000
Func Type	Read
Data Type	REAL
Address Type	FLAG
Address	0
Comment	

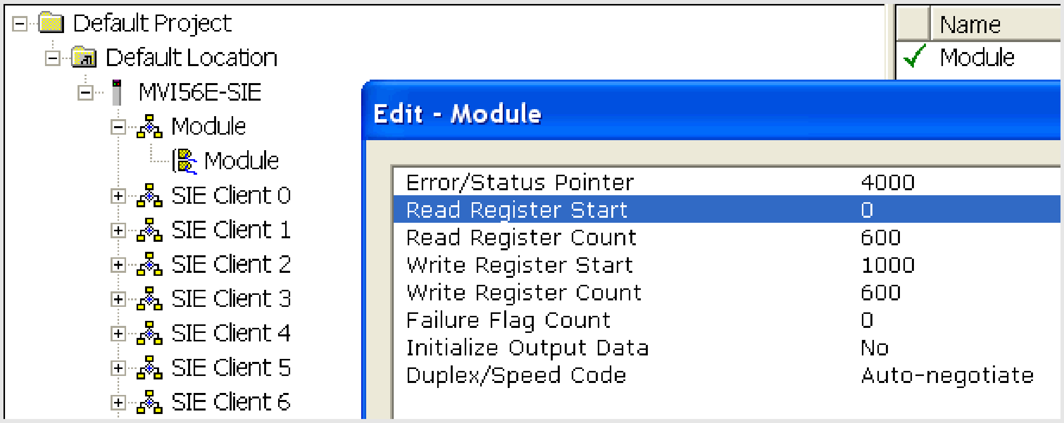
Enable

Enabled

Definition:

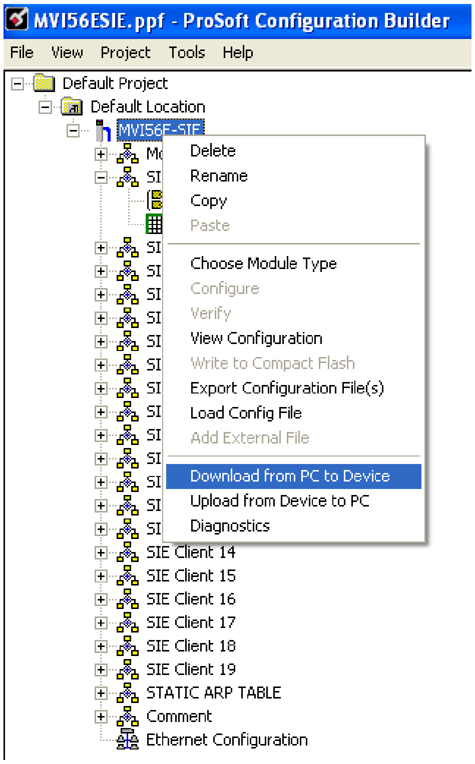
0, 1, 2
This field defines whether or not the command is to be executed and under what conditions.
0 = The command is disabled and will not be executed in the normal polling sequence.
1 = The command is executed each scan of the command list if the Poll Interval Time is set to zero. If the Poll Interval time is set, the command is executed when the interval timer expires.
2 = The command executes

Note: In this example, the *Internal Address* parameter configures this command to place the data received from S7-300 processor into the first register of the module's Read Data area. The start address of the module's Read Data area is determined by the *Read Register Start* parameter in the module configuration. By default, the *Read Register Start* parameter is set to 0.

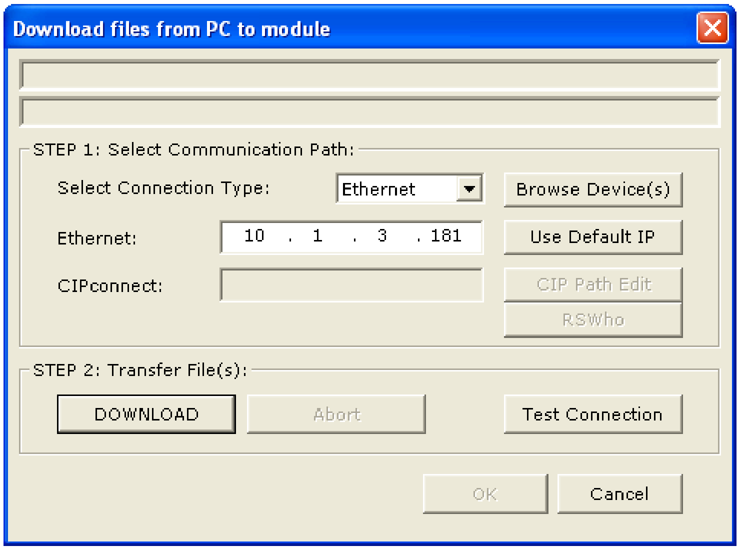


Error/Status Pointer	4000
Read Register Start	0
Read Register Count	600
Write Register Start	1000
Write Register Count	600
Failure Flag Count	0
Initialize Output Data	No
Duplex/Speed Code	Auto-negotiate

8 Right-click the **MVI56E-SIE** module icon and select **DOWNLOAD FROM PC TO DEVICE** from the dropdown list.



9 Click the **DOWNLOAD** button in the *Download* dialog box.


Download files from PC to module

STEP 1: Select Communication Path:

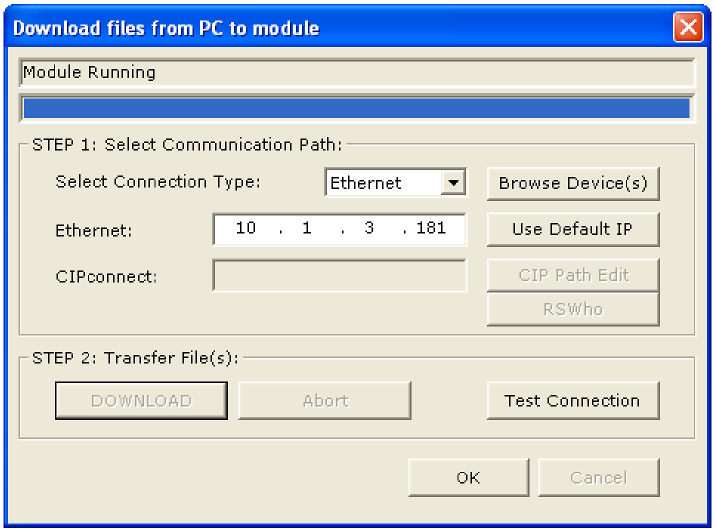
Select Connection Type: Ethernet

Ethernet: 10 . 1 . 3 . 181

CIPconnect:

STEP 2: Transfer File(s):

10 When the download is complete, click the **OK** button.



Download files from PC to module

Module Running

STEP 1: Select Communication Path:

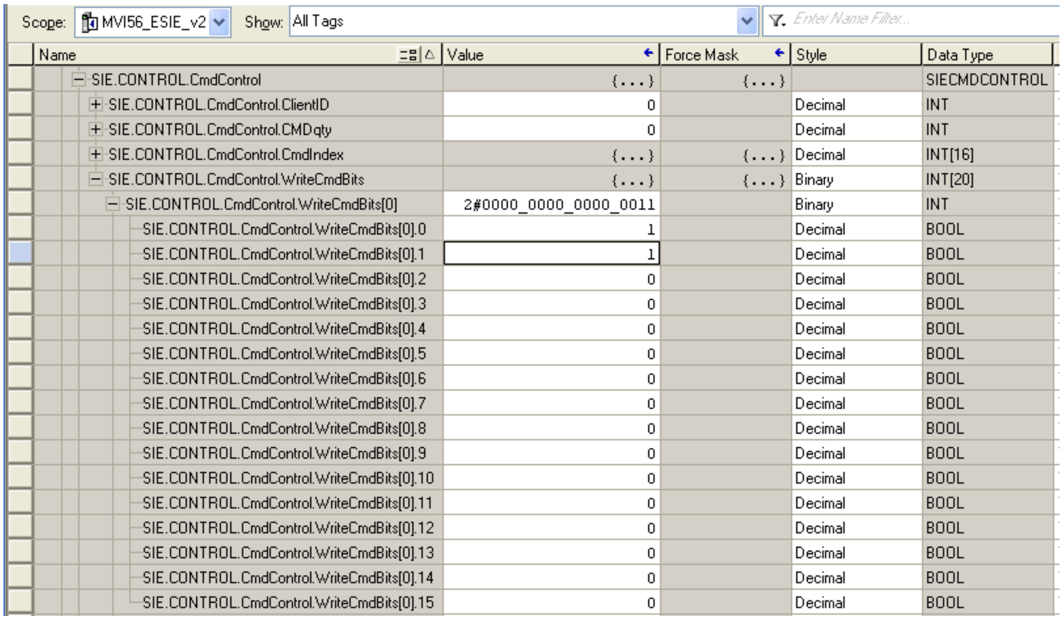
Select Connection Type: Ethernet

Ethernet: 10 . 1 . 3 . 181

CIPconnect:

STEP 2: Transfer File(s):

11 In the RSLogix 5000 program, expand the *SIE.CONTROL* structure. Assuming this is the second command in the Command List, enable the second Client 0 command by entering **1** in *SIE.CONTROL.CmdControl.WriteCmdBits[0].1.*

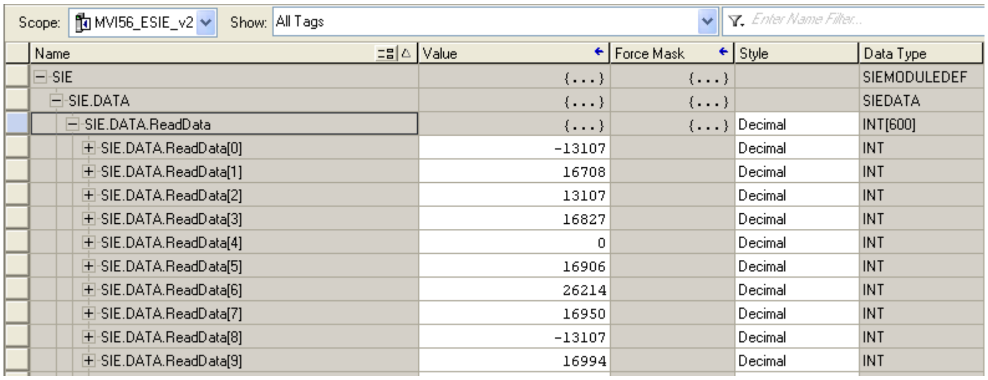


Scope: MVI56_ESIE_v2 Show: All Tags

Name	Value	Force Mask	Style	Data Type
SIE.CONTROL.CmdControl	{...}	{...}		SIECMDCONTROL
SIE.CONTROL.CmdControl.ClientID	0		Decimal	INT
SIE.CONTROL.CmdControl.CMDqty	0		Decimal	INT
SIE.CONTROL.CmdControl.CmdIndex	{...}	{...}	Decimal	INT[16]
SIE.CONTROL.CmdControl.WriteCmdBits	{...}	{...}	Binary	INT[20]
SIE.CONTROL.CmdControl.WriteCmdBits[0]	2#0000_0000_0000_0011		Binary	INT
SIE.CONTROL.CmdControl.WriteCmdBits[0].0	1		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].1	1		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].2	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].3	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].4	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].5	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].6	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].7	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].8	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].9	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].10	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].11	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].12	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].13	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].14	0		Decimal	BOOL
SIE.CONTROL.CmdControl.WriteCmdBits[0].15	0		Decimal	BOOL

12 Make sure that RSLogix is online with the ControlLogix processor. Check to see if the data read from the S7-300 processor appears in controller tags *SIE.DATA.ReadData[0]* through *SIE.DATA.ReadData[9]*.

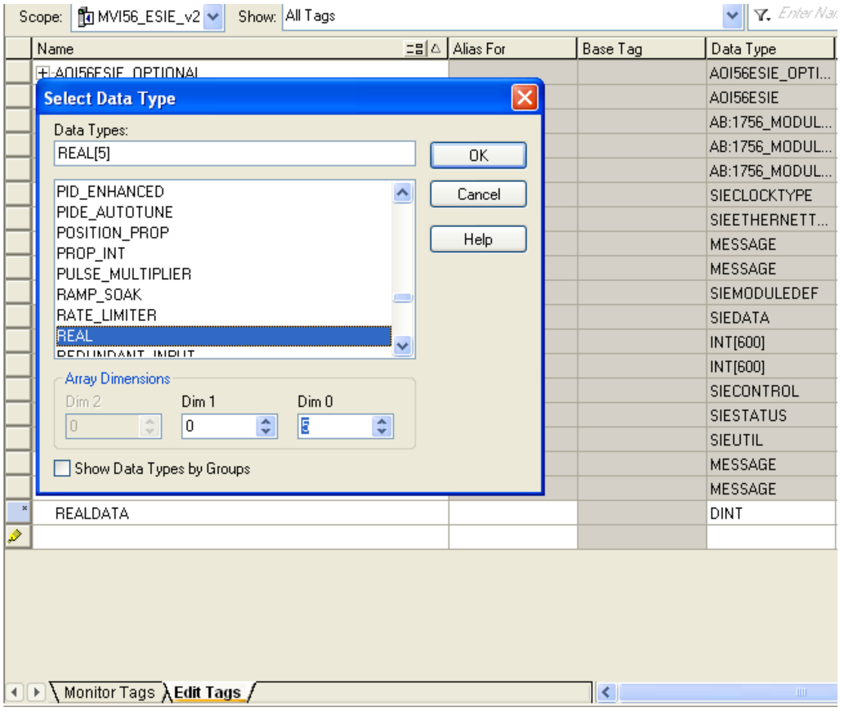
> **Note:** Since *SIE.DATA.ReadData* is an array of integers, the data will not appear in floating-point format. We will format the data in the following steps.



Scope: MVI56_ESIE_v2 Show: All Tags

Name	Value	Force Mask	Style	Data Type
SIE	{...}	{...}		SIEMODULEDEF
SIE.DATA	{...}	{...}		SIEDATA
SIE.DATA.ReadData	{...}	{...}	Decimal	INT[600]
SIE.DATA.ReadData[0]	-13107		Decimal	INT
SIE.DATA.ReadData[1]	16708		Decimal	INT
SIE.DATA.ReadData[2]	13107		Decimal	INT
SIE.DATA.ReadData[3]	16827		Decimal	INT
SIE.DATA.ReadData[4]	0		Decimal	INT
SIE.DATA.ReadData[5]	16906		Decimal	INT
SIE.DATA.ReadData[6]	26214		Decimal	INT
SIE.DATA.ReadData[7]	16950		Decimal	INT
SIE.DATA.ReadData[8]	-13107		Decimal	INT
SIE.DATA.ReadData[9]	16994		Decimal	INT

13 To display the data in floating-point format, copy it into an array with a REAL data type. Click the **EDIT TAGS** tab at the bottom of the *Controller Tags* window and enter a name for the array (in this example, **REALDATA**). In the *Data Type* column, enter **REAL[5]**.



Scope: MVI56_ESIE_v2 Show: All Tags

Name	Alias For	Base Tag	Data Type
AOI56ESIE_OPTIONAL			AOI56ESIE_OPTI...
			AOI56ESIE
			AB:1756_MODUL...
			AB:1756_MODUL...
			AB:1756_MODUL...
			SIECLOCKTYPE
			SIEETHERNETT...
			MESSAGE
			MESSAGE
			SIEMODULEDEF
			SIEDATA
			INT[600]
			INT[600]
			SIECONTROL
			SIESTATUS
			SIEUTIL
			MESSAGE
			MESSAGE
REALDATA			DINT

Select Data Type

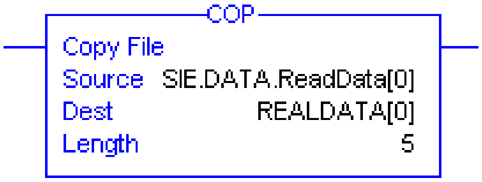
Data Types: REAL[5]

PID_ENHANCED
PIDE_AUTOTUNE
POSITION_PROP
PROP_INT
PULSE_MULTIPLIER
RAMP_SOAK
RATE_LIMITER
REAL
REDUNDANT_INPUT

Array Dimensions: Dim 2: 0 Dim 1: 0 Dim 0: 5

□ Show Data Types by Groups

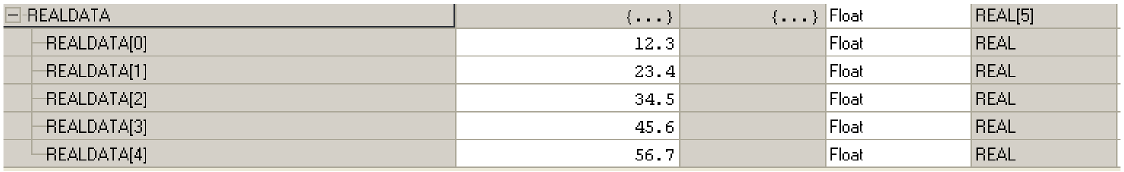
14 Add a copy instruction to the *MainRoutine* ladder to copy the 10 words from *SIE.DATA.ReadData* into the REALDATA array created in the previous step.



COP
Copy File
Source SIE.DATA.ReadData[0]
Dest REALDATA[0]
Length 5

Note: The length in the COP instruction is the length of the destination. Since 10 INTs (words) become 5 REALS (floating-point numbers), the copy length is 5.

15 Verify that the data transferred from S7-300 processor appears correctly in the *REALDATA[0]* through *REALDATA[4]* controller tags.



⊟ REALDATA	{...}	{...}	Float	REAL[5]
REALDATA[0]	12.3		Float	REAL
REALDATA[1]	23.4		Float	REAL
REALDATA[2]	34.5		Float	REAL
REALDATA[3]	45.6		Float	REAL
REALDATA[4]	56.7		Float	REAL

2.1.7 Static ARP Table

The Static ARP Table defines a list of static IP addresses that the module will use when an ARP (Address Resolution Protocol) is required. The module will accept up to 40 static IP/MAC address data sets.

Use the Static ARP table to reduce the amount of network traffic by specifying IP addresses and their associated MAC (hardware) addresses that the MVI56E-SIE module will be communicating with regularly.

Important: If the device in the field is changed, this table must be updated to contain the new MAC address for the device and downloaded to the module. If the MAC is not changed, no communications with the module will be provided.

IP Address

Dotted notation

This table contains a list of static IP addresses that the module will use when an ARP is required. The module will accept up to 40 static IP/MAC address data sets.

Important: If the device in the field is changed, this table must be updated to contain the new MAC address for the device and downloaded to the module. If the MAC is not changed, no communications with the module will occur.

Hardware MAC Address

Hex value

This table contains a list of static MAC addresses that the module will use when an ARP is required. The module will accept up to 40 static IP/MAC address data sets.

Important: If the device in the field is changed, this table must be updated to contain the new MAC address for the device and downloaded to the module. If the MAC is not changed, no communications with the module will occur.

2.1.8 Ethernet Configuration

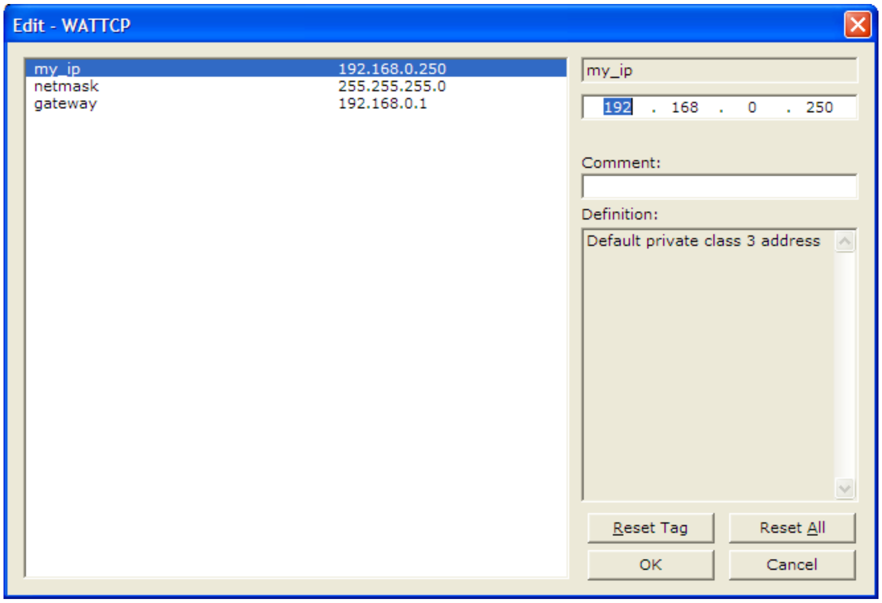
Use this procedure to configure the Ethernet settings for the module. An IP address, subnet mask and gateway address must be assigned. After this step is completed, the module can be connected with an Ethernet cable.

1 Determine the network settings for the module, with the help of the network administrator if necessary. The following information will be needed:

- IP address (fixed IP required) _____ . _____ . _____ . _____
- Subnet mask _____ . _____ . _____ . _____
- Gateway address _____ . _____ . _____ . _____

Note: The gateway address is optional, and is not required for networks that do not use a default gateway.

2 Double-click the **ETHERNET CONFIGURATION** icon. This action opens the *Edit* dialog box.



Edit - WATTCP

my_ip	192.168.0.250
netmask	255.255.255.0
gateway	192.168.0.1

my_ip

192 . 168 . 0 . 250

Comment:

Definition:

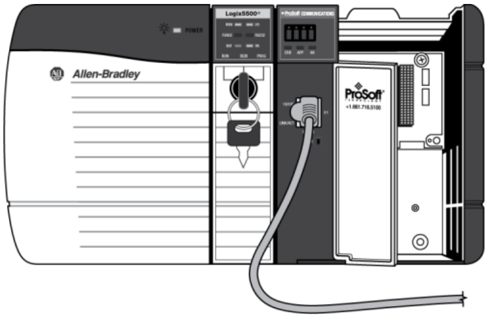
Default private class 3 address

3 Edit the values for *my_ip*, *netmask* (subnet mask) and *gateway* (default gateway).

4 When finished editing, click **OK** to save the changes and return to the *ProSoft Configuration Builder* window.

2.2 Connecting the PC to the Module

With the module securely mounted, connect one end of the Ethernet cable to the *Config (E1)* Port, and the other end to an Ethernet hub or switch accessible from the same network as the PC. A connection can lso be established directly from the Ethernet Port on the PC to the *Config (E1)* Port on the module by using an Ethernet crossover cable (not included).

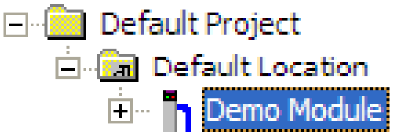


2.2.1 Setting Up a Temporary IP Address

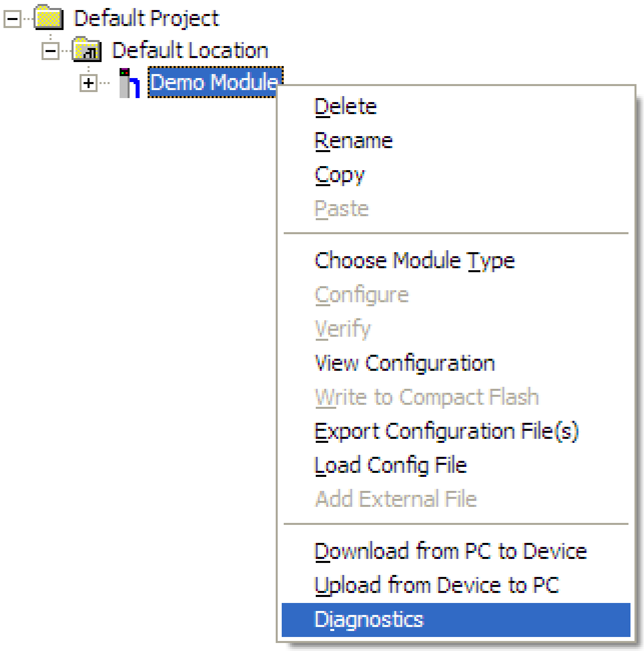
Important: *ProSoft Configuration Builder* locates MVI56E-SIE modules through UDP broadcast messages. These messages may be blocked by routers or layer 3 switches. In that case, *ProSoft Discovery Service* will be unable to locate the modules.

To use *ProSoft Configuration Builder*, arrange the Ethernet connection so that there is no router/ layer 3 switch between the computer and the module OR reconfigure the router/ layer 3 switch to allow routing of the UDP broadcast messages.

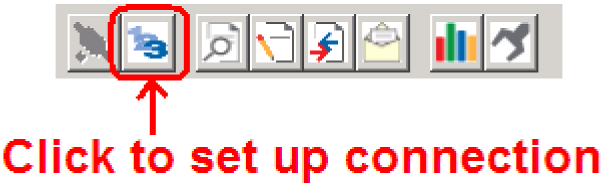
1 In the tree view in *ProSoft Configuration Builder*, select the **MVI56E-SIE** module.



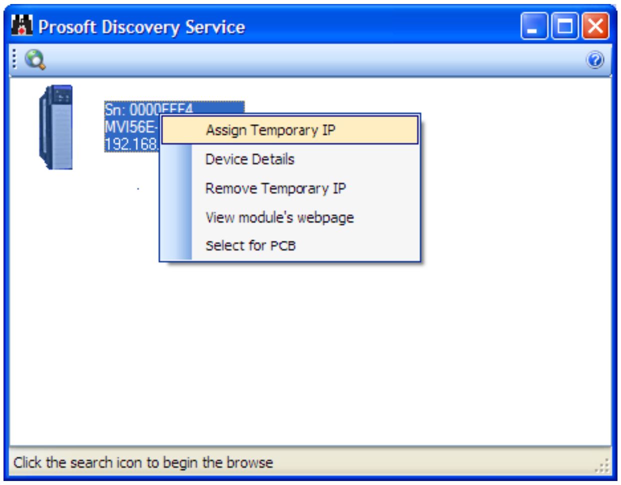
2 Click the right mouse button to open a shortcut menu. On the shortcut menu, choose **DIAGNOSTICS.**



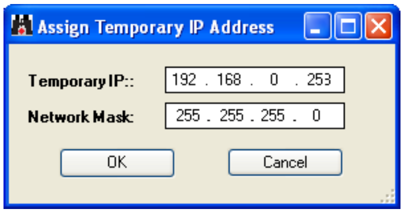
3 In the *Diagnostics* window, click the **SET UP CONNECTION** button.



4 In the *Connection Setup* dialog box, click the **BROWSE DEVICE(S)** button to open the *ProSoft Discovery Service*. Select the module, then right-click and choose **ASSIGN TEMPORARY IP**.



5 The module's default IP address is 192.168.0.250. Choose an unused IP within the same subnet, and then click **OK**.

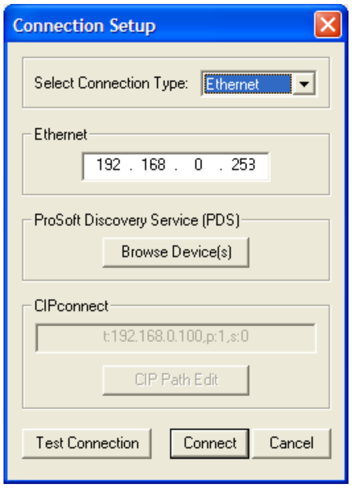

Assign Temporary IP Address

Temporary IP:: 192 . 168 . 0 . 253

Network Mask: 255 . 255 . 255 . 0

Important: The temporary IP address is only valid until the next time the module is initialized. For information on how to set the module's permanent IP address, see Ethernet Configuration (page 78).

6 Close the *ProSoft Discovery Service* window. Enter the temporary IP in the Ethernet address field of the *Connection Setup* dialog box, then click the **TEST CONNECTION** button to verify that the module is accessible with the current settings.


Connection Setup

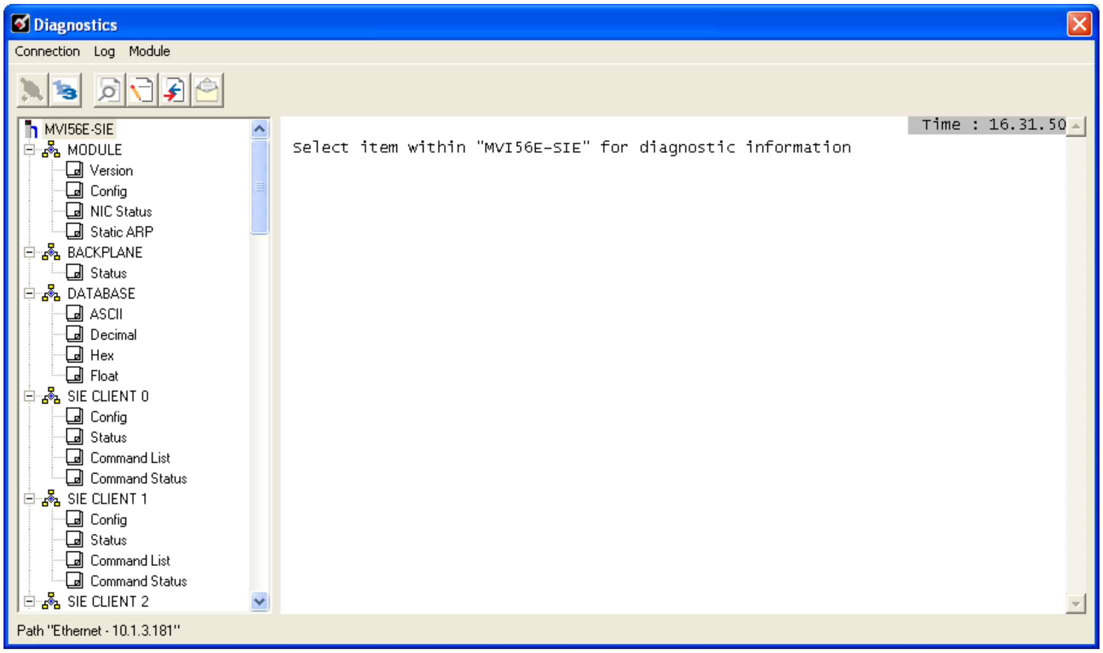
Select Connection Type: Ethernet

Ethernet: 192 . 168 . 0 . 253

ProSoft Discovery Service (PDS)

CIPconnect: t:192.168.0.100,p:1,s:0

7 If the *Test Connection* is successful, click **CONNECT**. The *Diagnostics* menu will display in the *Diagnostics* window.



```
Time : 16.31.50
Select item within "MVI56E-SIE" for diagnostic information
```

Path "Ethernet - 10.1.3.181"

2.3 Downloading the Project to the Module

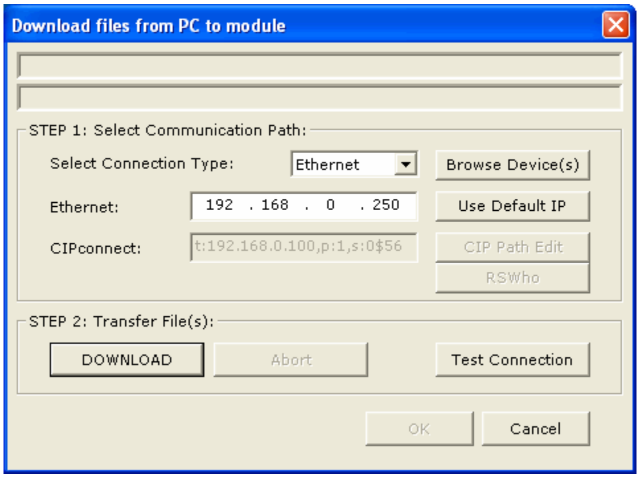
Note: For alternative methods of connecting to the module with your PC, refer to Using CIPconnect to Connect to the Module (page 84) or Using RSWho to Connect to the Module (page 93).

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

1 In the tree view in *ProSoft Configuration Builder*, click once to select the **MVI56E-SIE** module.

2 Open the **PROJECT** menu, and then choose **MODULE / DOWNLOAD.**

This action opens the *Download* dialog box. Notice that the Ethernet address field contains the temporary IP address you assigned previously. *ProSoft Configuration Builder* will use this temporary IP address to connect to the module.


Download files from PC to module
STEP 1: Select Communication Path:
Browse Device(s)
Use Default IP
CIP Path Edit
RSWho
STEP 2: Transfer File(s):
DOWNLOAD
Abort
Test Connection
OK
Cancel

Select Connection Type: Ethernet

Ethernet: 192 . 168 . 0 . 250

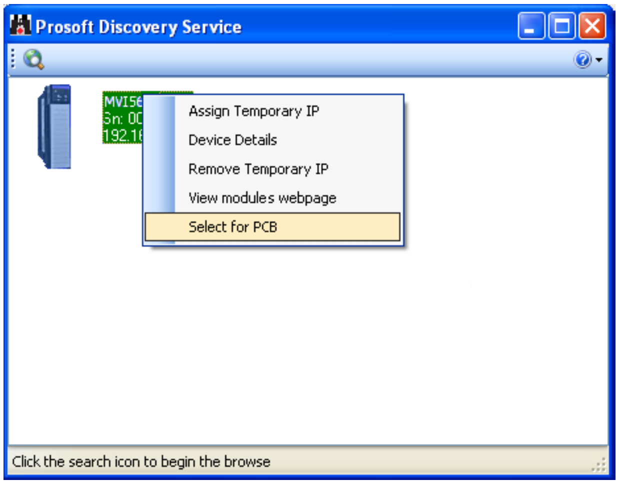
CIPconnect: t:192.168.0.100,p:1,s:0$56

Click **TEST CONNECTION** to verify that the IP address allows access to the module.

3 If the connection succeeds, click **DOWNLOAD** to transfer the Ethernet configuration to the module.

If the *Test Connection* procedure fails, an error message will appear. To correct the error, follow these steps:

1 Click **OK** to dismiss the error message.
2 In the *Download* dialog box, click **BROWSE DEVICE(S)** to open *ProSoft Discovery Service*.



3 Select the module, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **SELECT FOR PCB**.
4 Close *ProSoft Discovery Service*.
5 Click **DOWNLOAD** to transfer the configuration to the module.

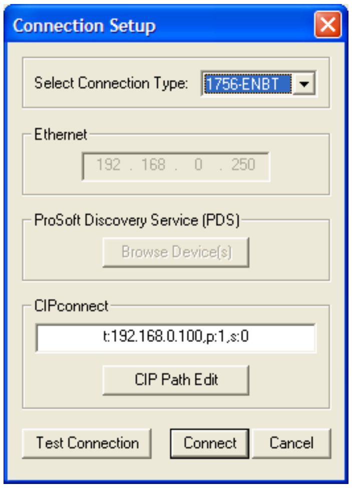
2.3.1 Using CIPconnect to Connect to the Module

CIPconnect® can be used to connect a PC to the MVI56E-SIE module over Ethernet using Rockwell Automation's 1756-ENBT EtherNet/IP® module. This allows for configuringof the MVI56E-SIE module and network, uploading and downloading of files, and viewing of network and module diagnostics from a PC. RSLinx is not required when using CIPconnect. All that is needed:

- The IP addresses and slot numbers of any 1756-ENBT modules in the path
- The ControlNet node numbers and slot numbers of any 1756-CNBx ControlNet Bridge modules in the path
- The slot number of the MVI56E-SIE in the destination ControlLogix chassis (the last ENBT/CNBx and chassis in the path).

To use CIPconnect, follow these steps.

1 In the *Select Connection Type* dropdown list, choose **1756-ENBT**. The default path appears in the text box, as shown in the following illustration.



Connection Setup

Select Connection Type: 1756-ENBT

Ethernet

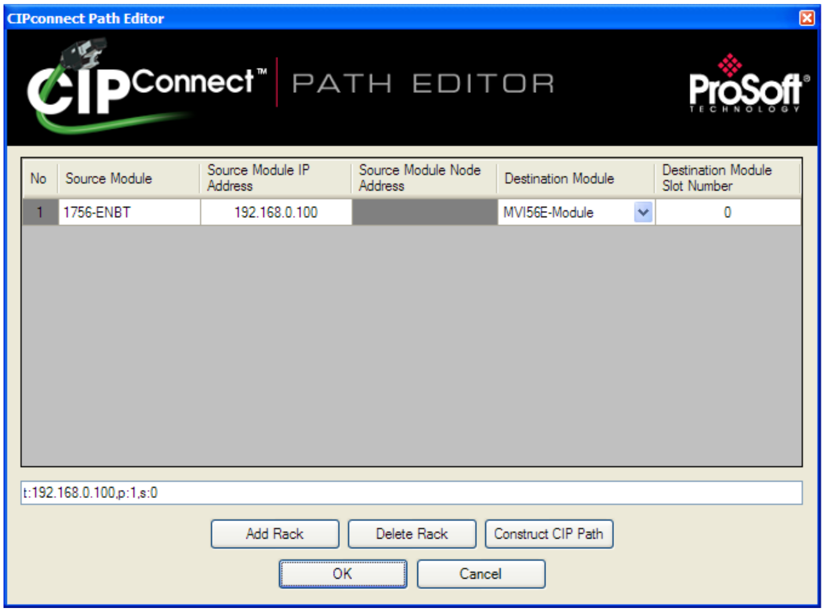
192 . 168 . 0 . 250

ProSoft Discovery Service (PDS)

CIPconnect

t:192.168.0.100,p:1,s:0

2 Click **CIP PATH EDIT** to open the *CIPconnect Path Editor* dialog box.



CIPconnect Path Editor

No	Source Module	Source Module IP Address	Source Module Node Address	Destination Module	Destination Module Slot Number
1	1756-ENBT	192.168.0.100		MVI56E-Module	0

t:192.168.0.100,p:1,s:0

The *CIPconnect Path Editor* allows the path between the PC and the MVI56E-SIE module to be defined. The first connection from the PC is always a 1756-ENBT (Ethernet/IP) module.

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

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

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

To use the CIPconnect Path Editor, follow these steps.

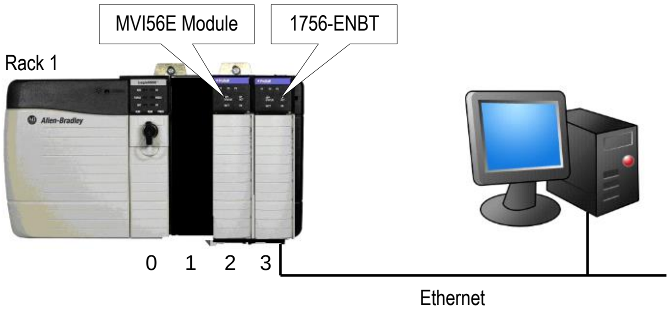
1 Configure the path between the 1756-ENBT connected to the PC and the MVI56E-SIE module.
  - If the module is located in a remote rack, add more racks to configure the full path.
  - The path can only contain ControlNet or Ethernet/IP networks.
  - The maximum number of supported racks is six.

2 Click **CONSTRUCT CIP PATH** to build the path in text format

3 Click **OK** to confirm the configured path.

Example 1: Local Rack Application

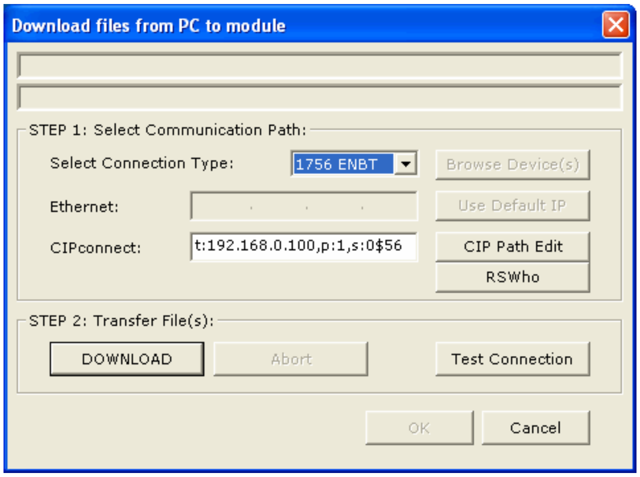
For this example, the MVI56E-SIE module is located in the same rack as the 1756-ENBT that is connected to the PC.



Rack 1

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

1 In the *Download* dialog box, click **CIP PATH EDIT.**


Download files from PC to module

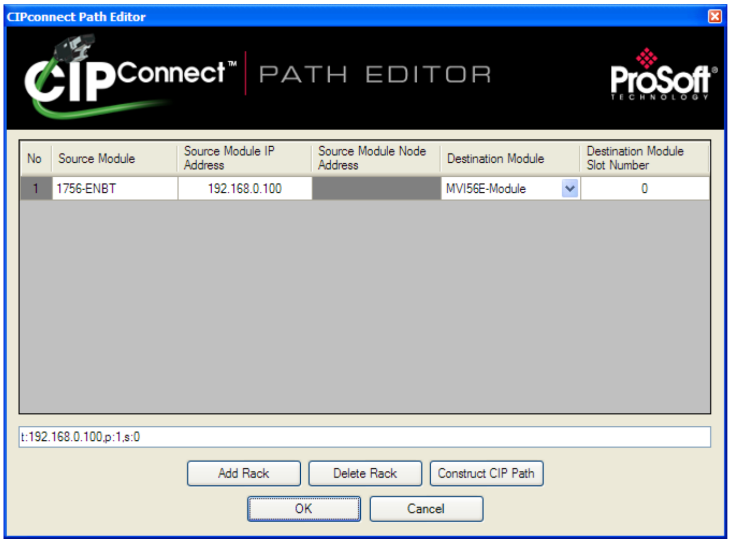
STEP 1: Select Communication Path:

Select Connection Type: 1756 ENBT

Ethernet:

CIPconnect: t:192.168.0.100,p:1,s:0$56

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

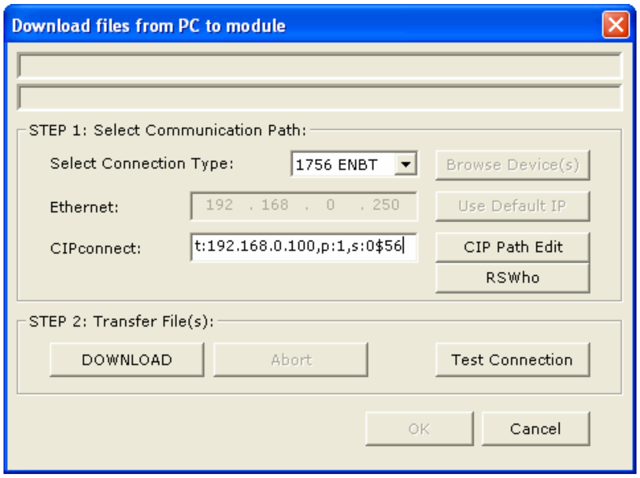


No	Source Module	Source Module IP Address	Source Module Node Address	Destination Module	Destination Module Slot Number
1	1756-ENBT	192.168.0.100		MVI56E-Module	0

t:192.168.0.100,p:1,s:0

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

3 Check the new path in the *Download* dialog box.



STEP 1: Select Communication Path:

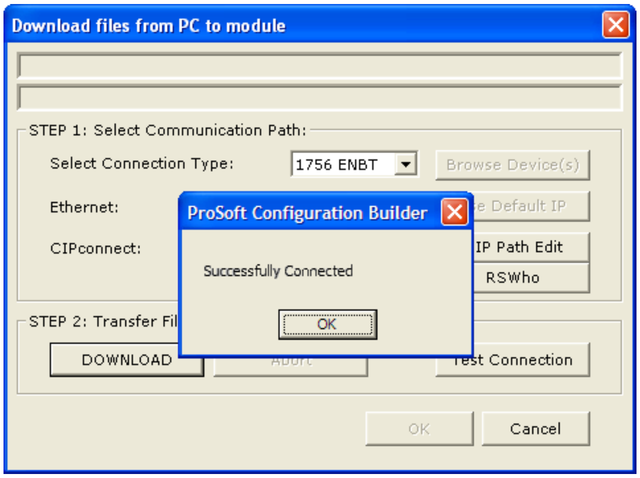
Select Connection Type: 1756 ENBT

Ethernet: 192 . 168 . 0 . 250

CIPconnect: t:192.168.0.100,p:1,s:0$56

STEP 2: Transfer File(s):

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



Download files from PC to module

STEP 1: Select Communication Path:

Select Connection Type: 1756 ENBT

Ethernet:

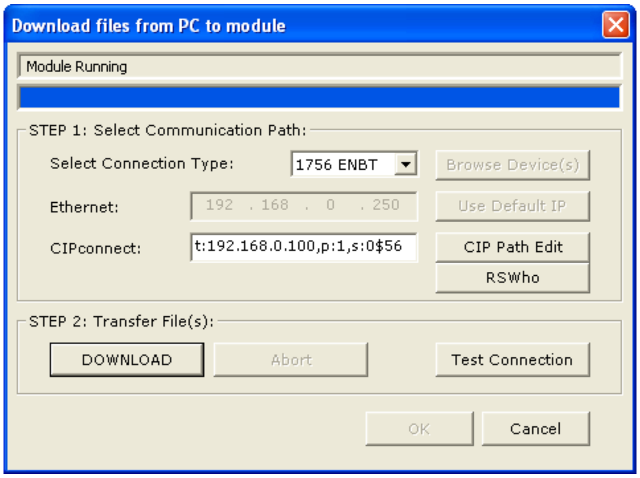
CIPconnect:

STEP 2: Transfer Fil

ProSoft Configuration Builder

Successfully Connected

5 Click **OK** to close the Test Connection pop-up and then click **DOWNLOAD** to download the configuration files to the module through the path.



Download files from PC to module

Module Running

STEP 1: Select Communication Path:

Select Connection Type: 1756 ENBT

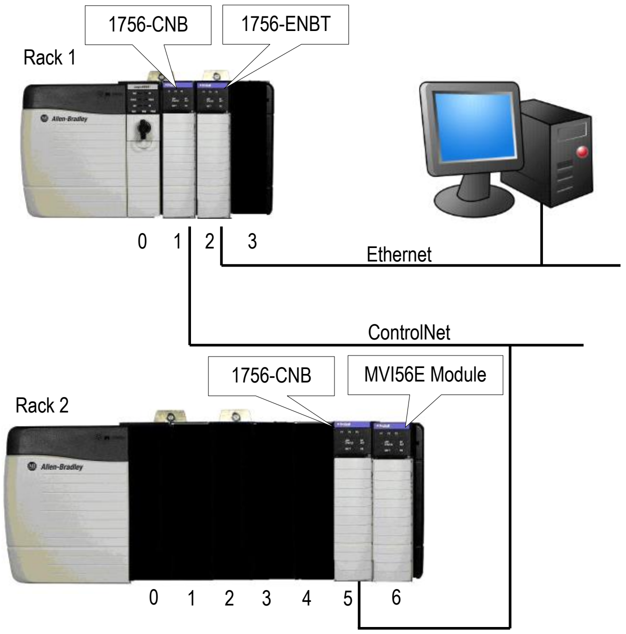
Ethernet: 192 . 168 . 0 . 250

CIPconnect: t:192.168.0.100,p:1,s:0$56

STEP 2: Transfer File(s):

Example 2: Remote Rack Application

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



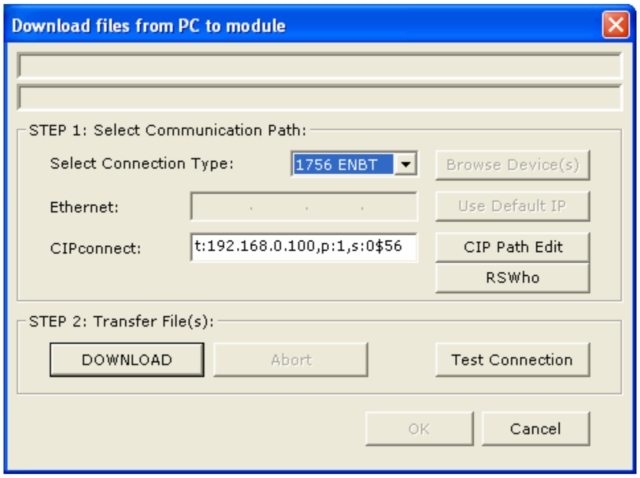
Rack 1

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

Rack 2

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

1 In the *Download* dialog box, click **CIP PATH EDIT**.



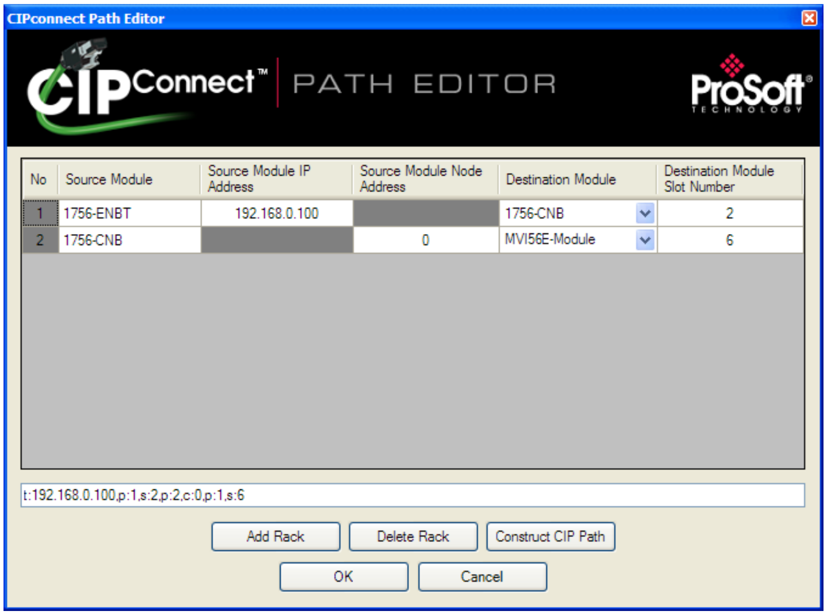
Download files from PC to module
STEP 1: Select Communication Path:
Browse Device(s)
Use Default IP
CIP Path Edit
RSWho
STEP 2: Transfer File(s):
DOWNLOAD
Abort
Test Connection
OK
Cancel

Select Connection Type: 1756 ENBT

Ethernet: . . .

CIPconnect: t:192.168.0.100,p:1,s:0$56

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

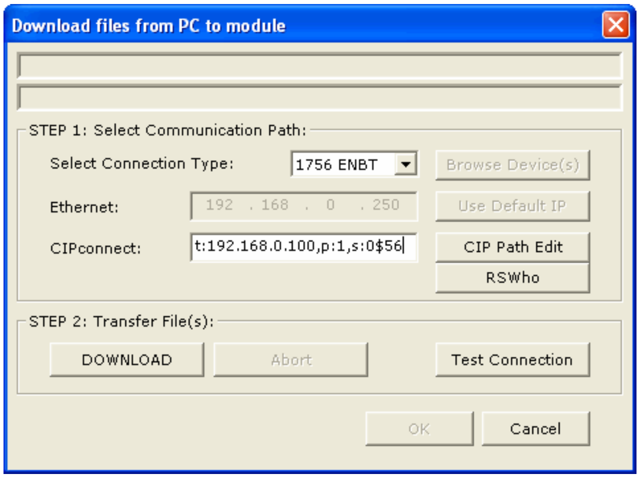


No	Source Module	Source Module IP Address	Source Module Node Address	Destination Module	Destination Module Slot Number
1	1756-ENBT	192.168.0.100		1756-CNB	2
2	1756-CNB		0	MVI56E-Module	6

t:192.168.0.100,p:1,s:2,p:2,c:0,p:1,s:6

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

3 Check the new path in the *Download* dialog box.


Download files from PC to module

STEP 1: Select Communication Path:

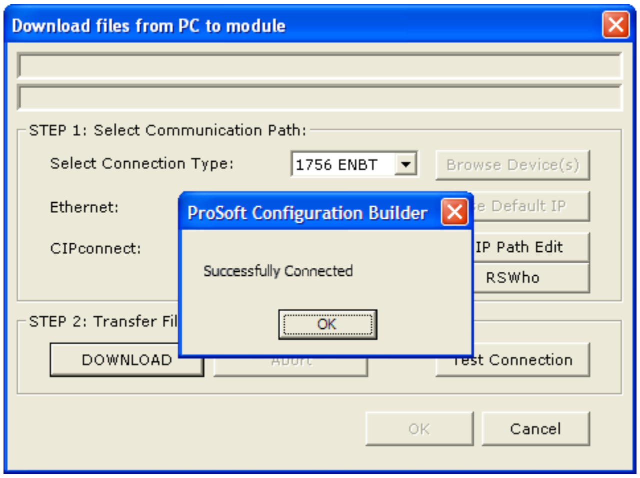
Select Connection Type: 1756 ENBT

Ethernet: 192 . 168 . 0 . 250

CIPconnect: t:192.168.0.100,p:1,s:0$56

STEP 2: Transfer File(s):

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


Download files from PC to module

STEP 1: Select Communication Path:

Select Connection Type: 1756 ENBT

Ethernet:

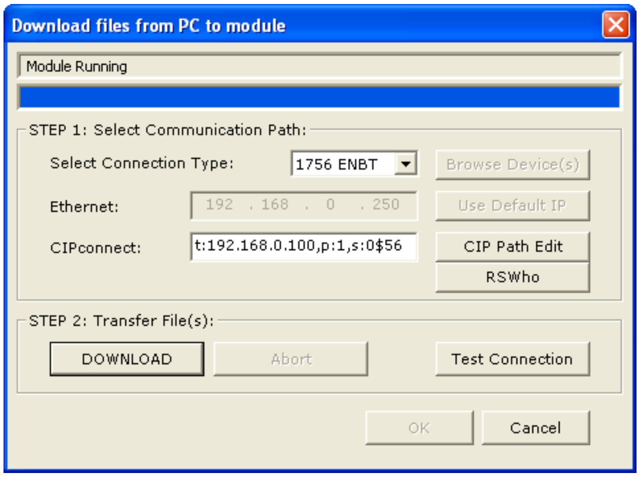
CIPconnect:

ProSoft Configuration Builder

Successfully Connected

STEP 2: Transfer Fil

5 Click **DOWNLOAD** to download the configuration files to the module through the path.


Download files from PC to module

Module Running

STEP 1: Select Communication Path:

Select Connection Type: 1756 ENBT

Ethernet: 192 . 168 . 0 . 250

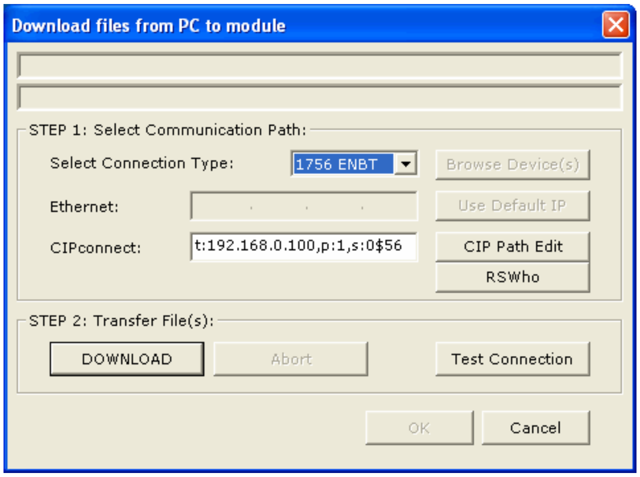
CIPconnect: t:192.168.0.100,p:1,s:0$56

STEP 2: Transfer File(s):

2.3.2 Using RSWho to Connect to the Module

> **Note:** RSLinx must be installed on your PC to use this feature. You also need an ENBT module set up in the rack. For information on setting up the ENBT module, see Using CIPconnect to Connect to the Module (page 84).

1. In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-SIE** module.
2. From the shortcut menu, choose **DOWNLOAD FROM PC TO DEVICE**.
3. In the *Download* dialog box, choose **1756 ENBT** from the *Select Connection Type* dropdown box.



Download files from PC to module

STEP 1: Select Communication Path:

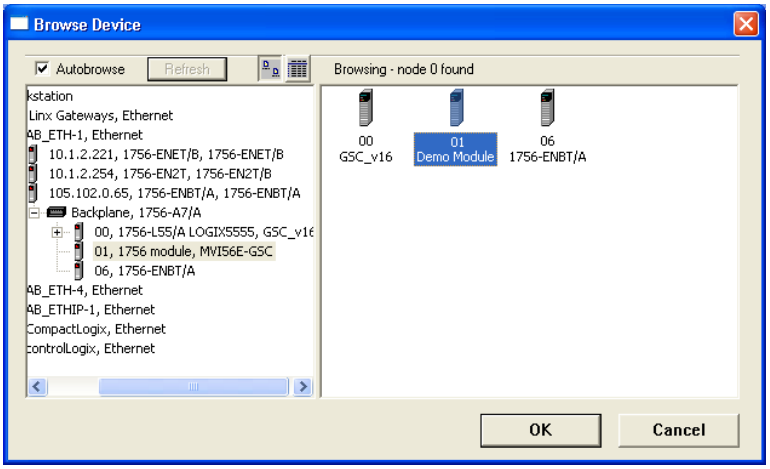
Select Connection Type: 1756 ENBT

Ethernet: . . .

CIPconnect: t:192.168.0.100,p:1,s:0$56

STEP 2: Transfer File(s):

4. Click **RSWHO** to display modules on the network. The MVI56E-SIE module will automatically be identified on the network.



Browse Device

☑ Autobrowse

Browsing - node 0 found

kstation
Linx Gateways, Ethernet
AB_ETH-1, Ethernet
10.1.2.221, 1756-ENET/B, 1756-ENET/B
10.1.2.254, 1756-EN2T, 1756-EN2T/B
105.102.0.65, 1756-ENBT/A, 1756-ENBT/A
Backplane, 1756-A7/A
00, 1756-L55/A LOGIX5555, GSC_v16
01, 1756 module, MVI56E-GSC
06, 1756-ENBT/A
AB_ETH-4, Ethernet
AB_ETHIP-1, Ethernet
CompactLogix, Ethernet
ControlLogix, Ethernet

5. Select the module, and then click **OK**.
6. In the *Download* dialog box, click **DOWNLOAD**.

3 Ladder Logic

Ladder logic is required for managing communication between the MVI56E-SIE module and the processor. The ladder logic handles tasks such as:

- Module backplane data transfer
- Special block handling
- Status data receipt

Additionally, a power-up handler may be needed to initialize the module's database and may clear some processor fault conditions.

The sample Import Rung with Add-On Instruction is extensively commented to provide information on the purpose and function of each user-defined data type and controller tag. For most applications, the Import Rung with Add-On Instruction will work without modification.

3.1 Controller Tags

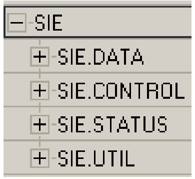
Data related to the MVI56E-SIE is stored in the ladder logic in variables called controller tags. Individual controller tags can be grouped into collections of controller tags called controller tag structures. A controller tag structure can contain any combination of:

- Individual controller tags
- Controller tag arrays
- Lower-level controller tag structures

The controller tags for the module are pre-programmed into the Add-On Instruction Import Rung ladder logic. *Controller Tags* can be found in the subfolder, located in the *Controller* folder in the *Controller Organizer* pane of the main RSLogix 5000 window.

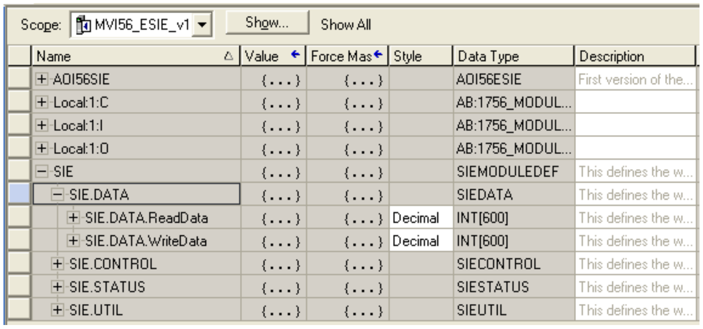
This controller tag structure is arranged as a tree structure. Individual controller tags are found at the lowest level of the tree structure. Each individual controller tag is defined to hold data of a specific type, such as integer or floating-point data. Controller tag structures are declared with user-defined data types, which are collections of data types.

The main *SIE* controller tag structure is divided into four lower-level tag structures.



```
⊟ SIE
  ⊞ SIE.DATA
  ⊞ SIE.CONTROL
  ⊞ SIE.STATUS
  ⊞ SIE.UTIL
```

The four lower-level controller tag structures contain other controller tags and controller tag structures. Click the **[+]** sign next to any controller tag structure to expand it and view the next level in the structure. For example, the *SIE.DATA* controller tag structure is expanded, it can be seen that there are two controller tag arrays, *SIE.DATA.ReadData* and *SIE.DATA.WriteData*, which are 600-element integer arrays by default.



Scope: MVI56_ESIE_v1 Show... Show All

Name	Value	Force Mas	Style	Data Type	Description
+ AOI56SIE	{...}	{...}		AOI56ESIE	First version of the...
+ Local:1:C	{...}	{...}		AB:1756_MODUL...	
+ Local:1:I	{...}	{...}		AB:1756_MODUL...	
+ Local:1:O	{...}	{...}		AB:1756_MODUL...	
− SIE	{...}	{...}		SIEMODULEDEF	This defines the w...
− SIE.DATA	{...}	{...}		SIEDATA	This defines the w...
+ SIE.DATA.ReadData	{...}	{...}	Decimal	INT[600]	This defines the w...
+ SIE.DATA.WriteData	{...}	{...}	Decimal	INT[600]	This defines the w...
+ SIE.CONTROL	{...}	{...}		SIECONTROL	This defines the w...
+ SIE.STATUS	{...}	{...}		SIESTATUS	This defines the w...
+ SIE.UTIL	{...}	{...}		SIEUTIL	This defines the w...

Each controller tag in the Add-On Instruction is commented in the *Description* column.

Note: The *Data Type* column displays the data types used to declare each controller tag, controller tag array or controller tag structure. Individual controller tags are declared with basic data types, such as INT and BOOL. Controller tag arrays are declared with arrays of basic data types. Controller tag structures are declared with user-defined data types (UDTs).

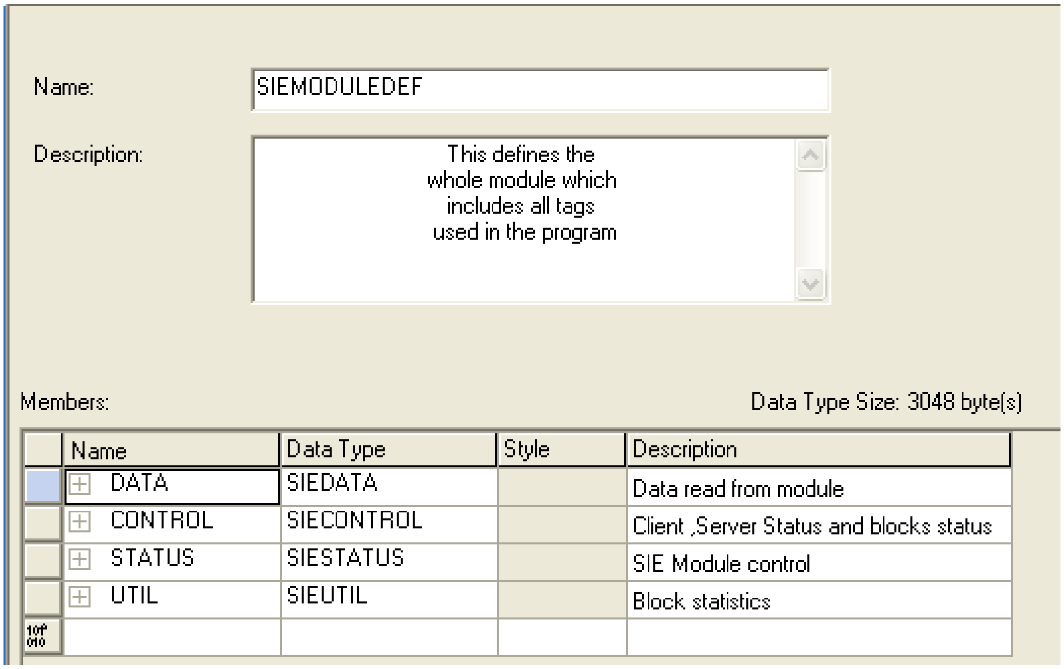
3.2 User-Defined Data Types (UDTs)

User-defined data types (UDTs) allow users to organize collections of data types into groupings. These groupings, or data type structures, can then be used to declare the data types for controller tag structures. Another advantage of defining a UDT is that it may be re-used in other controller tag structures that use the same data types.

The Add-On Instruction Import Rung ladder logic for the module has pre-defined UDTs. You can find them in the *User-Defined* subfolder, located in the *Data Types* folder in the *Controller Organizer* pane of the main RSLogix window. Like the controller tags, the UDTs are organized in a multiple-level tree structure.

Eleven different UDTs are defined for the MVI56E-SIE Add-On Instruction.

The main UDT, *SIEMODULEDEF*, contains all the data types for the module and was used to create the main controller tag structure, *SIE.* There are four UDTs one level below *SIEMODULEDEF*. These lower-level UDTs were used to create the *SIE.DATA*, *SIE.CONTROL*, *SIE.STATUS*, and *SIE.UTIL* controller tag structures.



Name: SIEMODULEDEF

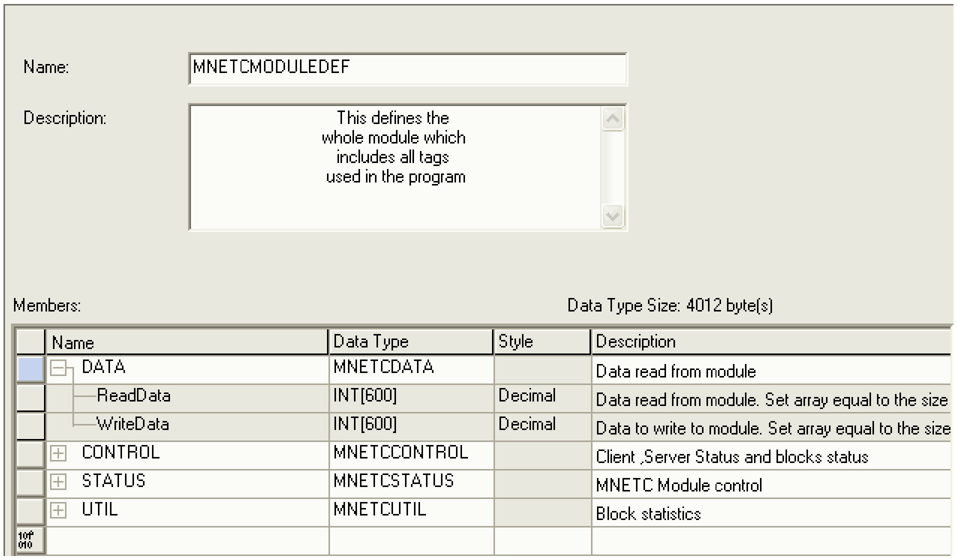
Description: This defines the whole module which includes all tags used in the program

Members: Data Type Size: 3048 byte(s)

Name	Data Type	Style	Description
⊞ DATA	SIEDATA		Data read from module
⊞ CONTROL	SIECONTROL		Client ,Server Status and blocks status
⊞ STATUS	SIESTATUS		SIE Module control
⊞ UTIL	SIEUTIL		Block statistics

Click the **[+]** signs to expand the UDT structures and view lower-level UDTs.

For example, if you expand *SIE.DATA is expanded*, two UDTs can be seen, *ReadData* and *WriteData*. Both of these are 600-element integer arrays by default.



Name: MNETCMODULEDEF

Description: This defines the whole module which includes all tags used in the program

Members: Data Type Size: 4012 byte(s)

Name	Data Type	Style	Description
⊟ DATA	MNETCDATA		Data read from module
ReadData	INT[600]	Decimal	Data read from module. Set array equal to the size
WriteData	INT[600]	Decimal	Data to write to module. Set array equal to the size
⊞ CONTROL	MNETCCONTROL		Client ,Server Status and blocks status
⊞ STATUS	MNETCSTATUS		MNETC Module control
⊞ UTIL	MNETCUTIL		Block statistics

Note: These UDTs are the data types used to declare the *SIE.DATA.ReadData* and *SIE.DATA.WriteData* controller tag arrays.

Each UDT is commented in the *Description* column.

3.3 Using Controller Tags

Controller tags can be used to:

- View read and write data that is being transferred between the module and the processor.
- View status data for the module.
- Set up and trigger special functions.
- Initiate module restarts (Warm Boot or Cold Boot).

3.4 Controller Tag Overview

Controller Tag	Description
SIE.DATA	SIE input and output data transferred between the processor and the module
SIE.CONTROL	Governs the data movement between the PLC rack and the module
SIE.STATUS	Status information
SIE.UTIL	Block statistics and generic tags used for internal ladder processing (DO NOT MODIFY)

The following sections describe each of these controller tag structures in more detail.

3.4.1 SIE.DATA

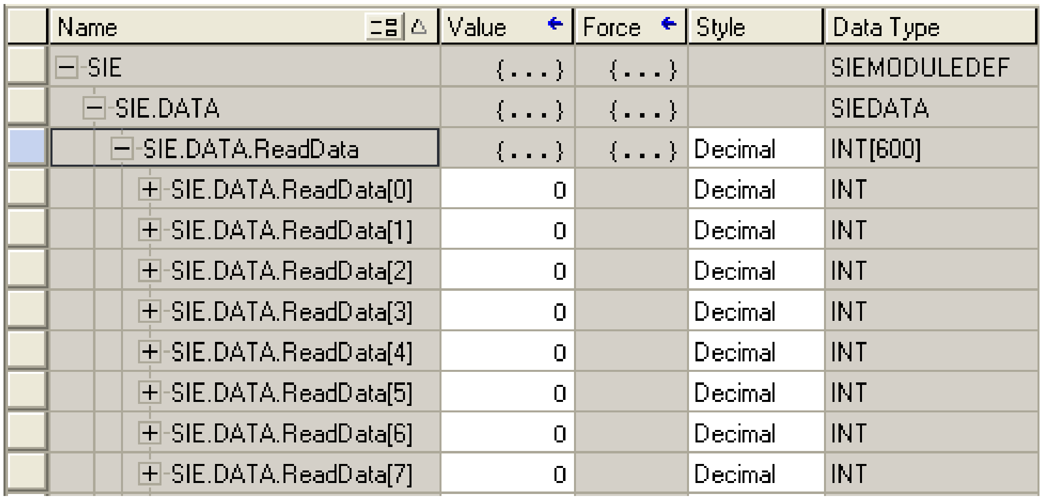
The controller tags in *SIE.DATA* hold data to be transferred between the processor and the MVI56E-SIE module. This read and write data is transferred between the processor and module as "pages," or blocks, of data up to 200 words long.

The data types for the *SIE.DATA.ReadData* and *SIE.DATA.WriteData* controller tag arrays are integer arrays containing variable numbers of elements.

Controller Tag	Data Type	Description
ReadData	INT[x]	Data read from module. Array size is equal to the size set in the configuration.
WriteData	INT[x]	Data to write to module. Array size is equal to the size set in the configuration.

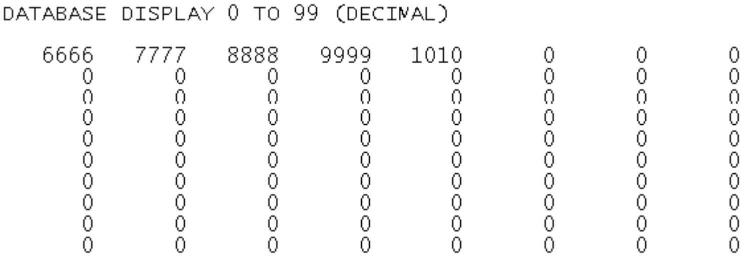
SIE.DATA.ReadData

ReadData is a controller tag array that automatically adjusts to match the value entered in the *Read Register Count* (page 36) parameter of the configuration. For ease of use, this array should be dimensioned as a multiple of 200 words. This data is paged up to 200 words at a time from the module to the processor. The ladder logic places the data received into the proper position in the *ReadData* array. This data is used for status and control in the processor ladder logic.



Name	Value	Force	Style	Data Type
⊟ SIE	{...}	{...}		SIEMODULEDEF
⊟ SIE.DATA	{...}	{...}		SIEDATA
⊟ SIE.DATA.ReadData	{...}	{...}	Decimal	INT[600]
⊞ SIE.DATA.ReadData[0]	0		Decimal	INT
⊞ SIE.DATA.ReadData[1]	0		Decimal	INT
⊞ SIE.DATA.ReadData[2]	0		Decimal	INT
⊞ SIE.DATA.ReadData[3]	0		Decimal	INT
⊞ SIE.DATA.ReadData[4]	0		Decimal	INT
⊞ SIE.DATA.ReadData[5]	0		Decimal	INT
⊞ SIE.DATA.ReadData[6]	0		Decimal	INT
⊞ SIE.DATA.ReadData[7]	0		Decimal	INT

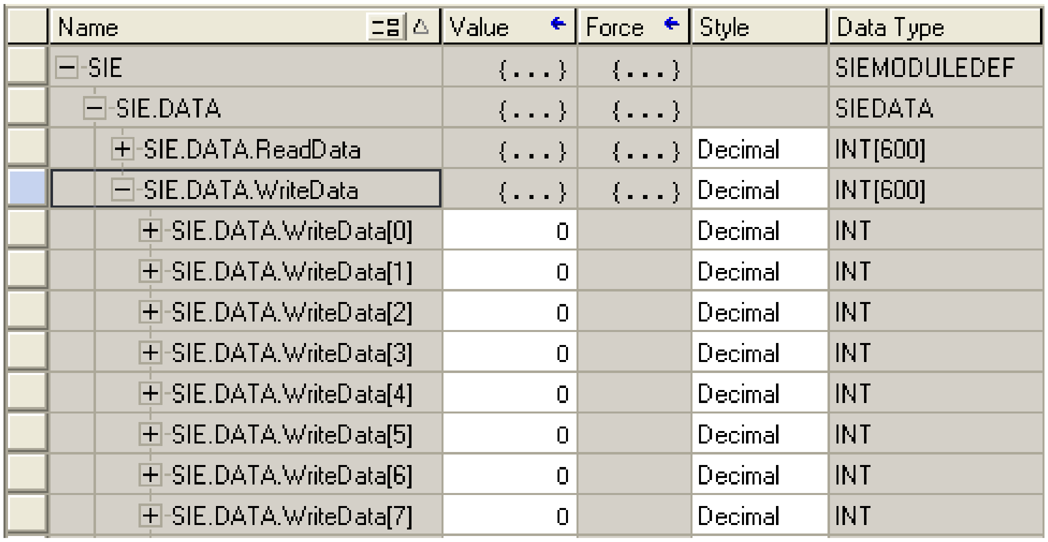
The *ReadData* array is related to the contents of the Read Data area of the module's internal database. To view the actual registers in the module's internal database, access the database display from *ProSoft Configuration Builder's Diagnostics* menu. For more information, see the section on *PCB* Diagnostics (page 112).



```
DATABASE DISPLAY 0 TO 99 (DECIMAL)

   6666    7777    8888    9999    1010       0       0       0
      0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0
      0       0       0       0       0       0       0       0
```

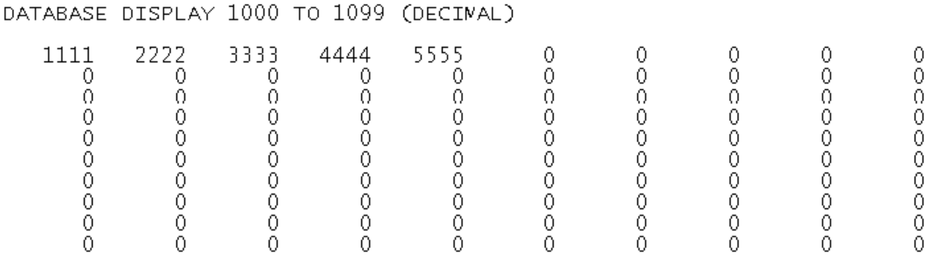
SIE.DATA.WriteData

WriteData is a controller tag array that automatically adjusts to match the value entered in the *Write Register Count* (page 37) parameter of the configuration. For ease of use, this array should be dimensioned as a multiple of 200 words. This data is paged up to 200 words at a time from the processor to the module. The ladder logic places the write data into the output image for transfer to the module. This data is passed from the processor to the module for status and control information for use in other nodes on the network.



Name	Value	Force	Style	Data Type
⊟ SIE	{...}	{...}		SIEMODULEDEF
⊟ SIE.DATA	{...}	{...}		SIEDATA
⊞ SIE.DATA.ReadData	{...}	{...}	Decimal	INT[600]
⊟ SIE.DATA.WriteData	{...}	{...}	Decimal	INT[600]
⊞ SIE.DATA.WriteData[0]	0		Decimal	INT
⊞ SIE.DATA.WriteData[1]	0		Decimal	INT
⊞ SIE.DATA.WriteData[2]	0		Decimal	INT
⊞ SIE.DATA.WriteData[3]	0		Decimal	INT
⊞ SIE.DATA.WriteData[4]	0		Decimal	INT
⊞ SIE.DATA.WriteData[5]	0		Decimal	INT
⊞ SIE.DATA.WriteData[6]	0		Decimal	INT
⊞ SIE.DATA.WriteData[7]	0		Decimal	INT

The *WriteData* array is related to the contents of the Write Data area of the module's internal database. To view the actual registers in the module's internal database, access the database display from *ProSoft Configuration Builder's Diagnostics* menu. For more information, see the section on *PCB* Diagnostics (page 112).



```
DATABASE DISPLAY 1000 TO 1099 (DECIMAL)

   1111   2222   3333   4444   5555      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0
```

3.4.2 SIE.CONTROL

This controller tag structure is used to request special tasks from the module. For more information, see Special Function Blocks (page 130).

Controller Tag	Data Type	Description
BootTimer	TIMER	Timer used to clear both cold and warm boot requests
ColdBoot	BOOL	Hardware reset of the module
WarmBoot	BOOL	Configuration data reset in the module
ResetStatus	BOOL	Reset Status trigger
CmdID	INT	Command ID from 1 to 16
CmdControl	SIECMDCONTROL	Holds Command Control statistics
CmdControlPending	BOOL	Halts rung until module is ready
CmdControlTrigger	BOOL	Command Control Trigger
IPAddress	SIEIPADDRESS	Getting and setting IP address to and from module

3.4.3 SIE.STATUS

This controller tag structure contains module and Client status data. For a more complete description of the *SIE.STATUS* controller tag structure, refer to the Status Data Definition (page 117).

Name	Data Type	Description
PassCnt	INT	Program cycle counter
ProductVersion	INT	This is used to pass the product version to the processor
ProductCode	INT[2]	This is used to pass the product code to the processor
BlockStats	SIEBLOCKSTATS	Block transfer statistics
CmdBits	INT[20]	Command bits array to be used for 20 Clients
ClientStatsTrigger	BOOL	Get Client status
ClientID	INT	Client ID to get status from
ClientStatus	SIECLIENTSTATS[30]	Client status data
CmdErrorList	INT[16]	Command Error List

3.4.4 SIE.UTIL

This controller tag structure stores the variables required for the data transfer between the processor and the MVI56E-SIE module.

Name	Data Type	Description
LastRead	INT	Index of last read block
LastWrite	INT	Index of last write block
BlockIndex	INT	Computed block offset for data table
StatusIndex	INT	Computed block offset for status data
ReadDataSizeGet	INT	Gets *ReadData* array length
WriteDataSizeGet	INT	Gets *WriteData* array length
ReadDataBlkCount	INT	Holds the value of the block counts of the *ReadData* array
WriteDataBlkCount	INT	Holds the value of the block counts of the *WriteData* array
RBTSremainder	INT	Holds remainder calculation value from the read array
WBTSremainder	INT	Holds remainder calculation value from the write array
IPsetPending	BOOL	Allows setting module IP address
IPgetPending	BOOL	Allows getting module IP address
InitOutputData	SIEINITOUTDATA	Used to bring the module into a known state after a restart operation
CheckInitialization	BOOL	Check initialization trigger
StatusSeqIndex	INT	Index of Status Sequence

The *LastRead* tag stores the latest Read Block ID received from the module. The *LastWrite* tag stores the latest Write Block ID to be sent to the module. The *BlockIndex* tag is an intermediate variable used during the block calculation.

4 Diagnostics and Troubleshooting

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

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

4.1 LED Status Indicators

4.1.1 Scrolling LED Status Indicators

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

Initialization Messages

Code	**Message**
Boot / DDOK	Module is initializing
Ladd	Module is waiting for required module configuration data from ladder logic to configure the application port(s)
Waiting for Processor Connection	Module did not connect to processor during initialization ▪ Sample ladder logic or AOI is not loaded on processor ▪ Module is located in a different slot than the one configured in the ladder logic/AOI ▪ Processor is not in RUN or REM RUN mode
Last config: <date>	Indicates the last date when the module changed its IP address. You can update the module date and time through the module's web page, or with the Optional MVI56E Add-On Instruction.
	After power up and every reconfiguration, the module will display the configuration of the application port(s). The information consists of: **Client** ▪ C0 C2 C3 C4 ….. C29

Operation Messages

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

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

Code	Message
<Backplane Status>	OK: Module is communicating with processor ERR: Module is unable to communicate with processor. For this scenario, the <Port Status> message above is replaced with "Processor faulted or is in program mode".
<IP Address>	Module IP address
<C0>	OK: Port is communicating without error Communication Errors: port is having communication errors. Refer to PCB diagnostics (page 103) for further information about the error.

4.1.2 Ethernet LED Indicators

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

LED	State	Description
10/100	Off	No activity on the Ethernet port.
	Green Flash	The Ethernet port is actively transmitting or receiving data.
LINK/ACT	Off	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	Green Solid	Physical network connection detected. This LED must be On solid for Ethernet communication to be possible.

4.1.3 Non-Scrolling LED Status Indicators

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

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

4.1.4 Troubleshooting

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

Processor Errors

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

Module Errors

Problem description	**Steps to take**
BP ACT LED (not present on MVI56E modules) remains OFF or blinks slowly MVI56E modules with scrolling LED display: <*Backplane Status*> condition reads ERR	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in RUN or REM RUN mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write data block transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is properly configured in the processor I/O configuration and ladder logic.
OK LED remains RED	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert it, and then restore power to the rack.

4.1.5 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.

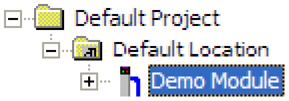
If the module's OK LED does not turn GREEN, verify that the module is inserted completely into the rack. If this does not cure the problem, contact ProSoft Technology Technical Support.

4.2 Using the Diagnostics Menu in ProSoft Configuration Builder

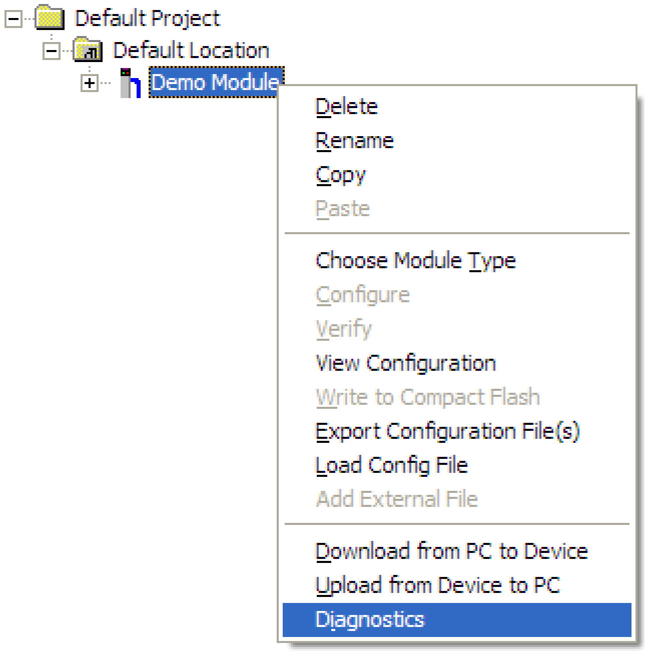
Tip: ProSoft Configuration Builder *Diagnostics* is able to have more than one module window open at one time.

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

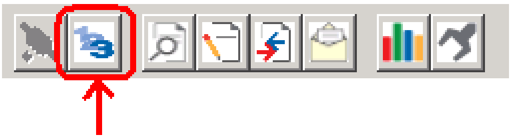
1 In *ProSoft Configuration Builder*, select the module, and then click the right mouse button to open a shortcut menu.



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

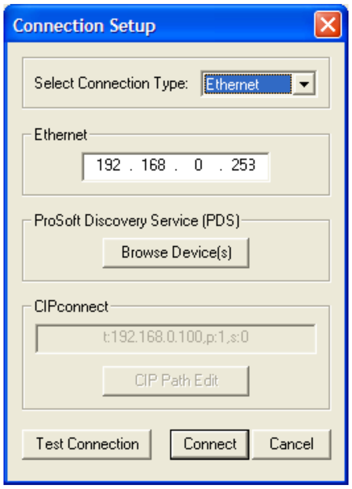


3 In the *Diagnostics* window, click the **SET UP CONNECTION** button to browse for the module's IP address.



Click to set up connection

4 In the *Connection Setup* dialog box, click the **TEST CONNECTION** button to verify that the module is accessible with the current settings.



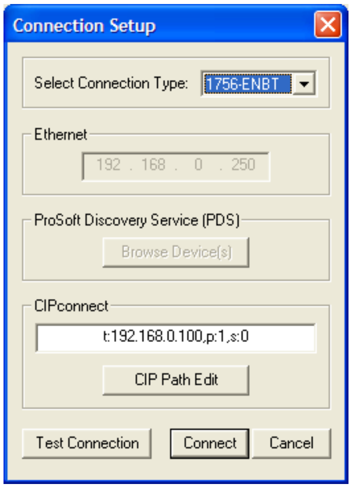
Connection Setup

Select Connection Type: Ethernet

Ethernet: 192 . 168 . 0 . 253

CIPconnect: t:192.168.0.100,p:1,s:0

You can also use CIPconnect® to connect to the module through a 1756-ENBT card. Refer to Using CIPconnect to Connect to the Module (84) for information on how to construct a CIP path.



Connection Setup

Select Connection Type: 1756-ENBT

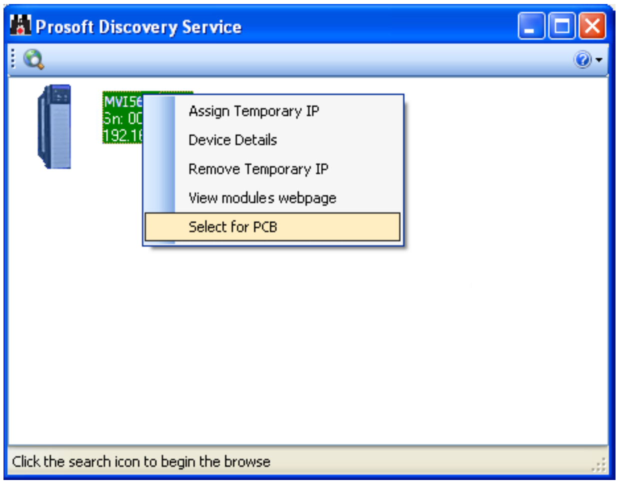
Ethernet: 192 . 168 . 0 . 250

CIPconnect: t:192.168.0.100,p:1,s:0

5 If the *Test Connection* is successful, click **CONNECT**.

If *PCB* is unable to connect to the module:

1 Click the **BROWSE DEVICE(S)** button to open the *ProSoft Discovery Service*. Select the module, then right-click and choose **SELECT FOR PCB**.



Click the search icon to begin the browse

2 Close *ProSoft Discovery Service*, and click the **CONNECT** button again.

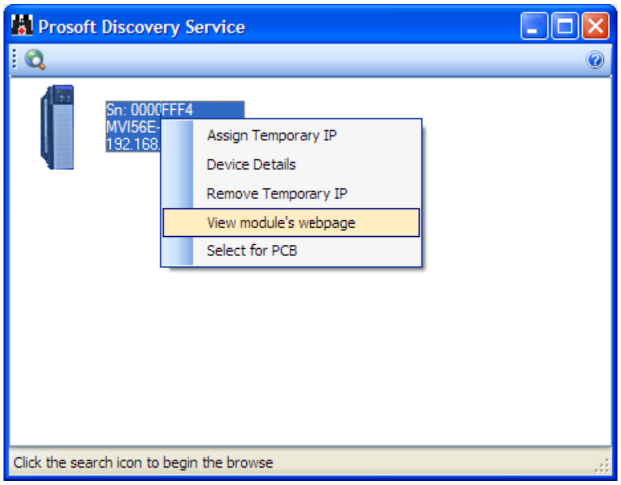
3 If these troubleshooting steps fail, verify that the Ethernet cable is connected properly between the computer and the module, either through a hub or switch (using the grey cable) or directly between the computer and the module (using a red crossover cable).

If a connection is still not able to be established, contact ProSoft Technology for assistance.

4.2.1 Connecting to the Module's Webpage

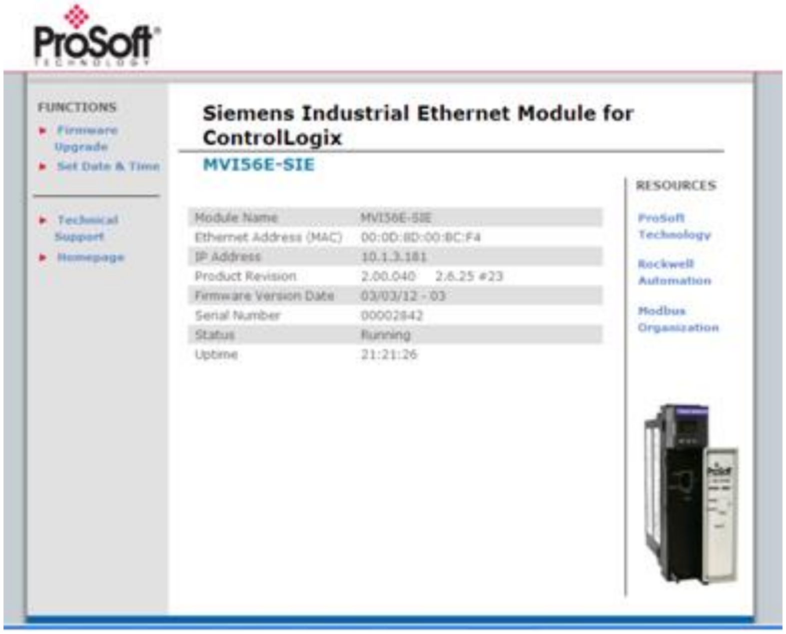
The module's internal web server provides access to general product information, firmware download link, and links to the ProSoft Technology Website.

1 In *ProSoft Discovery Service*, select the module, and then click the right mouse button to open a shortcut menu.



Click the search icon to begin the browse

2 On the shortcut menu, choose **VIEW MODULE'S WEBPAGE**.



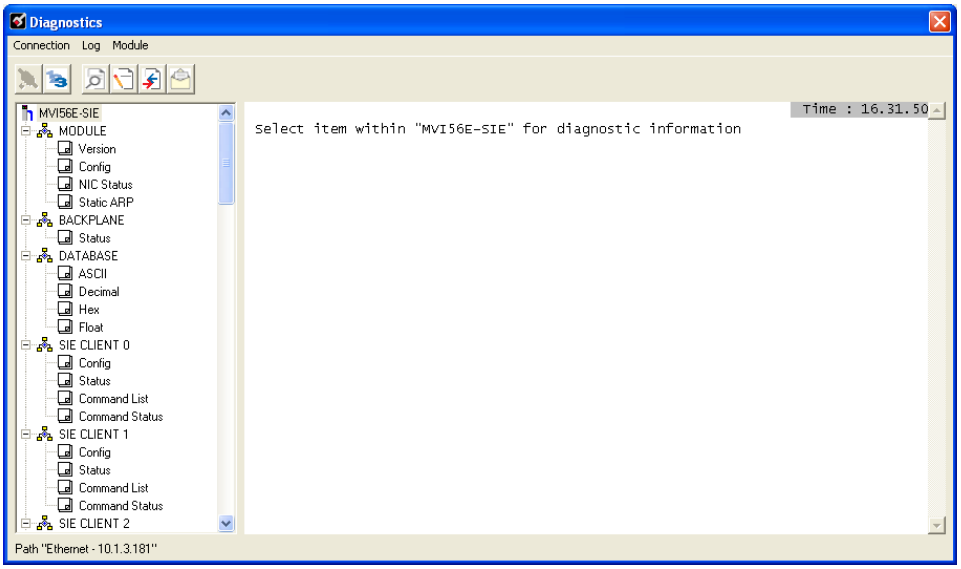
Siemens Industrial Ethernet Module for ControlLogix

MVI56E-SIE

Module Name	MVI56E-SIE
Ethernet Address (MAC)	00:0D:8D:00:BC:F4
IP Address	10.1.3.181
Product Revision	2.00.040 2.6.25 #23
Firmware Version Date	03/03/12 - 03
Serial Number	00002842
Status	Running
Uptime	21:21:26

4.2.2 Diagnostics Menu

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



```
Time : 16.31.50
Select item within "MVI56E-SIE" for diagnostic information
```

Path "Ethernet - 10.1.3.181"

4.2.3 Monitoring Module Information

Use the *MODULE* menu to view configuration and hardware information for the MVI56E-SIE module's backplane and Ethernet application port.

Version

Use the *Version* menu to view module hardware and firmware information.

Config

Use the *Configuration* menu to view backplane configuration settings for the module.

The information on this menu corresponds with the configuration information in the *Module* settings in *ProSoft Configuration Builder*.

NIC Status

Use the *NIC Status* (Network Interface Card) menu to view configuration and status information for the MVI56E-SIE module's Ethernet application port.

This information is useful for troubleshooting Ethernet network connectivity problems.

Static ARP

Use the *Static ARP* menu to view the list of IP and MAC addresses that are configured not to receive ARP (Address Resolution Protocol) messages from the module.

The Static ARP Table (page 77) defines a list of static IP addresses that the module will use when an ARP is required.

4.2.4 Monitoring Backplane Information

Use the *BACKPLANE* menu to view the backplane status information for the MVI56E-SIE module.

Backplane Status

Use the *Status* menu to view current backplane status, including

- Number of retries
- Backplane status
- Fail count
- Number of words read
- Number of words written
- Number of words parsed
- Error count
- Command count

During normal operation, the read, write, and parsing values should increment continuously, while the error value should not increment.

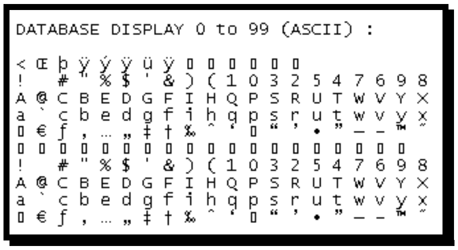
The status values on this menu correspond with members of the Status Data Definition (page 117).

4.2.5 Monitoring Database Information

Use the *DATABASE* menu to view the contents of the MVI56E-SIE module’s internal database.

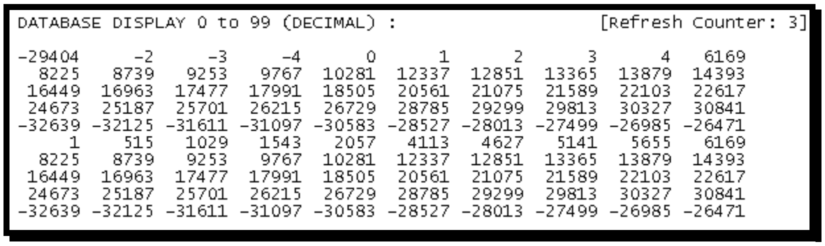
You can view data in the following formats:

ASCII



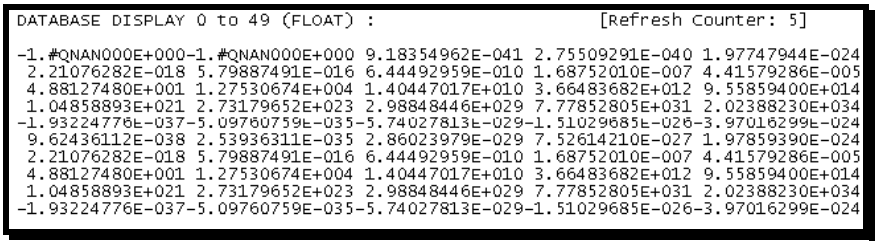
```
DATABASE DISPLAY 0 to 99 (ASCII) :

< Œ þ ÿ ý ÿ ü ÿ ▯ ▯ ▯ ▯ ▯ ▯
!   # " % $ ' & ) ( 1 0 3 2 5 4 7 6 9 8
A @ C B E D G F I H Q P S R U T W V Y X
a ` c b e d g f i h q p s r u t w v y x
▯ € ƒ , … „ ‡ † ‰ ˆ ‘ ▯ “ ’ • ” — – ™ ˜
▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯ ▯
!   # " % $ ' & ) ( 1 0 3 2 5 4 7 6 9 8
A @ C B E D G F I H Q P S R U T W V Y X
a ` c b e d g f i h q p s r u t w v y x
▯ € ƒ , … „ ‡ † ‰ ˆ ‘ ▯ “ ’ • ” — – ™ ˜
```

Decimal



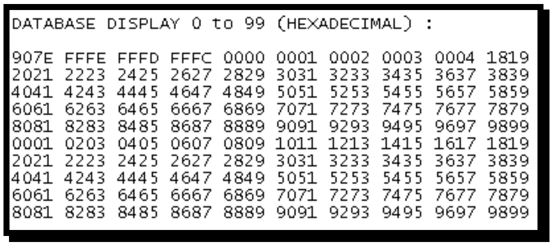
```
DATABASE DISPLAY 0 to 99 (DECIMAL) :                  [Refresh Counter: 3]

-29404     -2     -3     -4      0      1      2      3      4   6169
  8225   8739   9253   9767  10281  12337  12851  13365  13879  14393
 16449  16963  17477  17991  18505  20561  21075  21589  22103  22617
 24673  25187  25701  26215  26729  28785  29299  29813  30327  30841
-32639 -32125 -31611 -31097 -30583 -28527 -28013 -27499 -26985 -26471
     1    515   1029   1543   2057   4113   4627   5141   5655   6169
  8225   8739   9253   9767  10281  12337  12851  13365  13879  14393
 16449  16963  17477  17991  18505  20561  21075  21589  22103  22617
 24673  25187  25701  26215  26729  28785  29299  29813  30327  30841
-32639 -32125 -31611 -31097 -30583 -28527 -28013 -27499 -26985 -26471
```

Float



```
DATABASE DISPLAY 0 to 49 (FLOAT) :                    [Refresh Counter: 5]

-1.#QNAN000E+000-1.#QNAN000E+000 9.18354962E-041 2.75509291E-040 1.97747944E-024
 2.21076282E-018 5.79887491E-016 6.44492959E-010 1.68752010E-007 4.41579286E-005
 4.88127480E+001 1.27530674E+004 1.40447017E+010 3.66483682E+012 9.55859400E+014
 1.04858893E+021 2.73179652E+023 2.98848446E+029 7.77852805E+031 2.02388230E+034
-1.93224776E-037-5.09760759E-035-5.74027813E-029-1.51029685E-026-3.97016299E-024
 9.62436112E-038 2.53936311E-035 2.86023979E-029 7.52614210E-027 1.97859390E-024
 2.21076282E-018 5.79887491E-016 6.44492959E-010 1.68752010E-007 4.41579286E-005
 4.88127480E+001 1.27530674E+004 1.40447017E+010 3.66483682E+012 9.55859400E+014
 1.04858893E+021 2.73179652E+023 2.98848446E+029 7.77852805E+031 2.02388230E+034
-1.93224776E-037-5.09760759E-035-5.74027813E-029-1.51029685E-026-3.97016299E-024
```

Hexadecimal



```
DATABASE DISPLAY 0 to 99 (HEXADECIMAL) :

907E FFFE FFFD FFFC 0000 0001 0002 0003 0004 1819
2021 2223 2425 2627 2829 3031 3233 3435 3637 3839
4041 4243 4445 4647 4849 5051 5253 5455 5657 5859
6061 6263 6465 6667 6869 7071 7273 7475 7677 7879
8081 8283 8485 8687 8889 9091 9293 9495 9697 9899
0001 0203 0405 0607 0809 1011 1213 1415 1617 1819
2021 2223 2425 2627 2829 3031 3233 3435 3637 3839
4041 4243 4445 4647 4849 5051 5253 5455 5657 5859
6061 6263 6465 6667 6869 7071 7273 7475 7677 7879
8081 8283 8485 8687 8889 9091 9293 9495 9697 9899
```

Use the scroll bar on the right edge of the window to view each page (100 words) of data.

4.2.6 Monitoring SIE Client Information

Use the *SIE CLIENT x* menu to view the configuration and status information for the SIE Client(s).

Config

Use the *Configuration* menu to view configuration settings for SIE Client x.

The information on this menu corresponds with the configuration information in the *SIE Client x* settings in *ProSoft Configuration Builder*.

Status

Use the *Status* menu to view status for SIE Client x. During normal operation, the number of requests and responses should increment, while the number of errors should not change.

Command List

Use the *Command List* menu to view the command list settings for SIE Client x. The information on this menu corresponds with the *SIE Client x Commands* settings in *ProSoft Configuration Builder*.

Use the scroll bar on the right edge of the window to view each SIE Client command.

Command Status

Use the *Command Status* menu to view SIE Client x Command status.

A zero indicates no error.

A non-zero value indicates an error. For an explanation of each value, refer to Client Command Error (page 120).

4.3 Reading Status Data from the Module

Module status information is useful for troubleshooting and can be accessed in several different ways.

In the ladder logic's *SIE.STATUS* controller tag structure.

The MVI56E-SIE module returns status data in the input image that can be used to determine the module’s operating status. This data is transferred from the module to the ControlLogix processor continuously as part of the normal data transfer block sequence (page 127). You can view this data in the *SIE.STATUS* controller tag structure in the ladder logic.

Client status data can also be requested and returned in a special Client Status block (page 132), outside of the normal data transfer block sequence. The status data contained in the Client Status block is different from the status data in the normal data transfer blocks. It can also be viewed in the *SIE.STATUS* controller tag structure.

For more information about status data in *SIE.STATUS*, see the Status Data Definition (page 117).

In *ProSoft Configuration Builder's Diagnostics* screens.

For more information, see the section on *PCB Diagnostics* (page 112).

In database locations specified by *Error/Status Pointers* (optional).

If optional *Error/Status Pointers* are enabled, status data can also be found in the Read Data area of the module’s database at the locations specified by the pointer configuration parameters. For more information, see Backplane Error/Status Pointer (page 36), Client Error/Status Pointer (page 39) and Command Error Pointer (page 40).

4.3.1 Status Data Definition

This section contains a description of the controller tags in the *SIE.STATUS* controller tag structure, which contains module and Client status data.

- The first ten controller tags contain status data routinely transferred from the module to the processor in the normal data transfer block sequence (page 127).
- The remaining controller tags are used to request and receive Client status data via the Client Status block functionality (page 132).

Note: In order to access up-to-date status data from these remaining controller tags, a Client Status block must be recently received from the module. Client Status blocks are not routinely sent from the module; they are returned on a once-per-request basis as a response to a Client Status block request from the processor.

Controller Tag	Data Type	Description
PassCnt	INT	This value is incremented each time a complete program cycle occurs in the module.
ProductVersion	INT	Product version
ProductCode	INT[2]	Product code
BlockStats.Read	INT	Total number of read blocks transferred from the module to the processor
BlockStats.Write	INT	Total number of write blocks transferred from the processor to the module
BlockStats.Parse	INT	Total number of blocks successfully parsed that were received from the processor
BlockStats.Cmd	INT	Total number of Command Control blocks received from the processor
BlockStats.Err	INT	Total number of block errors recognized by the module
CmdBits[x]	INT	Displays enabled or disabled status of all 16 commands in the *Client x Command List* for each Client
ClientStatsTrigger	BOOL	Initiates request for Client Status block from module when set to **1**
ClientID	INT	Specifies Client (**0**-**19**) to request status data from
ClientStatus[x].CmdReq	INT	Total number of command list requests sent from Client
ClientStatus[x].CmdResp	INT	Total number of command list responses received by Client
ClientStatus[x].CmdErr	INT	This value is incremented each time an error message is received from a remote unit or a local error is generated for a command.
ClientStatus[x].Requests	INT	Not used
ClientStatus[x].Responses	INT	Not used
ClientStatus[x].ErrSent	INT	Not used
ClientStatus[x].ErrRec	INT	Not used
ClientStatus[x].CfgErrWord	INT	Configuration Error Word - This word contains a bitmap that indicates general module configuration errors.
ClientStatus[x].CurErr	INT	Most recent error code recorded for the Client
ClientStatus[x].LastErr	INT	Previous most recent error code recorded for the Client
CmdErrorList[x]	INT	Command error code for each command (0-15) on the specified Client's command list

4.3.2 Configuration Error Word

The *Configuration Error Word* contains Client configuration error indications, in a bit-mapped format. Specific bits in the module's *Configuration Error Word* are turned on (set to 1) to indicate various configuration errors. The *Configuration Error Word* appears in the *SIE.STATUS.ClientStatus[x]* controller tag array.

Bits set to 1 in the *Configuration Error Word* indicate the following errors.

Bit	Description	Hex Value
0	Reserved - not currently used	0001h
1	Reserved - not currently used	0002h
2	Reserved - not currently used	0004h
3	Reserved - not currently used	0008h
4	Invalid retry count parameter	0010h
5	The float flag parameter is not valid.	0020h
6	The float start parameter is not valid.	0040h
7	The float offset parameter is not valid.	0080h
8	The ARP Timeout is not in range (ARP Timeout parameter 0 or greater than 60000 milliseconds) and will default to 5000 milliseconds.	0100h
9	The Command Error Delay is > 300 and will default to 300.	0200h
10	Reserved - not currently used	0400h
11	Reserved - not currently used	0800h
12	Reserved - not currently used	1000h
13	Reserved - not currently used	2000h
14	Reserved - not currently used	4000h
15	Reserved - not currently used	8000h

Combinations of errors will result in more than one bit being set in the error word. Correct any invalid data in the configuration for proper module operation. A value of zero (0) in this word indicates all bits are clear, which means that all module configuration parameters contain valid values. However, this does not mean that the configuration is valid for the user application. Make sure each parameter is set correctly for the intended application.

4.3.3 Client Command Errors

There are several different ways to view Client Command Errors.

- In the *SIE.STATUS.CmdErrorList* controller tag array
- On the Client status data screens in the *ProSoft Configuration Builder Diagnostics*
- At a module database location specified by the configuration's *SIE Client x Command Error Pointer*, if the *Command Error Pointer* is enabled. This means that the first register refers to command 1 and so on.

Word Offset	Description
0	Command 0 Error
1	Command 1 Error
2	Command 2 Error
3	Command 3 Error
…	….
…	…
15	Command 15 Error
16	Command 16 Error

For every command that has an error, the module automatically sets the poll delay parameter to 30 seconds. This instructs the module to wait 30 seconds until it attempts to issue the command again.

As the commands in the Client Command Last are polled and executed, an error value is maintained in the module for each command. This error list can be transferred to the processor.

Module Communication Error Codes

Decimal	Hex	Description
1	0x0001	No data from I/O module
3	0x0003	The desired item is not available in the PLC (200 family)
5	0x0005	The desired address is beyond limit for this PLC
6	0x0006	The CPU does not support reading a bit block of length<>1
7	0x0007	Write data size error
10	0x000a	The desired item is not available in the PLC
-123	0xff85	Cannot evaluate the received PDU
-124	0xff84	The PLC returned a packet with no result data
-125	0xff83	The PLC returned an error code not understood by this library
-126	0xff82	this result contains no data
-127	0xff81	cannot work with an undefined result set
-128	0xff80	Unexpected function code in answer";
-129	0xff7f	PLC responds with an unknown data type
-1024	0xfc00	Short packet from PLC
-1025	0xfbff	Timeout when waiting for PLC response
-32767	0x8001	Not allowed in current operating status
-32511	0x8101	Hardware fault
-32509	0x8103	Object access not allowed
-32508	0x8104	Context is not supported. Step7 says: Function not implemented or error in telegram
-32507	0x8105	Invalid address
-32506	0x8106	Data type not supported
-32505	0x8107	Data type not consistent
-32502	0x810A	Object does not exist
-31999	0x8301	Insufficient CPU memory
-31742	0x8402	CPU already in RUN or already in STOP
-31740	0x8404	Severe error
-31488	0x8500	Incorrect PDU size
-30974	0x8702	Address invalid
-12286	0xd002	Step7: Variant of command is illegal
-12284	0xd004	Step7: Status for this command is illegal
-12127	0xd0A1	Step7: Function is not allowed in the current protection level
-11775	0xd201	Block name syntax error
-11774	0xd202	Syntax error function parameter
-11773	0xd203	Syntax error block type
-11772	0xd204	No linked block in storage medium

Decimal	Hex	Description
-11771	0xd205	Object already exists
-11770	0xd206	Object already exists
-11769	0xd207	Block exists in EPROM
-11767	0xd209	Block does not exist/could not be found
-11762	0xd20e	No block present
-11760	0xd210	Block number too big
-11712	0xd240	Coordination rules were violated
-11711	0xd241	Operation not permitted in current protection level
-11710	0xd242	Protection violation while processing F-blocks. F-blocks can only be processed after password input
-11263	0xd401	Invalid SZL ID
-11262	0xd402	Invalid SZL index
-11258	0xd406	Diagnosis: Info not available
-11255	0xd409	Diagnosis: DP error
-9215	0xdc01	Invalid BCD code or Invalid time format

SIE Client Specific Error Codes

Decimal	Hex	Description
-33	0xffdf	Failed to connect to server specified in command
-34	0xffde	Failed to create a socket
-36	0xffdc	SIE command response timeout. No valid response received.
-37	0xffdb	TCP/IP connection ended before session finished

Command List Entry Error Codes

Decimal	Hex	Description
-40	0xffd8	Too few parameters
-41	0xffd7	Invalid enable code
-42	0xffd6	Internal address > maximum address
-44	0xffd4	Count parameter set to 0
-45	0xffd3	Invalid function code
-46	0xffd2	Invalid swap code
-47	0xffd1	Invalid TSAP code

5 Reference

5.1 Product Specifications

The MVI56E Siemens Industrial Ethernet Client Communication Module allows Rockwell Automation® ControlLogix® Programmable Automation Controllers (PACs) to interface easily with multiple Siemens Industrial Ethernet server-compatible instruments and devices. The multi-Client module improves performance when controlling multiple servers on a Siemens Industrial Ethernet network, by supporting up to 20 Clients.

The MVI56E-SIE includes configuration and management through the module's Ethernet port, and CIPconnect® technology for bridging though ControlNet™ and EtherNet/IP™ networks.

5.1.1 General Specifications

- Single-slot 1756 ControlLogix backplane compatible
- 10/100 Mbps auto crossover detection Ethernet configuration and application port
- User-definable module data memory mapping of up to 5000 16-bit registers
- CIPconnect-enabled network configuration and diagnostics monitoring using ControlLogix 1756-ENxT and 1756-CNB modules
- ProSoft Configuration Builder (PCB) software supported, a Windows-based graphical user interface providing simple product and network configuration
- Add-On Instructions (AOI) are used for data transfer between module and processor
- 4-character, alpha-numeric, scrolling LED display of status and diagnostics data in plain English – no cryptic error or alarm codes to decipher
- ProSoft Discovery Service (PDS) software used to locate the module on the network and assign temporary IP address
- Personality Module - a non-volatile industrial-grade Compact Flash (CF) card used to store network and module configuration for easy disaster recovery, allowing quick in-the-field product replacement by transferring the CF card

Siemens Industrial Ethernet Client (Master)

The MVI56E-SIE is a Client-only module that will operate on a local or remote rack. This module was created to improve performance when controlling multiple servers on a Siemens Industrial Ethernet network.

- Offers 20 Client connections with up to 16 commands each to talk to multiple servers
- Actively reads data from and writes data to Siemens Industrial Ethernet devices, using Siemens Industrial Ethernet protocol.
- ControlLogix processor can be programmed to use special functions to control the activity on the Client by actively selecting commands to execute from the command list (Command Control)

5.1.2 Functional Specifications

- Siemens Industrial Ethernet data types overlap in the module's memory database, so the same data can be conveniently read or written as bit-level or register-level data.
- Configurable floating-point data movement is supported
- Configurable parameters for the Client including a minimum response delay of 0 to 65535 ms
- Supports up to 20 Clients with up to 16 commands for each Client
- Error codes, counters, and module status available from module memory through the Clients, or through the ladder logic and controller tags in RSLogix 5000

5.1.3 Hardware Specifications

Specification	Description
Dimensions	Standard 1756 ControlLogix® single-slot module
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 Hz to 150 Hz
Relative humidity	5% to 95% (with no condensation)
LED indicators	Battery Status (ERR) Application Status (APP) Module Status (OK)
4-character, scrolling, alphanumeric LED display	Shows module, version, IP, application port setting, port status, and error information
Ethernet port	10/100 Base-T, RJ45 Connector, for CAT5 cable Link and Activity LED indicators Auto-crossover cable detection

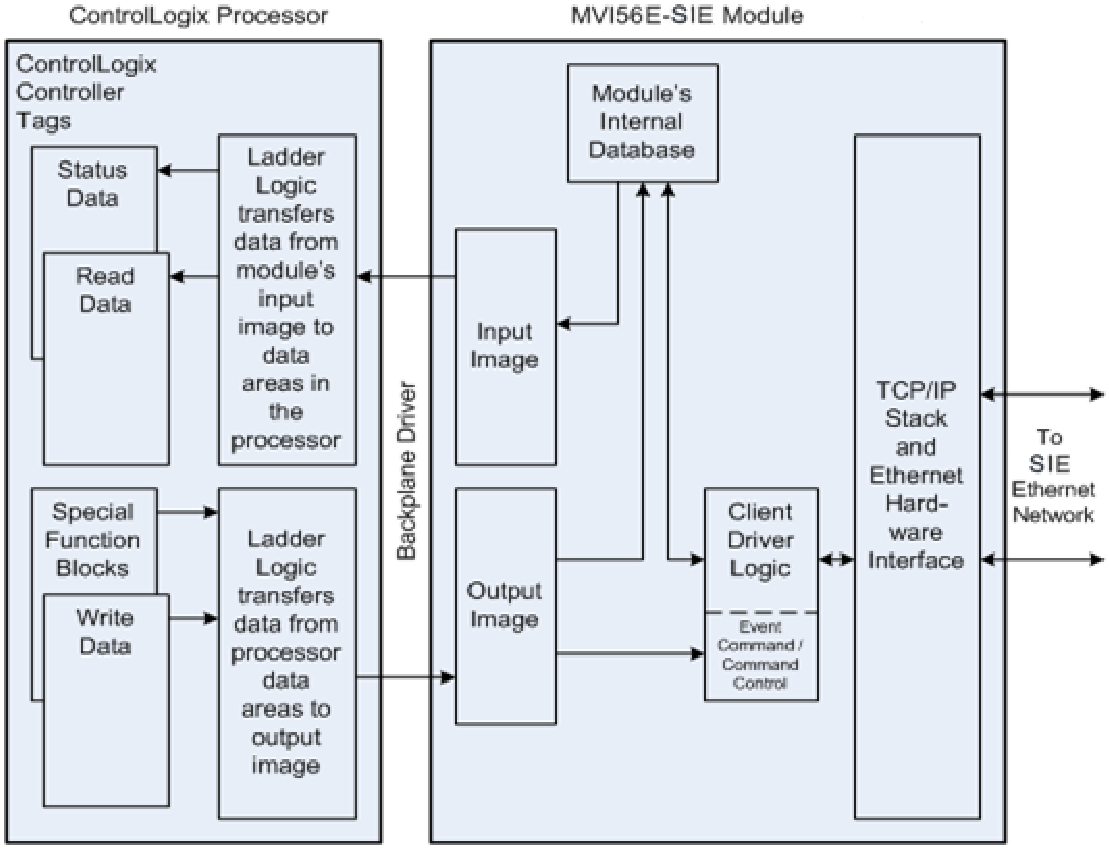
5.2 Backplane Data Transfer

The MVI56E-SIE 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 update times range from 1 to 10 milliseconds.

This bi-directional transfer of data is accomplished by the module putting data in the input image to send to the processor. Data in the input image is placed in the processor's controller tags by ladder logic. The input image is set to 250 words.

Processor logic inserts data to the output image to be transferred to the module. The module's firmware program extracts the data and places it in the module's internal database. The output image is set to 248 words.

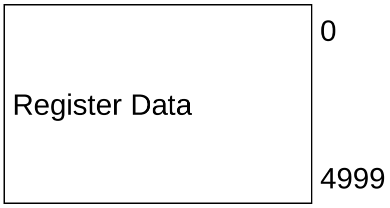
The following illustration shows the data transfer method used to move data between the ControlLogix processor, the MVI56E-SIE module and the Siemens Industrial Ethernet Network.


ControlLogix Processor
MVI56E-SIE Module
ControlLogix Controller Tags
Status Data
Read Data
Ladder Logic transfers data from module's input image to data areas in the processor
Special Function Blocks
Write Data
Ladder Logic transfers data from processor data areas to output image
Backplane Driver
Module's Internal Database
Input Image
Output Image
Client Driver Logic
Event Command / Command Control
TCP/IP Stack and Ethernet Hard-ware Interface
To SIE Ethernet 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. This database is defined as a data table with addresses from 0 to 4999

Module’s Internal Database Structure

5000 registers for user data



Data contained in this database is transferred in blocks, or pages, using the input and output images. ControlLogix ladder logic and the MVI56E-SIE module's program work together to coordinate these block transfers. Up to 200 words of data can be transferred from the module to the processor (read block - input image) or from the processor to the module (write block - output image) in each block transfer. The block structure of each block type depends on the data content and function of the block. The module uses the following block identification numbers:

Block ID Range	Descriptions
-1	Null block
0	This feature enhances performance, especially when using less than 200 words of read/write data: ▪ If Read Register Count in the module configuration file is set > 200 words, Block ID 0 is not used. ▪ If Read Register Count in the module configuration file is set >0 and <= 200 words, Block ID contains the same data as block 1 (both read data and status data).
1 to 25	Read or Write blocks
1000 to 1024	Initialize Output Data blocks
3000 to 3029	Client Status blocks
5001 to 5016	Command Control blocks
9990	Set Module IP Address block
9991	Get Module IP Address block
9998	Warm-boot block
9999	Cold-boot block

These block identification codes can be broken down into two groups:

Normal data transfer blocks

- Read and Write blocks (-1 to 25)

Special function blocks

- Initialize Output Data blocks (1000 to 1024)
- Client Status blocks (3000 to 3029)
- Command Control blocks (5001 to 5016)
- Module IP Address blocks (9990 and 9991)
- Warm-boot and Cold-boot blocks (9998 and 9999)

5.2.1 Normal Data Transfer Blocks

Normal data transfer includes the paging of user data from the module's internal database (registers 0 to 4999), as well as paging of status data. These data are transferred through read (input image) and write (output image) blocks.

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 following table describes the structure of the input image.

Read Block from Module to Processor

Word Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2 to 201	Read Data	200
202	Program Scan Counter	1
203 to 208	Block Transfer Status	6
209 to 238	Command bit data for Clients	20
239 to 240	Product Code	2
241	Product Version	1
242 to 248	Reserved	7
249	Read Block ID	1

The Read Block ID is an index value used to determine where the 200 words of data from module memory will be placed in the *ReadData[x]* controller tag array of the ControlLogix processor. 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. The Write Block ID associated with the block requests data from the ControlLogix processor.

During 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 Siemens Industrial Ethernet network or operator control through the module's Configuration/Debug port.

Status Data in Read Block

The following table describes in more detail the status information found in the Read Block.

Word Offset	Content	Description
202	Program Scan Count	This value is incremented each time a complete program cycle occurs in the module.
203	Read Block Count	This field contains the total number of read blocks transferred from the module to the processor.
204	Write Block Count	This field contains the total number of write blocks transferred from the processor to the module.
205	Parse Block Count	This field contains the total number of blocks successfully parsed that were received from the processor.
207	Command Block Count	This field contains the total number of command blocks received from the processor.
208	Error Block Count	This field contains the total number of block errors recognized by the module.
209	Client 0 command execution word	Each bit in this word enables/disable the commands for Client 0. If the bit is set, the command will execute. If the bit is clear, the command will be disabled. This data is set in the output image (WriteBlock) from the ladder logic.
210 to 228	Client 1 to Client 19 command execution words	These 19 words are used for each of the other 19 Clients in the module. This data is set in the output image (WriteBlock) from the ladder logic.
239 to 240	Product Code	The product ID code for the module
241	Product Version	The firmware version number for the module

Status information transferred in the Read block can be viewed in the *SIE.STATUS* controller tag in the ladder logic. For more information, see the Status Data Definition (page 117).

Write Block

These blocks of data transfer information from the ControlLogix processor to the module. The following table describes the structure of the output image.

Write Block from Processor to Module

Word Offset	Description	Length
0	Write Block ID	1
1 to 200	Write Data	200
201 to 220	Command bit data for Clients (set)	20
231 to 246	Spare	16
247	Select Priority Read Block	1

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.

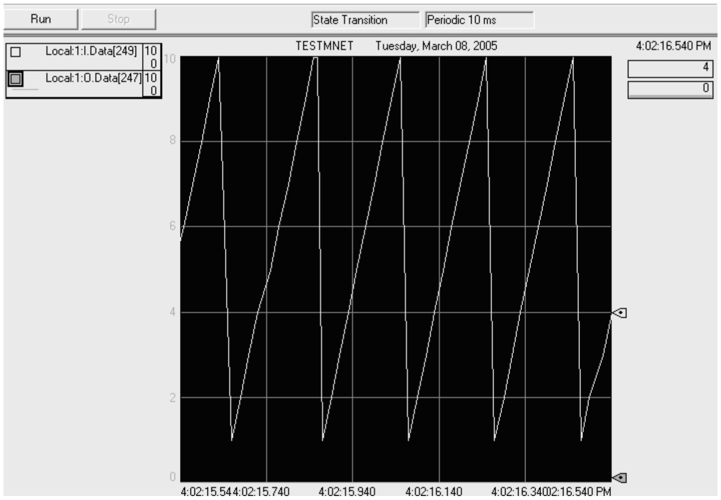
Select Priority Read Block (Write Block Offset 247)

This register allows the processor to select which read blocks will be returned from the module. If this register equals zero, the module will return all read blocks in sequential order.

If this register has a non-zero value, the module will return the read block selected, and the following one.

This feature can be used for applications that require some read blocks to be updated more frequently than other blocks.

The following illustrations show the effect of changing the value of the Select Priority Read Block register (Write Block offset 247). In the following histogram curve, the Select Priority Read Block is equal to 0.



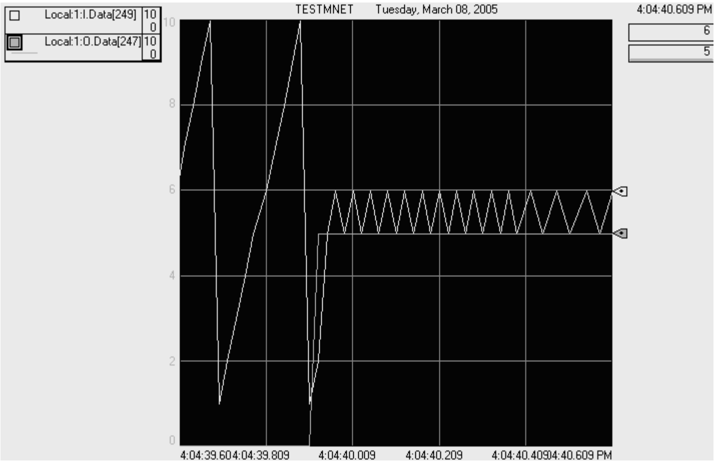
- Local:1.O.Data[247] = Select Priority Read Block.
- Local:1.I.Data[249] = Read Block ID.

In the example above, all read blocks (1 to 10) are returned in sequential order.

Select Priority Read Block = 5

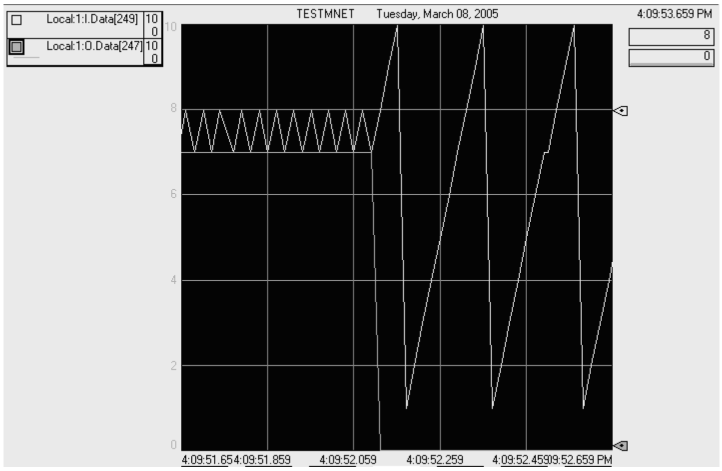
If the ladder logic changes the value of Local:1:O.Data[247] from 0 to 5, note that the Local:1:I.Data[249] value begins to alternate between Block IDs 5 and 6 as long as Local:1:I.Data[247] stays set to 5.

5-6-5-6-5-6-5-6-5-6-…



Select Priority Read Block = 0

After the ladder logic changes the value of Local:1:O.Data[247] from 5 to 0, then the Local:1:I.Data[249] value is updated as before, by returning all blocks 1 through 10 in a repeating sequence.

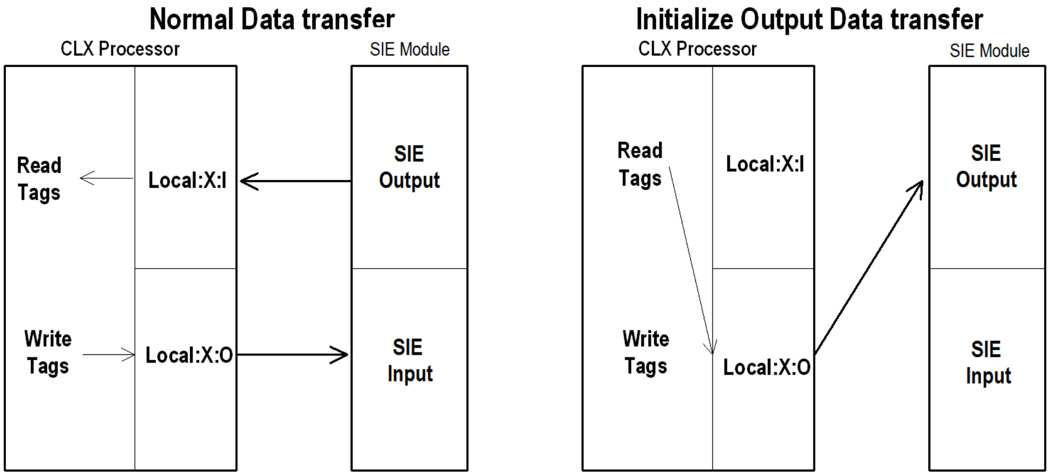


5.2.2 *Special Function Blocks*

Special function blocks are optional blocks used to request special tasks from the module.

Initialize Output Data (Blocks 1000 to 1024)

Use the *Initialize Output Data* parameter in the configuration to bring the module to a known state after a restart operation. If the *Initialize Output Data* parameter is enabled, when the module performs a restart operation, it will request blocks of output data from the *ReadData* array in the processor to initialize the Read Data area of the module's internal database.



Block Request from Module to Processor

Word Offset	Description	Length
0	Reserved	1
1	1000 to 1024	1
2 to 248	Spare	247
249	1000 to 1024	1

Ladder logic subtracts 1000 from the value contained in word 249 to determine a block index. This bock index determines which 200-word block of data will be taken from the *ReadData* array and placed in the output image to be returned to the module.

Block Response from Processor to Module

Word Offset	Description	Length
0	1000 to 1024	1
1 to 200	Output data to preset in module.	200
201 to 247	Spare	47

Client Status (Blocks 3000 to 3019)

Client status data for a specific Client can be requested and returned in a special Client Status block. The status data contained in the Client Status block is different from the status data contained in the normal data transfer blocks.

Block Request from Processor to Module

Word Offset	Description	Length
0	3000 to 3029 (last digits indicate which Client to consider)	1
1 to 247	Spare	247

Block Response from Module to Processor

Word Offset	Description	Length
0	0	1
1	Write Block ID	1
2	3000 to 3029 number requested	1
3 to 12	Client status data	10
13 to 28	Command error list data for Client	16
29 to 248	Reserved	220
249	3000 to 3029	1

Client Status Data

Word Offset	Client Status
3	Total number of command list requests
4	Total number of command list responses
5	Total number of command list errors
6	Not used
7	Not used
8	Not used
9	Not used
10	Configuration Error Word
11	Current Error
12	Last Error

Status information transferred in the Client Status block can be viewed in the *SIE.STATUS* controller tag in the ladder logic. For more information, see Status Data Definition (page 117).

Controller Tags

To issue a Client Status block request, enter the appropriate values in the following members of the *SIE.STATUS* controller tag in the ladder logic.

Controller Tag	Data Type	Description
ClientID	INT	Enter the Client (**0**-**19**) to request status data for.
ClientStatsTrigger	BOOL	Set the value of this tag to **1** to trigger the Client Status block request.

Command Control (Blocks 5001 to 5016)

Note: Command Control is not needed for normal Siemens Industrial Ethernet command list polling operations and is needed only occasionally for special circumstances.

During routine operation, the module continuously cycles through the user-defined *SIE Client x Command List* (page 41) for each Client, examining commands in the order they are listed, and sending enabled commands on the network. However, the module also has a special command priority queue, which is an internal buffer that holds commands from special function blocks until they can be sent on the network.

When one or more commands appear in the command priority queue:

1. The routine polling process is temporarily interrupted.
2. The commands in the command priority queue are executed until the queue is empty.
3. Then the module goes back to where it left off on the *SIE Client x Command List* and continues routine polling.

Command Control blocks place commands into the module's command priority queue. Command Control is used with commands already defined in the *SIE Client x Command List*.

Commands in the *SIE Client x Command List* may be either enabled for routine polling or disabled and excluded from routine polling. A disabled command has its bit in the *SIE.CONTROL.CmdControl.WriteCmdBits* controller tag set to zero (**0**) and is skipped during routine polling. An enabled command has its bit in the *WriteCmdBits* controller tag set to one (**1**) and is sent during routine polling. However, Command Control allows any command in the predefined *SIE Client x Command List* to be added to the command priority queue, whether it is enabled for routine polling or not.

Command Control also gives you the option to use ladder logic to have commands from the *SIE Client x Command List* executed at a higher priority and out of routine order, if such an option might be required in special circumstances.

A single Command Control block request can place up to 16 commands from the *SIE Client x Command List* into the command priority queue.

Block Request from Processor to Module

Word Offset	Description	Length
0	Command Control block identification code of **5001** to **5016**. The rightmost digit indicates the number of commands (**1** to **16**) to add to the command priority queue.	1
1	Client index	1
2	This word contains the Command Index for the first command to be entered into the queue.	1
3	Command Index 2	1
4	Command Index 3	1
5	Command Index 4	1
6	Command Index 5	1
7	Command Index 6	1
8	Command Index 7	1
9	Command Index 8	1
10	Command Index 9	1
11	Command Index 10	1
12	Command Index 11	1
13	Command Index 12	1
14	Command Index 13	1
15	Command Index 14	1
16	Command Index 15	1
17	Command Index 16	1
18 to 247	Spare	230

The last digit in the block identification code indicates the number of commands to process. For example, a block identification code of **5003** indicates that three commands are to be placed in the queue. In this case, the first three of the 16 available Command Indexes will be used to determine exactly which three commands will be added to the queue, and to set their order of execution.

Values to enter for the 16 Command Indexes range from **0** to **15** and correspond to the *SIE Client x Command List* entries, which are numbered from 1 to 16. To determine the Command Index value, subtract one (**1**) from the row number of the command in the *SIE Client x Command List*, as seen in the *Command Editor* window of *ProSoft Configuration Builder (PCB)*.

The module responds to a Command Control block request with a response block, indicating the number of commands added to the command priority queue.

Block Response from Module to Processor

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	5001 to 5016	1

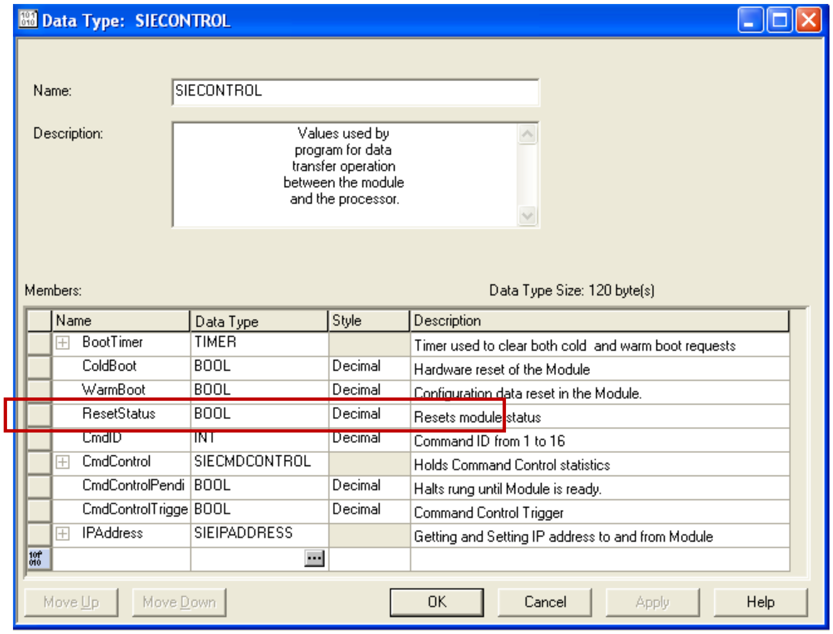
Controller Tags

The *SIE.CONTROL* controller tag array holds all the values needed to create one Command Control block, have it sent to the module, and control the processing of the returned response block.

Controller Tag	Description
CmdID	Enter a decimal value representing the quantity of commands to be requested in the Command Control block (**1** to **16**). This value is used by the ladder logic to generate the Command Control Block ID. The rightmost digits of the Command Control Block ID are the number of commands requested by the block.
CmdControl.ClientID	Enter the Client to issue the commands to (**0** to **19**)
CmdControl.CMDqty	Not used
CmdControl.CmdIndex	Enter the **ROW NUMBER** of the command in the *SIE Client x Command List* in *Prosoft Configuration Builder* minus **1**. This is a 16-element array. Each element holds one Command Index.
CmdControl.WriteCmdBits	Enter a **1** (enable) or a **0** (disable) to select which commands on the configuration's *Client x Command List* will be executed during routine polling. There is one 16-bit word for each of the 20 Clients. Each of the 16 bits corresponds to one of the 16 commands available to each Client. The state of these *WriteCmdBits* overrides whatever value may be assigned to the *Enable* parameter in the configuration. **Note:** This parameter only affects routine polling. It has no effect on Command Control blocks.
CmdControlPending	Not used
CmdControlTrigger	Set this tag to **1** to trigger the execution of a Command Control block after all the other parameters have been entered.

Reset Module Status (Block 9971) SIE

This block allows the processor to reset all status values available from the module to the processor or through the PCB diagnostics menu. This block is triggered through the following data type and controller tag elements:



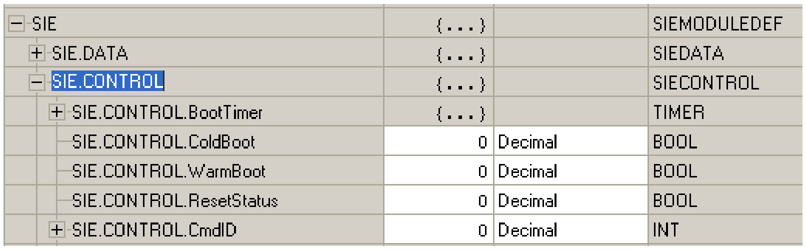
Name: SIECONTROL

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

Members:

Data Type Size: 120 byte(s)

Name	Data Type	Style	Description
⊞ BootTimer	TIMER		Timer used to clear both cold and warm boot requests
ColdBoot	BOOL	Decimal	Hardware reset of the Module
WarmBoot	BOOL	Decimal	Configuration data reset in the Module.
ResetStatus	BOOL	Decimal	Resets module status
CmdID	INT	Decimal	Command ID from 1 to 16
⊞ CmdControl	SIECMDCONTROL		Holds Command Control statistics
CmdControlPendi	BOOL	Decimal	Halts rung until Module is ready.
CmdControlTrigge	BOOL	Decimal	Command Control Trigger
⊞ IPAddress	SIEIPADDRESS		Getting and Setting IP address to and from Module



⊟ SIE	{...}		SIEMODULEDEF
⊞ SIE.DATA	{...}		SIEDATA
⊟ SIE.CONTROL	{...}		SIECONTROL
⊞ SIE.CONTROL.BootTimer	{...}		TIMER
SIE.CONTROL.ColdBoot	0	Decimal	BOOL
SIE.CONTROL.WarmBoot	0	Decimal	BOOL
SIE.CONTROL.ResetStatus	0	Decimal	BOOL
⊞ SIE.CONTROL.CmdID	0	Decimal	INT

Set Module IP Address (Block 9990)

Block Request from Processor to Module

Word Offset	Description	Length
0	9990	1
1	First digit of dotted IP address	1
2	Second digit of dotted IP address	1
3	Third digit of dotted IP address	1
4	Last digit of dotted IP address	1
5 to 247	Reserved	243

Block Response from Module to Processor

Word Offset	Description	Length
0	0	1
1	Write Block ID	1
2	First digit of dotted IP address	1
3	Second digit of dotted IP address	1
4	Third digit of dotted IP address	1
5	Last digit of dotted IP address	1
6 to 248	Spare data area	243
249	9990	1

Get Module IP Address (Block 9991)

Block Request from Processor to Module

Word Offset	Description	Length
0	9991	1
1 to 247	Spare data area	247

Block Response from Module to Processor

Word Offset	Description	Length
0	0	1
1	Write Block ID	1
2	First digit of dotted IP address	1
3	Second digit of dotted IP address	1
4	Third digit of dotted IP address	1
5	Last digit of dotted IP address	1
6 to 248	Spare data area	243
249	9991	1

Warm Boot (Block 9998)

This block is sent from the ControlLogix processor to the module (output image) when the module is required to perform a warm-boot (software reset) operation. This block is commonly sent to the module any time configuration data modifications are made in the controller tags data area. This will cause the module to read the new configuration information and to restart.

Block Request from Processor to Module

Offset	Description	Length
0	9998	1
1 to 247	Spare	247

Cold Boot (Block 9999)

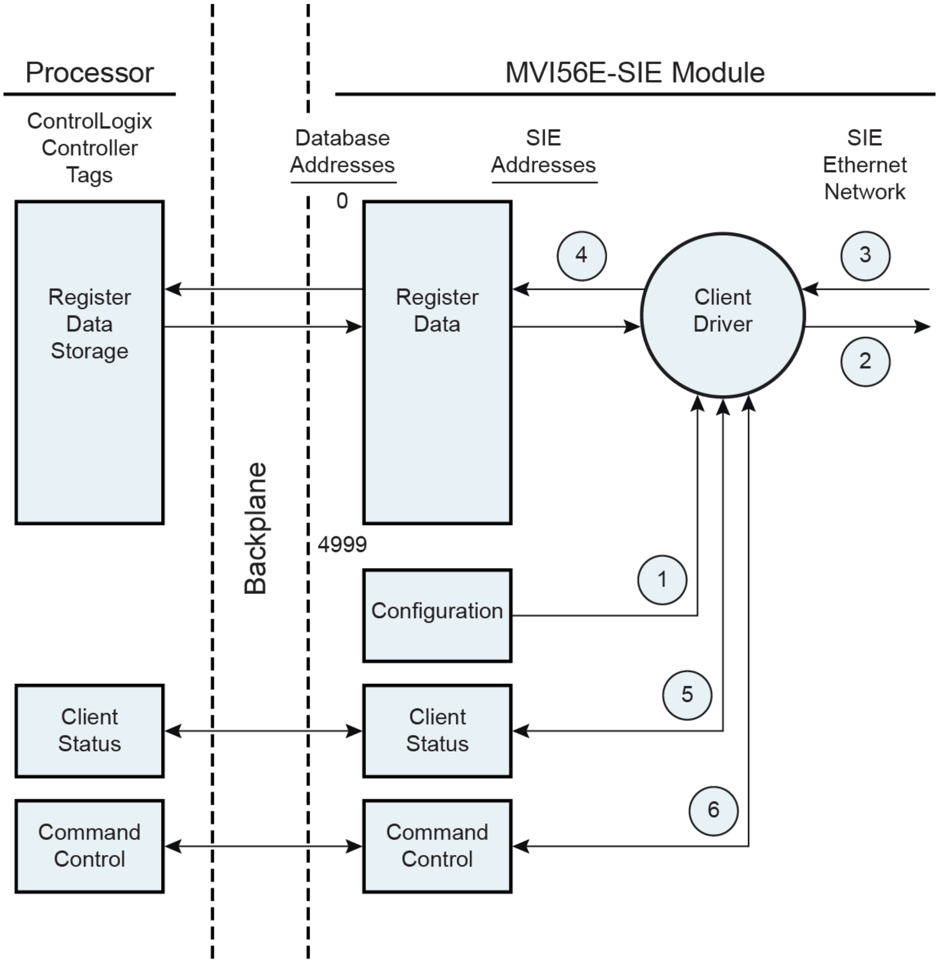
This block is sent from the ControlLogix processor to the module (output image) when the module is required to perform the cold boot (hardware reset) operation. This block is sent to the module when a hardware problem is detected by the ladder logic that requires a hardware reset.

Block Request from Processor to Module

Word Offset	Description	Length
0	9999	1
1 to 247	Spare	247

5.2.3 Client Driver

In the Client driver, the MVI56E-SIE module issues read or write commands to servers on the Siemens Industrial Ethernet network using up to 20 simulated Clients. The commands originate either from the module's user-configured *Client x Command List* for each Client. The commands from the *Client x Command List* are executed either via routine polling or as a result of special Command Control block requests from the processor. Client status data is returned to the processor in special Client Status blocks. The following flowchart describes the flow of data into and out of the module.



1 The Client driver obtains configuration data when the module restarts. This includes the timeout parameters and the Command List. These values are used by the driver to determine the types of commands to be issued to servers on the Siemens Industrial Ethernet network.

2 When configured, the Client driver begins transmitting read and/or write commands to servers on the network. The data for write commands is obtained from the module's internal database.

3 Assuming successful processing by the server specified in the command, a response message is received into the Client driver for processing.

4 Data received from the server is passed into the module's internal database, if the command was a read command. General module status information is routinely returned to the processor in the input images.

5 Status data for a specific Client can be requested by the processor and returned in a special Client Status block.

6 Special functions, such as Command Control options, can be generated by the processor and sent to the Client driver for action.

5.2.4 Client Command List

In order for the Client to function, the module's Client Command List must be defined in the *SIE Client x Commands* section of the configuration. This list contains up to 16 individual entries, with each entry containing the information required to construct a valid command. This includes the following:

- Command enable mode: (**0**) disabled or (**1**) continuous
- IP address and service port to connect to on the remote server
- Slave Node Address
- Command Type - Read or Write
- Database Source and Destination Register Address - Determines where data will be placed and/or obtained
- Count - Select the number of words to be transferred - see reference based on processor type
- Poll Delay - 1/10th seconds

For information on troubleshooting commands, see Client Command Errors (page 120).

5.3 Ethernet Cable Specifications

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

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

Ethernet cabling is like U.S. telephone cables, except that it has eight conductors. Some hubs have one input that can accept either a straight-through or crossover cable, depending on a switch position. In this case, you must ensure that the switch position and cable type agree.

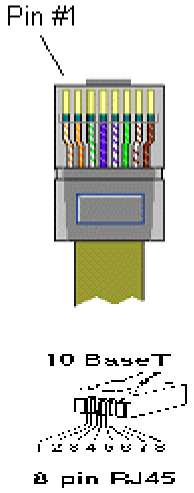
Refer to Ethernet cable configuration (page 141) for a diagram of how to configure Ethernet cable.

5.3.1 Ethernet Cable Configuration

Note: The standard connector view shown is color-coded for a straight-through cable.

Crossover cable

RJ-45 PIN	RJ-45 PIN
1 Rx+	3 Tx+
2 Rx-	6 Tx-
3 Tx+	1 Rx+
6 Tx-	2 Rx-



Straight- through cable

RJ-45 PIN	RJ-45 PIN
1 Rx+	1 Tx+
2 Rx-	2 Tx-
3 Tx+	3 Rx+
6 Tx-	6 Rx-

5.3.2 Ethernet Performance

Ethernet performance on the MVI56E-SIE module can affect the operation of the SIE application ports in the following ways.

- Accessing the web interface (refreshing the page, downloading files, and so on) may affect SIE performance
- High Ethernet traffic may impact SIE performance (consider CIPconnect (page 84) for these applications and disconnect the module Ethernet port from the network).

5.4 Using the Optional Add-On Instruction Rung Import

5.4.1 Before You Begin

- Make sure that RSLogix 5000 version 16 (or later) has been installed.
- Download the Optional Add-On file *MVI56ESIE_Optional_Rung_v1_0.L5X* from www.prosoft-technology.com
- Save a copy in a folder on the PC.

5.4.2 Overview

The Optional Add-On Instruction Rung Import contains optional logic for MVI56E-SIE applications to perform the following tasks.

- Read/Write Ethernet Configuration

  Allows the processor to read or write the module IP address, netmask and gateway values.

Note: This is an optional feature. The same task can be performed through PCB (ProSoft Configuration Builder). Even if the PC is in a different network group the module can still be accessed through PCB by setting a temporary IP address.

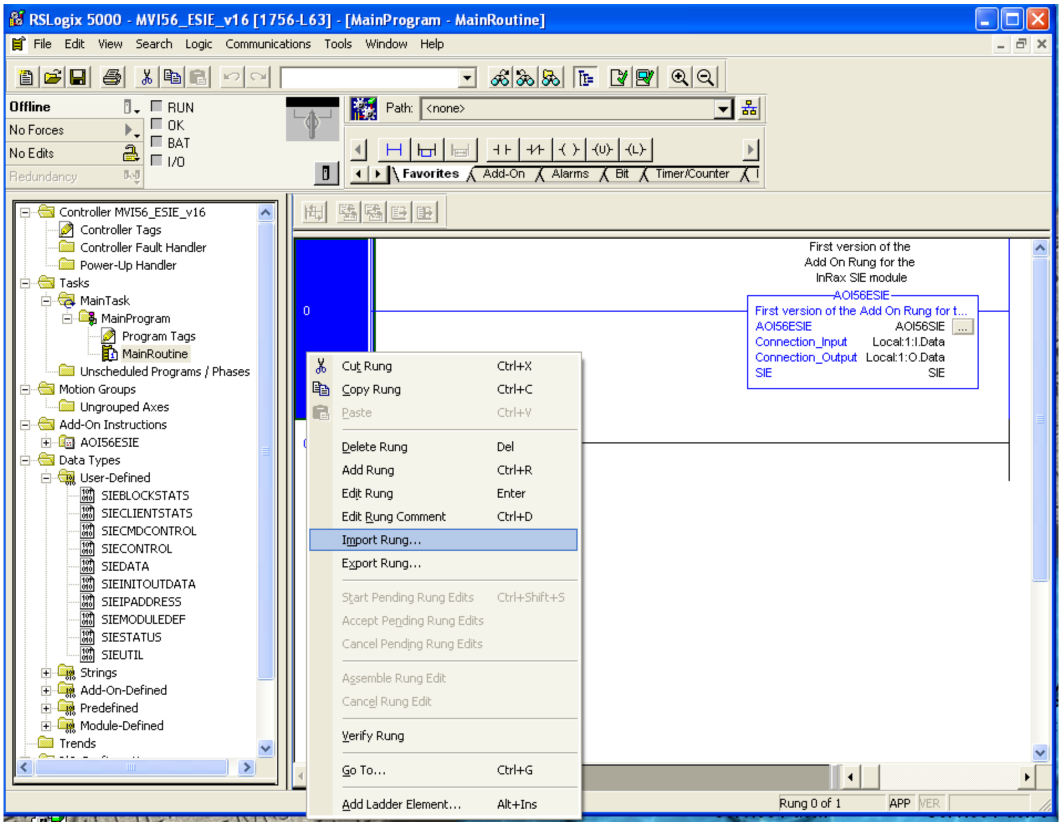
- Read/Write Module Clock Value

  Allows the processor to read and write the module clock settings. The module clock stores the last time that the Ethernet configuration was changed. The date and time of the last Ethernet configuration change is displayed in the scrolling LED during module power up.

Important: The Optional Add-On Instruction only supports the two features listed above. The sample ladder logic must be used for all other features including backplane transfer of Siemens Industrial Ethernet data.

5.4.3 Installing the Rung Import with Optional Add-On Instruction

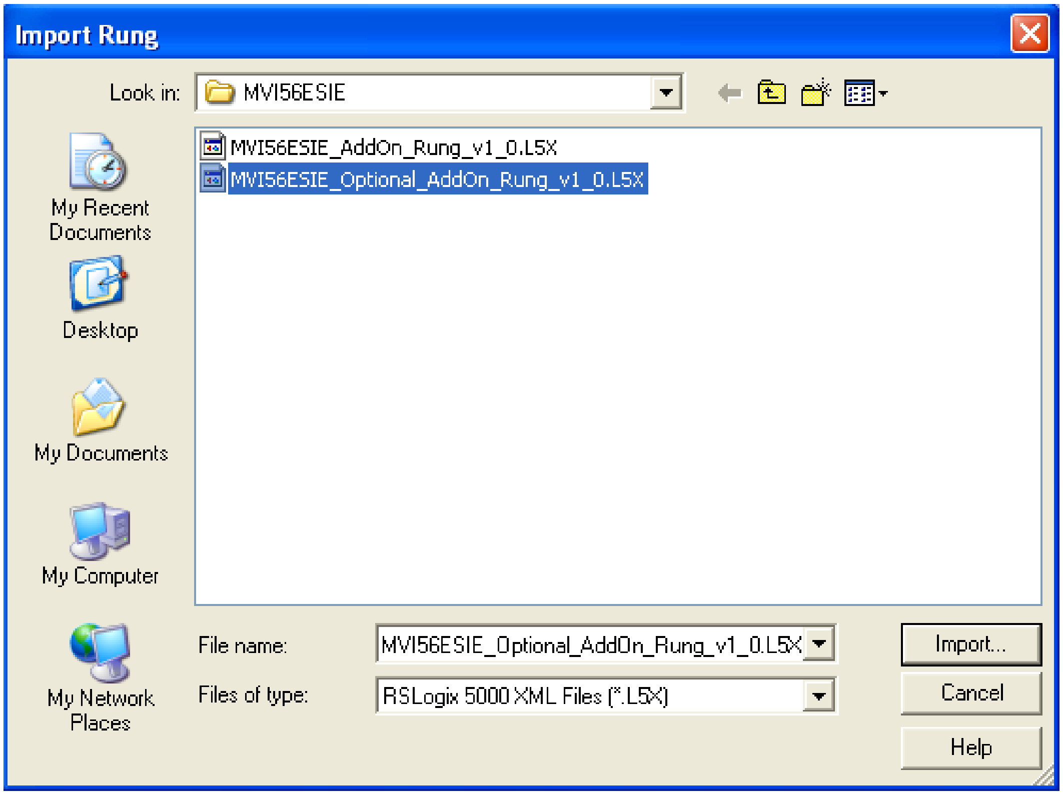
1 Right-click an empty rung in the main routine of the existing ladder logic and choose **IMPORT RUNG**.



First version of the
Add On Rung for the
InRax SIE module

AOI56ESIE
First version of the Add On Rung for t...
AOI56ESIE AOI56SIE
Connection_Input Local:1:I.Data
Connection_Output Local:1:O.Data
SIE SIE

2 Navigate to the folder where the MVI56ESIE_Optional_Rung_v1_x.L5X was saved and select the file.


Import Rung

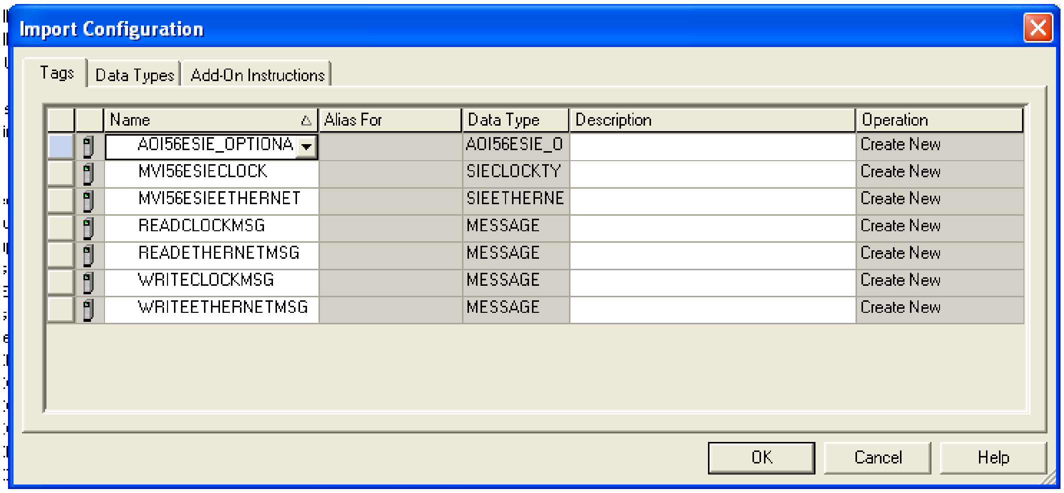
Look in: MVI56ESIE

MVI56ESIE_AddOn_Rung_v1_0.L5X
MVI56ESIE_Optional_AddOn_Rung_v1_0.L5X

File name: MVI56ESIE_Optional_AddOn_Rung_v1_0.L5X

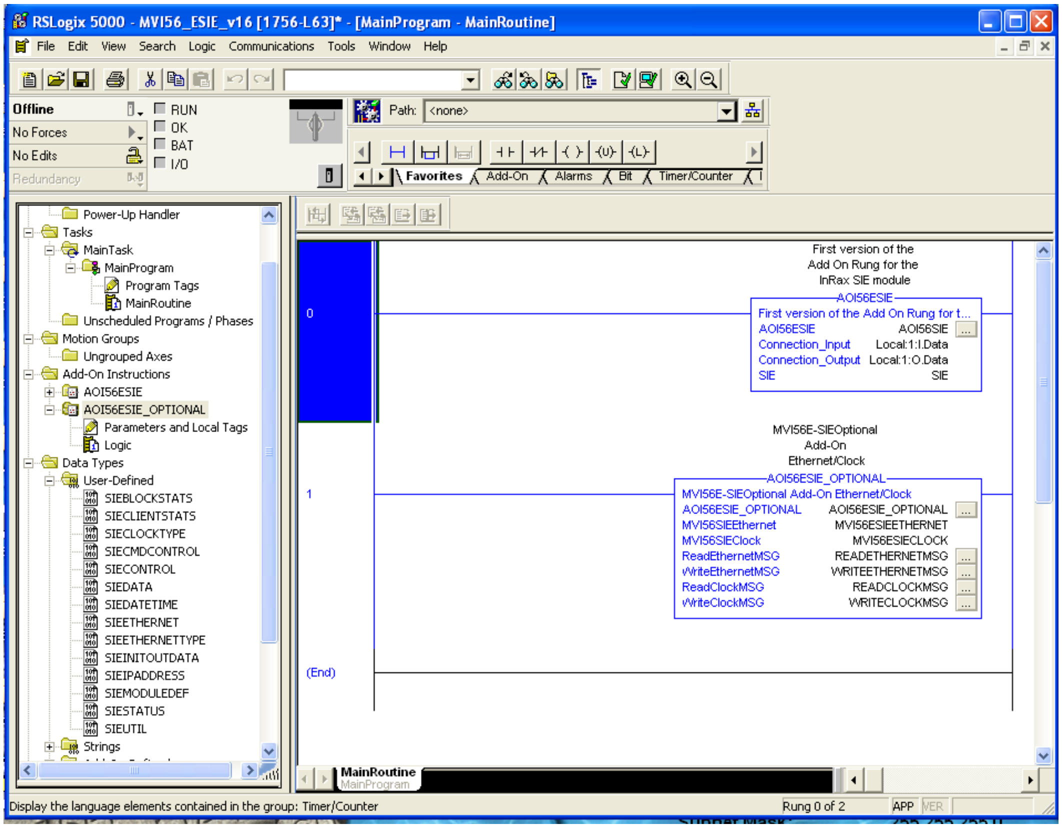
Files of type: RSLogix 5000 XML Files (*.L5X)

3 In the *Import Configuration* window, click **OK**.


Import Configuration

Name	Alias For	Data Type	Description	Operation
AOI56ESIE_OPTIONA		AOI56ESIE_O		Create New
MVI56ESIECLOCK		SIECLOCKTY		Create New
MVI56ESIEETHERNET		SIEETHERNE		Create New
READCLOCKMSG		MESSAGE		Create New
READETHERNETMSG		MESSAGE		Create New
WRITECLOCKMSG		MESSAGE		Create New
WRITEETHERNETMSG		MESSAGE		Create New

The Add-On Instruction will be now visible in the ladder logic. Observe that the procedure has also imported data types and controller tags associated with the Add-On Instruction.



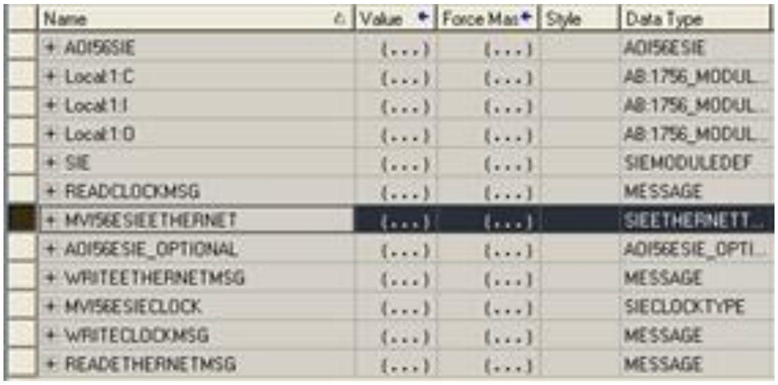
First version of the Add On Rung for the InRax SIE module

AOI56ESIE
First version of the Add On Rung for t...
AOI56ESIE AOI56ESIE
Connection_Input Local:1:I.Data
Connection_Output Local:1:O.Data
SIE SIE

MVI56E-SIEOptional Add-On Ethernet/Clock

AOI56ESIE_OPTIONAL
MVI56E-SIEOptional Add-On Ethernet/Clock
AOI56ESIE_OPTIONAL AOI56ESIE_OPTIONAL
MVI56SIEEthernet MVI56ESIEETHERNET
MVI56SIEClock MVI56ESIECLOCK
ReadEthernetMSG READETHERNETMSG
WriteEthernetMSG WRITEETHERNETMSG
ReadClockMSG READCLOCKMSG
WriteClockMSG WRITECLOCKMSG

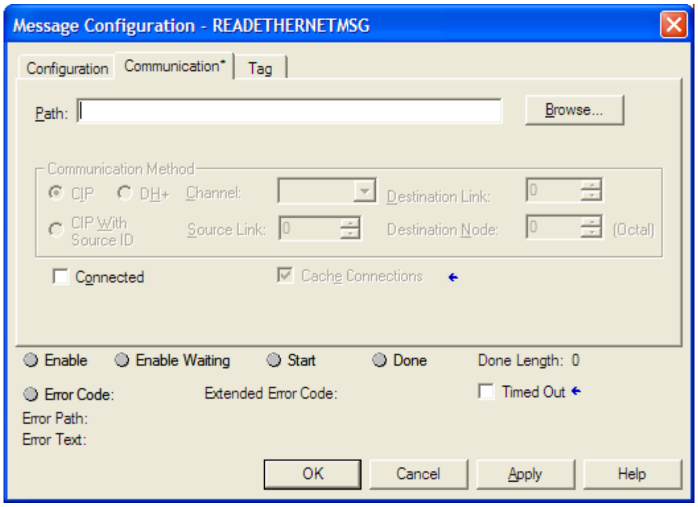
Note: New tags have been imported: *MVI56ESIEETHERNET*, *AOI56ESIE_OPTIONAL*, *MVI56ESIECLOCK*, and four *MESSAGE* tags.



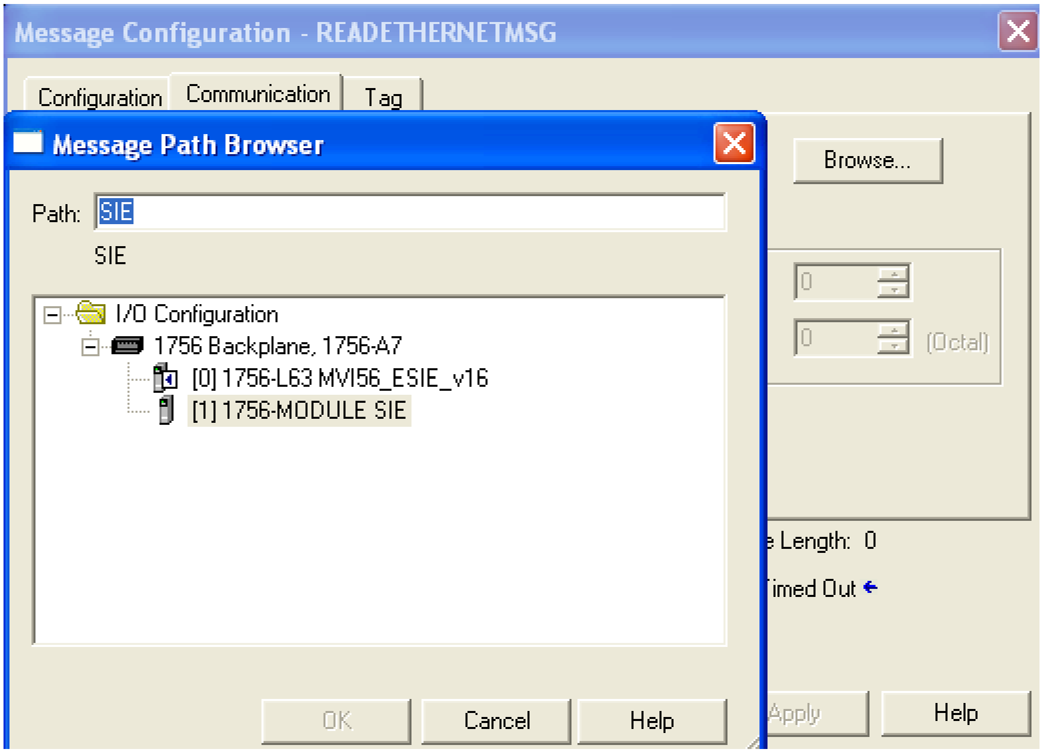
Name	Value	Force Mas	Style	Data Type
+ AOI56SIE	{...}	{...}		AOI56ESIE
+ Local:1:C	{...}	{...}		AB:1756_MODUL...
+ Local:1:I	{...}	{...}		AB:1756_MODUL...
+ Local:1:O	{...}	{...}		AB:1756_MODUL...
+ SIE	{...}	{...}		SIEMODULEDEF
+ READCLOCKMSG	{...}	{...}		MESSAGE
+ MVI56ESIEETHERNET	{...}	{...}		SIEETHERNETT...
+ AOI56ESIE_OPTIONAL	{...}	{...}		AOI56ESIE_OPTI...
+ WRITEETHERNETMSG	{...}	{...}		MESSAGE
+ MVI56ESIECLOCK	{...}	{...}		SIECLOCKTYPE
+ WRITECLOCKMSG	{...}	{...}		MESSAGE
+ READETHERNETMSG	{...}	{...}		MESSAGE

4 In the Add-On Instruction, click the **[...]** button next to any *MSG* tag to open the *Message Configuration* dialog box.

5 Click the **COMMUNICATION** tab and then click the **BROWSE** button.

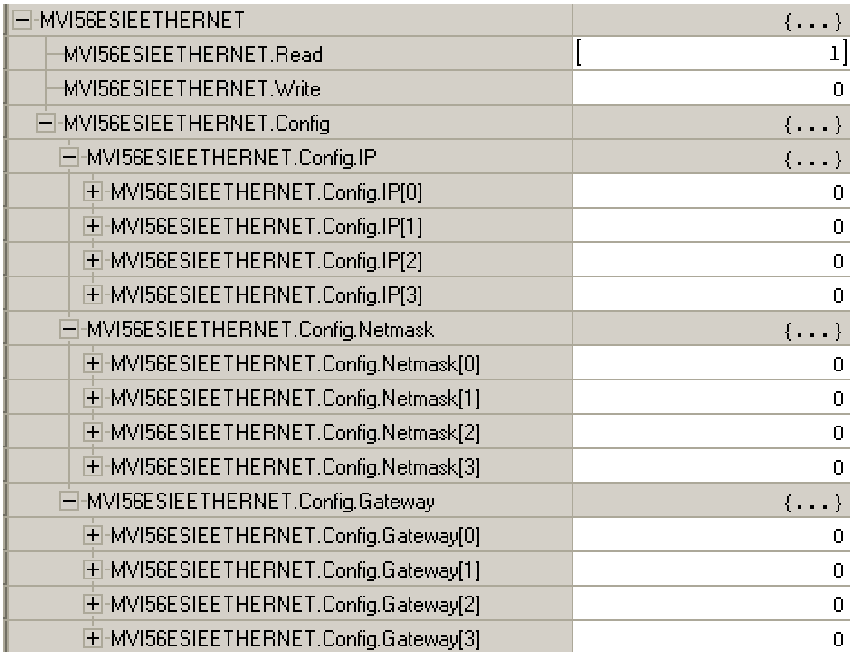


6 Select the module to configure the message path.



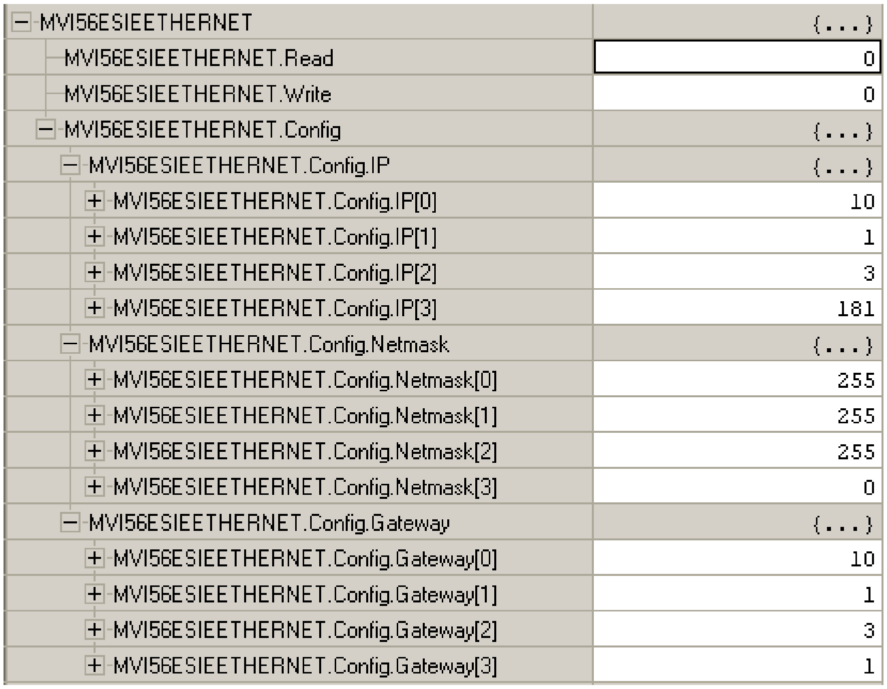
5.4.4 Reading the Ethernet Settings from the Module

Expand the *MVI56ESIEETHERNET* controller tag and move a value of **1** to *MVI56ESIEETHERNET.Read*.



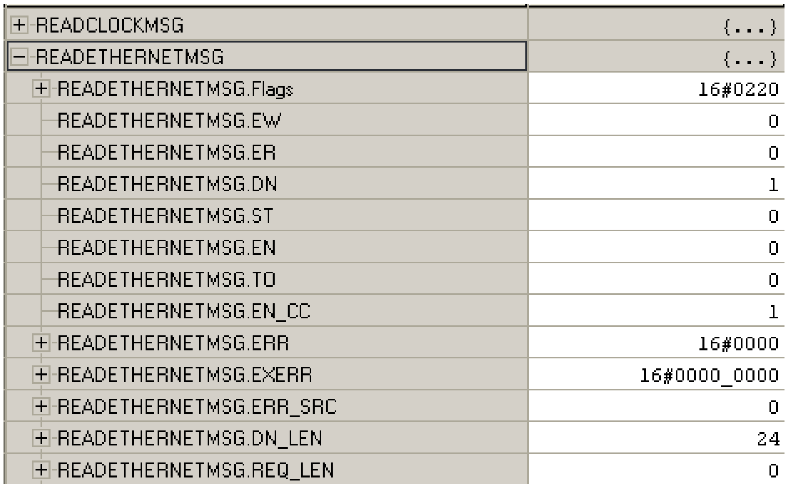
− MVI56ESIEETHERNET	{...}
MVI56ESIEETHERNET.Read	1
MVI56ESIEETHERNET.Write	0
− MVI56ESIEETHERNET.Config	{...}
− MVI56ESIEETHERNET.Config.IP	{...}
+ MVI56ESIEETHERNET.Config.IP[0]	0
+ MVI56ESIEETHERNET.Config.IP[1]	0
+ MVI56ESIEETHERNET.Config.IP[2]	0
+ MVI56ESIEETHERNET.Config.IP[3]	0
− MVI56ESIEETHERNET.Config.Netmask	{...}
+ MVI56ESIEETHERNET.Config.Netmask[0]	0
+ MVI56ESIEETHERNET.Config.Netmask[1]	0
+ MVI56ESIEETHERNET.Config.Netmask[2]	0
+ MVI56ESIEETHERNET.Config.Netmask[3]	0
− MVI56ESIEETHERNET.Config.Gateway	{...}
+ MVI56ESIEETHERNET.Config.Gateway[0]	0
+ MVI56ESIEETHERNET.Config.Gateway[1]	0
+ MVI56ESIEETHERNET.Config.Gateway[2]	0
+ MVI56ESIEETHERNET.Config.Gateway[3]	0

The bit will be automatically reset and the current Ethernet settings will be copied to *MVI56ESIEETHERNET* controller tag as follows.



⊟ MVI56ESIEETHERNET	{...}
MVI56ESIEETHERNET.Read	0
MVI56ESIEETHERNET.Write	0
⊟ MVI56ESIEETHERNET.Config	{...}
⊟ MVI56ESIEETHERNET.Config.IP	{...}
⊞ MVI56ESIEETHERNET.Config.IP[0]	10
⊞ MVI56ESIEETHERNET.Config.IP[1]	1
⊞ MVI56ESIEETHERNET.Config.IP[2]	3
⊞ MVI56ESIEETHERNET.Config.IP[3]	181
⊟ MVI56ESIEETHERNET.Config.Netmask	{...}
⊞ MVI56ESIEETHERNET.Config.Netmask[0]	255
⊞ MVI56ESIEETHERNET.Config.Netmask[1]	255
⊞ MVI56ESIEETHERNET.Config.Netmask[2]	255
⊞ MVI56ESIEETHERNET.Config.Netmask[3]	0
⊟ MVI56ESIEETHERNET.Config.Gateway	{...}
⊞ MVI56ESIEETHERNET.Config.Gateway[0]	10
⊞ MVI56ESIEETHERNET.Config.Gateway[1]	1
⊞ MVI56ESIEETHERNET.Config.Gateway[2]	3
⊞ MVI56ESIEETHERNET.Config.Gateway[3]	1

To check the status of the message, refer to the *ReadEthernetMSG* tag.



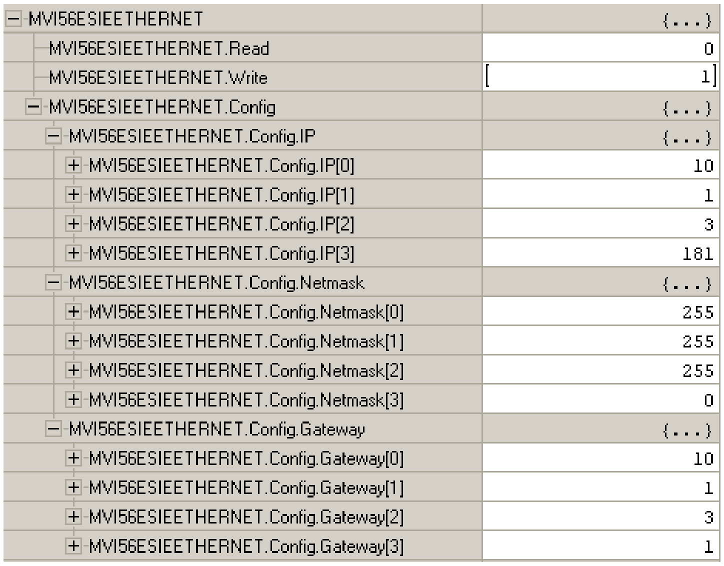
⊞ READCLOCKMSG	{...}
⊟ READETHERNETMSG	{...}
⊞ READETHERNETMSG.Flags	16#0220
READETHERNETMSG.EW	0
READETHERNETMSG.ER	0
READETHERNETMSG.DN	1
READETHERNETMSG.ST	0
READETHERNETMSG.EN	0
READETHERNETMSG.TO	0
READETHERNETMSG.EN_CC	1
⊞ READETHERNETMSG.ERR	16#0000
⊞ READETHERNETMSG.EXERR	16#0000_0000
⊞ READETHERNETMSG.ERR_SRC	0
⊞ READETHERNETMSG.DN_LEN	24
⊞ READETHERNETMSG.REQ_LEN	0

5.4.5 Writing the Ethernet Settings to the Module

Expand the *MVI56ESIEETHERNET* controller tag.

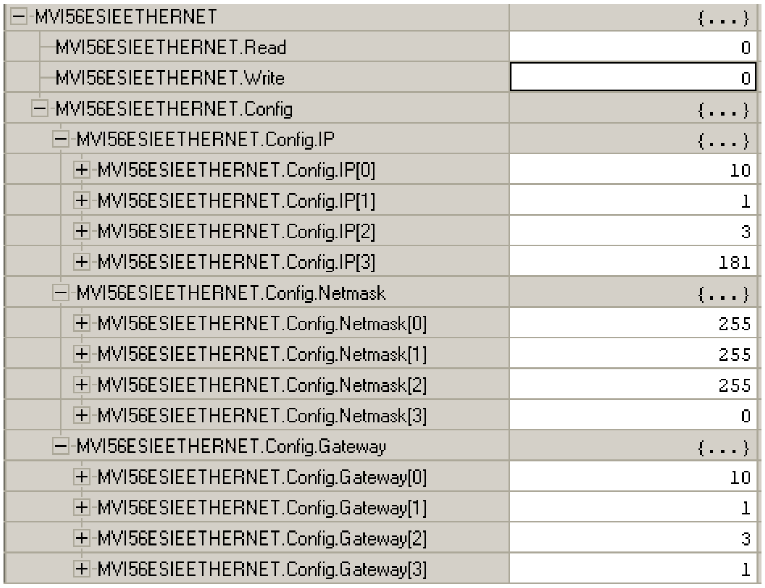
Set the new Ethernet configuration in *MVI56ESIEETHERNET.Config*:

Move a value of **1** to *MVI56ESIEETHERNET.Write*.



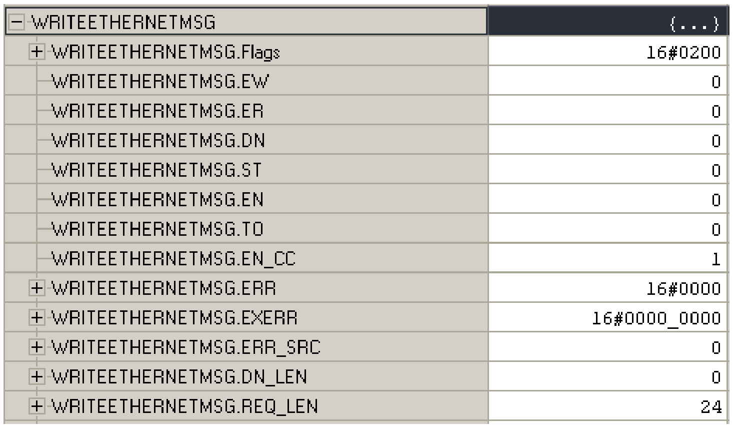
MVI56ESIEETHERNET	{...}
MVI56ESIEETHERNET.Read	0
MVI56ESIEETHERNET.Write	1
MVI56ESIEETHERNET.Config	{...}
MVI56ESIEETHERNET.Config.IP	{...}
MVI56ESIEETHERNET.Config.IP[0]	10
MVI56ESIEETHERNET.Config.IP[1]	1
MVI56ESIEETHERNET.Config.IP[2]	3
MVI56ESIEETHERNET.Config.IP[3]	181
MVI56ESIEETHERNET.Config.Netmask	{...}
MVI56ESIEETHERNET.Config.Netmask[0]	255
MVI56ESIEETHERNET.Config.Netmask[1]	255
MVI56ESIEETHERNET.Config.Netmask[2]	255
MVI56ESIEETHERNET.Config.Netmask[3]	0
MVI56ESIEETHERNET.Config.Gateway	{...}
MVI56ESIEETHERNET.Config.Gateway[0]	10
MVI56ESIEETHERNET.Config.Gateway[1]	1
MVI56ESIEETHERNET.Config.Gateway[2]	3
MVI56ESIEETHERNET.Config.Gateway[3]	1

After the message is executed, the *MVI56ESIEETHERNET.Write* bit resets to **0**.



MVI56ESIEETHERNET	{...}
MVI56ESIEETHERNET.Read	0
MVI56ESIEETHERNET.Write	0
MVI56ESIEETHERNET.Config	{...}
MVI56ESIEETHERNET.Config.IP	{...}
MVI56ESIEETHERNET.Config.IP[0]	10
MVI56ESIEETHERNET.Config.IP[1]	1
MVI56ESIEETHERNET.Config.IP[2]	3
MVI56ESIEETHERNET.Config.IP[3]	181
MVI56ESIEETHERNET.Config.Netmask	{...}
MVI56ESIEETHERNET.Config.Netmask[0]	255
MVI56ESIEETHERNET.Config.Netmask[1]	255
MVI56ESIEETHERNET.Config.Netmask[2]	255
MVI56ESIEETHERNET.Config.Netmask[3]	0
MVI56ESIEETHERNET.Config.Gateway	{...}
MVI56ESIEETHERNET.Config.Gateway[0]	10
MVI56ESIEETHERNET.Config.Gateway[1]	1
MVI56ESIEETHERNET.Config.Gateway[2]	3
MVI56ESIEETHERNET.Config.Gateway[3]	1

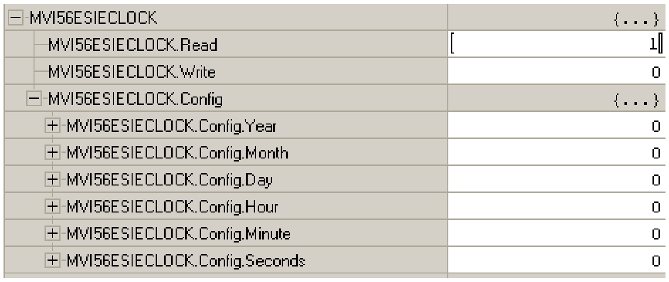
To check the status of the message, refer to the *WriteEthernetMSG* tag.



⊟ WRITEETHERNETMSG	{...}
⊞ WRITEETHERNETMSG.Flags	16#0200
WRITEETHERNETMSG.EW	0
WRITEETHERNETMSG.ER	0
WRITEETHERNETMSG.DN	0
WRITEETHERNETMSG.ST	0
WRITEETHERNETMSG.EN	0
WRITEETHERNETMSG.TO	0
WRITEETHERNETMSG.EN_CC	1
⊞ WRITEETHERNETMSG.ERR	16#0000
⊞ WRITEETHERNETMSG.EXERR	16#0000_0000
⊞ WRITEETHERNETMSG.ERR_SRC	0
⊞ WRITEETHERNETMSG.DN_LEN	0
⊞ WRITEETHERNETMSG.REQ_LEN	24

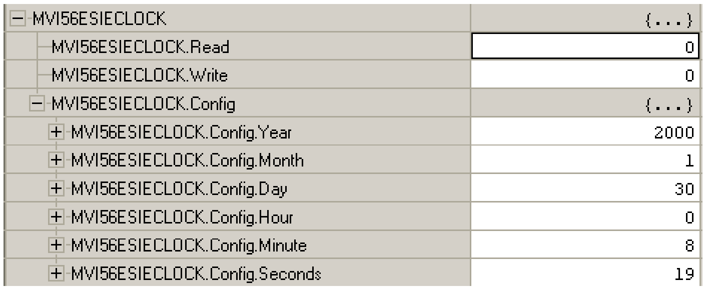
5.4.6 Reading the Clock Value from the Module

Expand the *MVI56ESIECLOCK* controller tag and move a value of **1** to *MVI56ESIECLOCK.Read*



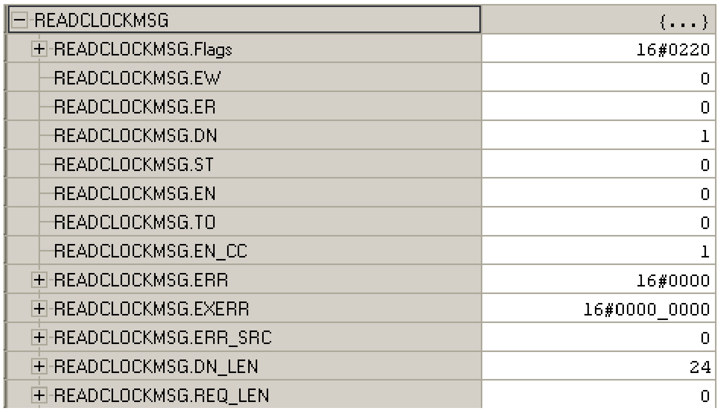
⊟ MVI56ESIECLOCK	{...}
MVI56ESIECLOCK.Read	1
MVI56ESIECLOCK.Write	0
⊟ MVI56ESIECLOCK.Config	{...}
⊞ MVI56ESIECLOCK.Config.Year	0
⊞ MVI56ESIECLOCK.Config.Month	0
⊞ MVI56ESIECLOCK.Config.Day	0
⊞ MVI56ESIECLOCK.Config.Hour	0
⊞ MVI56ESIECLOCK.Config.Minute	0
⊞ MVI56ESIECLOCK.Config.Seconds	0

The bit will be automatically reset and the current clock value will be copied to *MVI56ESIECLOCK.Config* controller tag as follows.



⊟ MVI56ESIECLOCK	{...}
MVI56ESIECLOCK.Read	0
MVI56ESIECLOCK.Write	0
⊟ MVI56ESIECLOCK.Config	{...}
⊞ MVI56ESIECLOCK.Config.Year	2000
⊞ MVI56ESIECLOCK.Config.Month	1
⊞ MVI56ESIECLOCK.Config.Day	30
⊞ MVI56ESIECLOCK.Config.Hour	0
⊞ MVI56ESIECLOCK.Config.Minute	8
⊞ MVI56ESIECLOCK.Config.Seconds	19

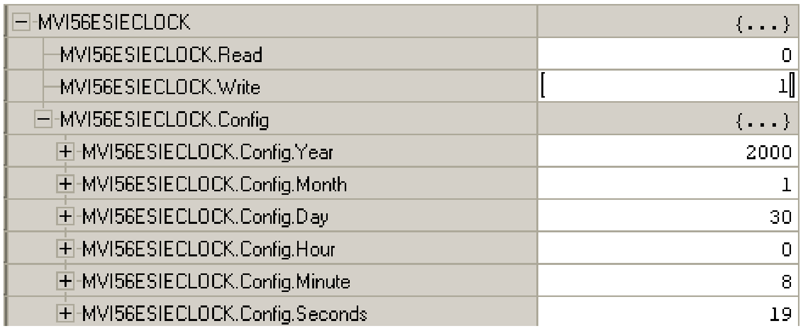
To check the status of the message, refer to the *ReadClockMSG* tag.



⊟ READCLOCKMSG	{...}
⊞ READCLOCKMSG.Flags	16#0220
READCLOCKMSG.EW	0
READCLOCKMSG.ER	0
READCLOCKMSG.DN	1
READCLOCKMSG.ST	0
READCLOCKMSG.EN	0
READCLOCKMSG.TO	0
READCLOCKMSG.EN_CC	1
⊞ READCLOCKMSG.ERR	16#0000
⊞ READCLOCKMSG.EXERR	16#0000_0000
⊞ READCLOCKMSG.ERR_SRC	0
⊞ READCLOCKMSG.DN_LEN	24
⊞ READCLOCKMSG.REQ_LEN	0

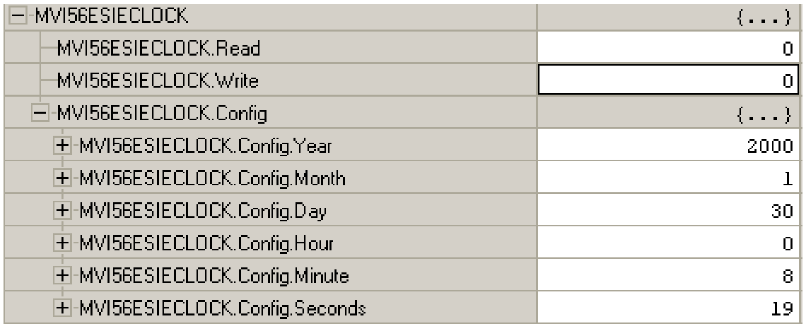
5.4.7 *Writing the Clock Value to the Module*

Expand the *MVI56ESIECLOCK* controller tag.
Set the new Clock value in *MVI56ESIECLOCK.Config*:
Move a value of **1** to *MVI56ESIECLOCK.Write*.



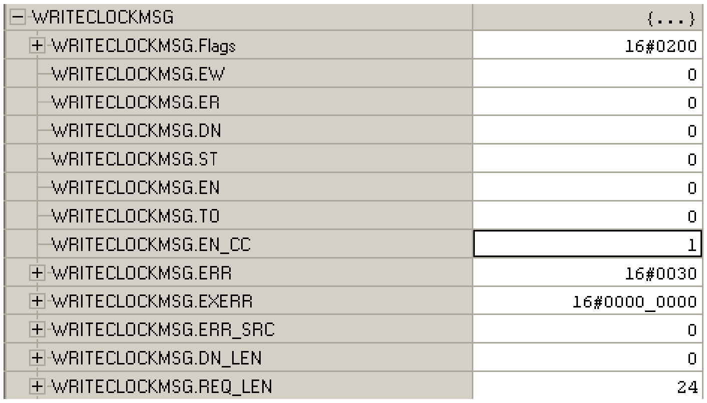
⊟ MVI56ESIECLOCK	{...}
MVI56ESIECLOCK.Read	0
MVI56ESIECLOCK.Write	1
⊟ MVI56ESIECLOCK.Config	{...}
⊞ MVI56ESIECLOCK.Config.Year	2000
⊞ MVI56ESIECLOCK.Config.Month	1
⊞ MVI56ESIECLOCK.Config.Day	30
⊞ MVI56ESIECLOCK.Config.Hour	0
⊞ MVI56ESIECLOCK.Config.Minute	8
⊞ MVI56ESIECLOCK.Config.Seconds	19

The bit will be automatically reset to **0**.



⊟ MVI56ESIECLOCK	{...}
MVI56ESIECLOCK.Read	0
MVI56ESIECLOCK.Write	0
⊟ MVI56ESIECLOCK.Config	{...}
⊞ MVI56ESIECLOCK.Config.Year	2000
⊞ MVI56ESIECLOCK.Config.Month	1
⊞ MVI56ESIECLOCK.Config.Day	30
⊞ MVI56ESIECLOCK.Config.Hour	0
⊞ MVI56ESIECLOCK.Config.Minute	8
⊞ MVI56ESIECLOCK.Config.Seconds	19

To check the status of the message, refer to the *WriteClockMSG* tag.



⊟ WRITECLOCKMSG	{...}
⊞ WRITECLOCKMSG.Flags	16#0200
WRITECLOCKMSG.EW	0
WRITECLOCKMSG.ER	0
WRITECLOCKMSG.DN	0
WRITECLOCKMSG.ST	0
WRITECLOCKMSG.EN	0
WRITECLOCKMSG.TO	0
WRITECLOCKMSG.EN_CC	1
⊞ WRITECLOCKMSG.ERR	16#0030
⊞ WRITECLOCKMSG.EXERR	16#0000_0000
⊞ WRITECLOCKMSG.ERR_SRC	0
⊞ WRITECLOCKMSG.DN_LEN	0
⊞ WRITECLOCKMSG.REQ_LEN	24

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 Technology's 24/7 after-hours phone support is available for urgent plant-down issues.

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

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

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