Where Automation Connects.

MVI69E-GEC

CompactLogix™ or MicroLogix™ Platform

Generic ASCII Ethernet Communication Module

February 24, 2020

USER MANUAL

Your Feedback Please

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

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

MVI69E-GEC User Manual

February 24, 2020

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided at our website:
www.prosoft-technology.com

Content Disclaimer

Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.

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

Important Safety Information

North America Warnings

A **WARNING** - EXPLOSION HAZARD - SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS I, DIV. 2;
B **WARNING** - EXPLOSION HAZARD - WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES.
C **WARNING** - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.
D **Class 2 Power**

ATEX/IECEx Warnings and Conditions of Safe Usage:

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

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

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

Agency Approvals & Certifications

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

Contents

1 Start Here

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

- **Studio 5000 Logix Designer ®:** 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 GEC and CompactLogix or MicroLogix devices to a power source and to the MVI69E-GEC.

1.1 System Requirements

The MVI69E-GEC module requires the following minimum hardware and software components:

- Rockwell Automation CompactLogix or MicroLogix 1500-LRP® processor (firmware version 10 or higher), with compatible power supply and one free slot in the rack, for the MVI69E-GEC module.

Important: The MVI69E-GEC module has a power supply distance rating of 4 (L43 and L45 installations on first 2 slots of 1769 bus). It consumes 500 mA at 5 VDC.

Important: For 1769-L23x processors, please make note of the following limitation:
1769-L23E-QBFC1B = 450 mA at 5 VDC (No MVI69E module can be used with this processor.)

- The module requires 500 mA of available 5 VDC power
- Rockwell Automation Studio 5000 Logix Designer programming software version 16 or higher
- 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® 8
  - Microsoft Windows® 7
  - Microsoft Windows Vista
  - Microsoft Windows XP Professional with Service Pack 1 or 2
- 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.

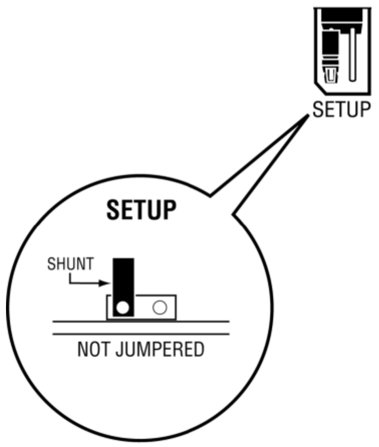
1.2 Package Contents

Qty.	Part Name	Part Number	Part Description
1	MVI69E-GEC Module	MVI69E-GEC	Generic ASCII Serial Communication Module

1.3 Setup Jumper

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. If an update of the firmware is needed, apply the Setup jumper to both pins.

The following illustration shows the MVI69E-GEC jumper configuration, with the Setup Jumper OFF.



1.4 Install the Module in the Rack

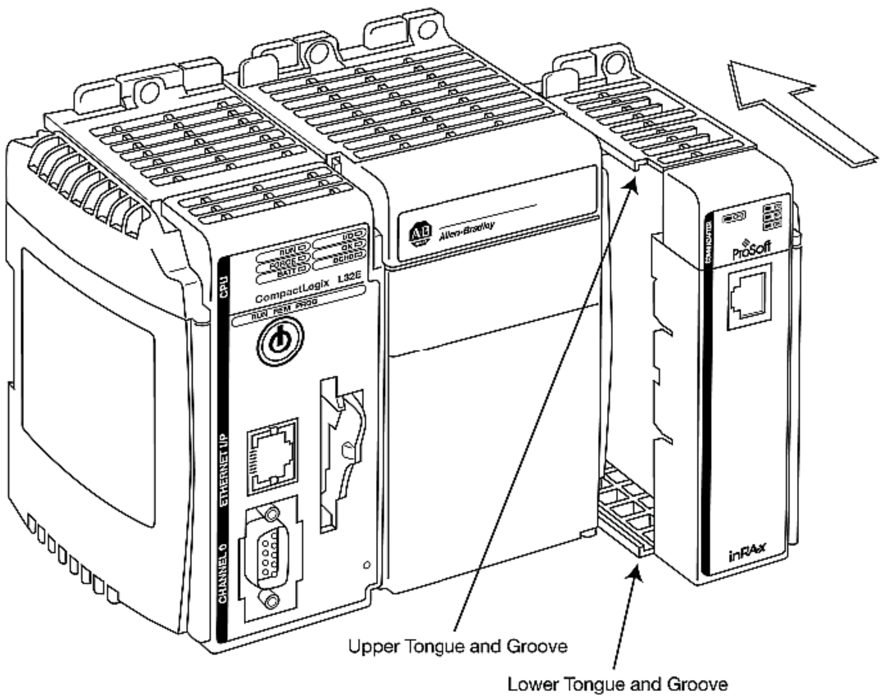
Make sure the processor and power supply are installed and configured before installing the MVI69E-GEC module. Refer to the Rockwell Automation product documentation for installation instructions.

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

After you verify the jumper placements, insert the MVI69E-GEC into the rack. Use the same technique recommended by Rockwell Automation to remove and install CompactLogix or MicroLogix 1500-LRP modules.

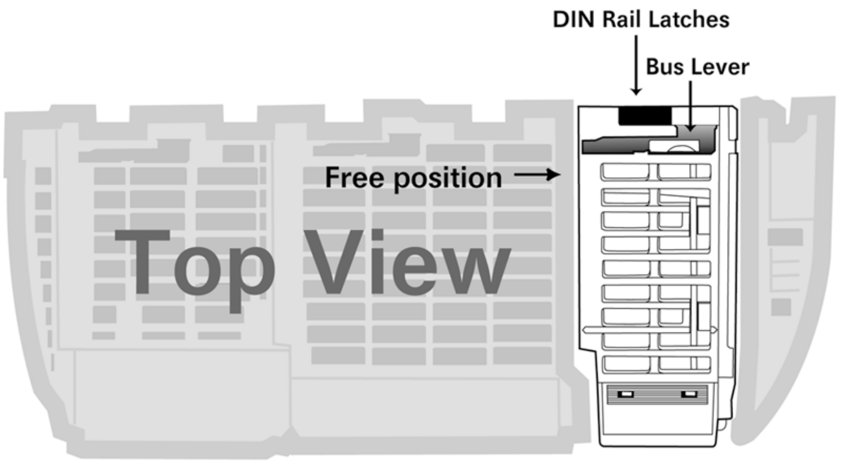
Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

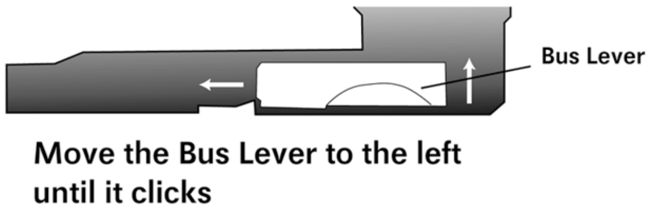
1 Align the module using the upper and lower tongue-and-groove slots with the adjacent module and slide forward in the direction of the arrow.

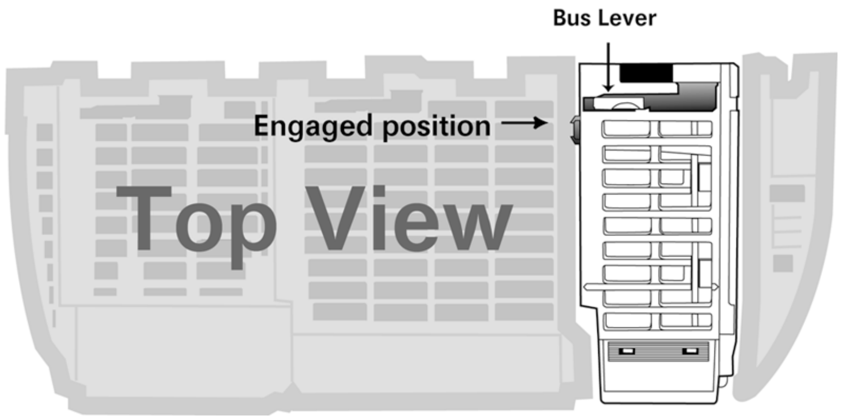


2 Move the module back along the tongue-and-groove slots until the bus connectors on the MVI69 module and the adjacent module line up with each other.

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

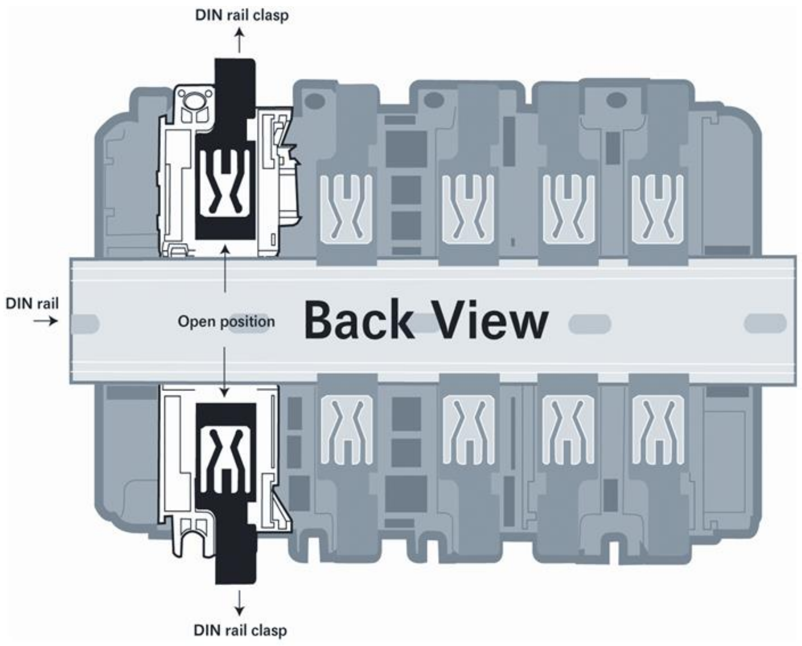


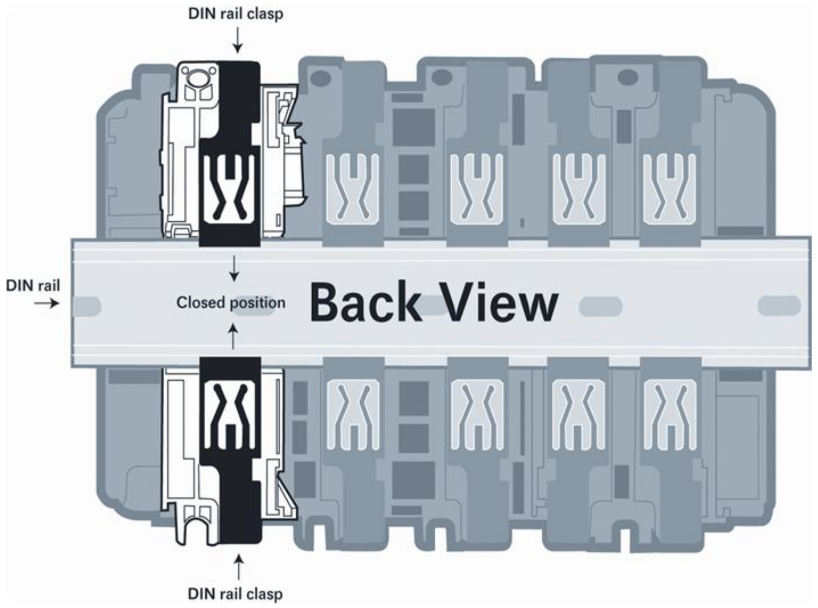




4 Close all DIN-rail latches.

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





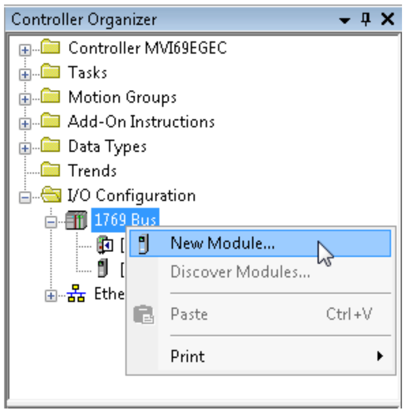
2 Configuring the Module in Studio 5000

2.1 Configuring the MVI69E-GEC in Studio 5000

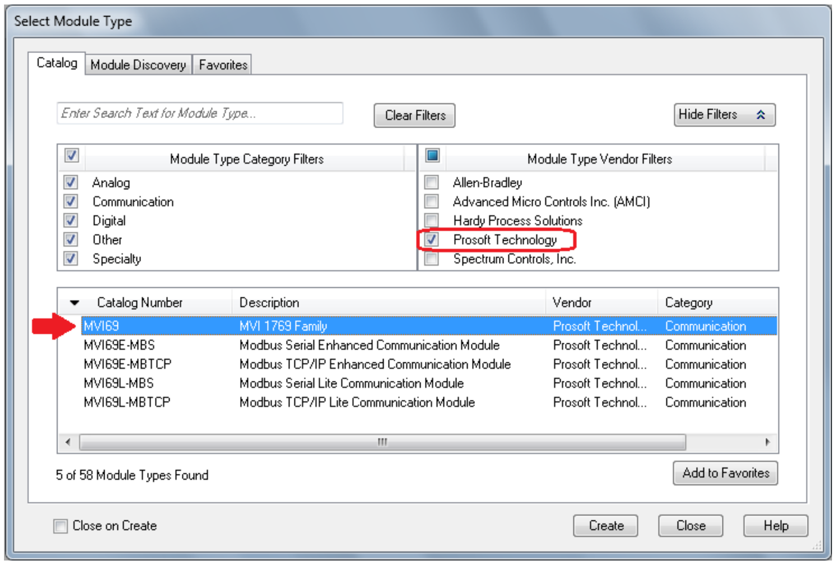
Important: The MVI69E-GEC module has a power supply distance rating of 4 (L43 and L45 installations on first 2 slots of 1769 bus)

If you are installing and configuring the module with a CompactLogix processor, follow these steps. If you are using a MicroLogix processor, refer to Adding the Module to an Existing MicroLogix Project (page 13).

1 Right-click the mouse button on the *I/O Configuration* option in the *Controller Organizer* window to display a pop-up menu. Select the **NEW MODULE** option.



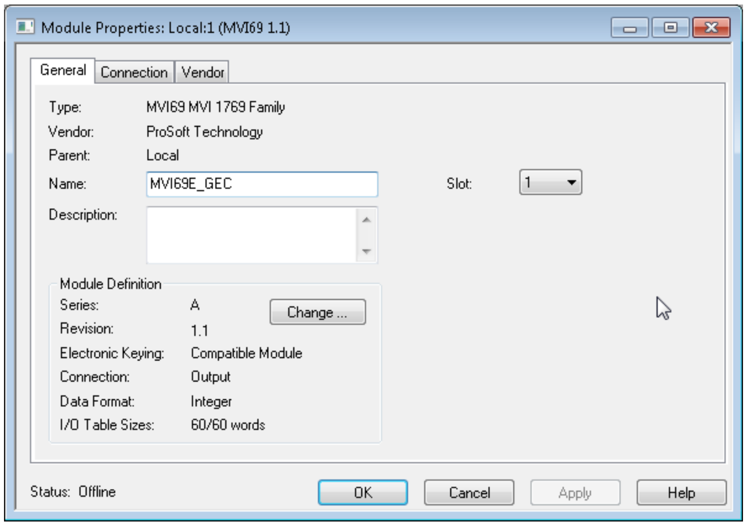
2 Under the *Module Type Vendor Filters* section, select **ProSoft Technology**. Then select the **MVI69** option and click the **CREATE** button.



Catalog Number	Description	Vendor	Category
MVI69	MVI 1769 Family	Prosoft Technol...	Communication
MVI69E-MBS	Modbus Serial Enhanced Communication Module	Prosoft Technol...	Communication
MVI69E-MBTCP	Modbus TCP/IP Enhanced Communication Module	Prosoft Technol...	Communication
MVI69L-MBS	Modbus Serial Lite Communication Module	Prosoft Technol...	Communication
MVI69L-MBTCP	Modbus TCP/IP Lite Communication Module	Prosoft Technol...	Communication

5 of 58 Module Types Found

3 In the *New Module* dialog, enter the *Name*, *Description*, and *Slot* options for your application, then click on the *Connection* tab.

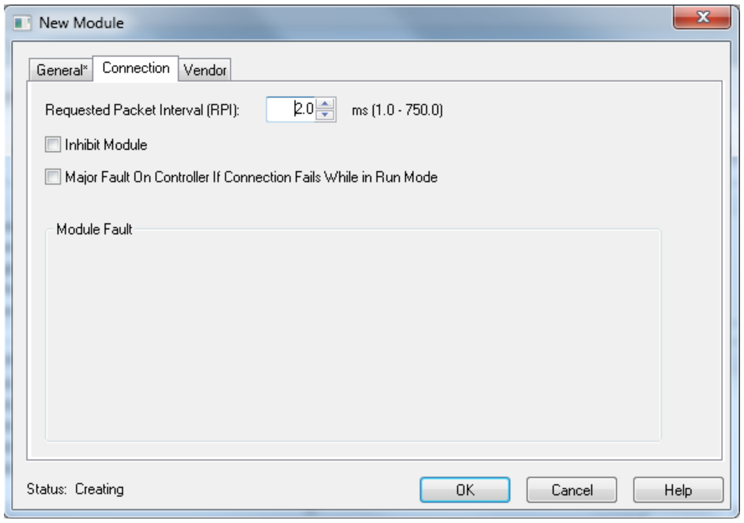


Type: MVI69 MVI 1769 Family
Vendor: ProSoft Technology
Parent: Local
Name: MVI69E_GEC
Slot: 1
Description:

Module Definition
Series: A
Revision: 1.1
Electronic Keying: Compatible Module
Connection: Output
Data Format: Integer
I/O Table Sizes: 60/60 words

Status: Offline

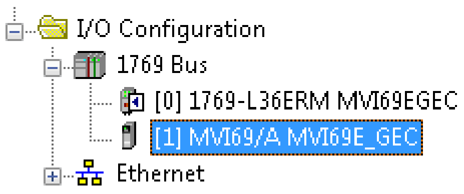
4 In the *Connection* tab, select the *Request Packet Interval* value for scanning the I/O on the module. This value represents the minimum frequency the module will handle scheduled events. This value should not be set to less than 1 millisecond. Values between 1 and 10 milliseconds should work with most applications. Click **OK**.



Requested Packet Interval (RPI): 2.0 ms (1.0 - 750.0)
Inhibit Module
Major Fault On Controller If Connection Fails While in Run Mode

Status: Creating

5 The *I/O Configuration* now displays the module's presence.

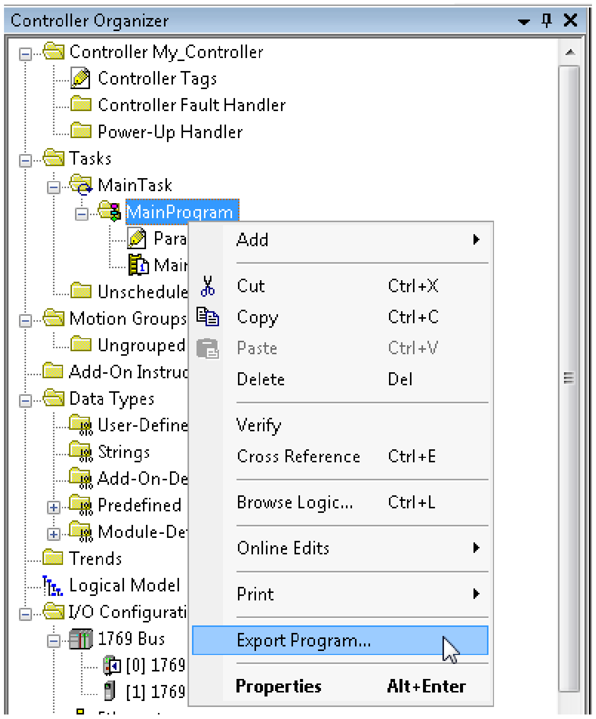


2.2 Ladder Logic

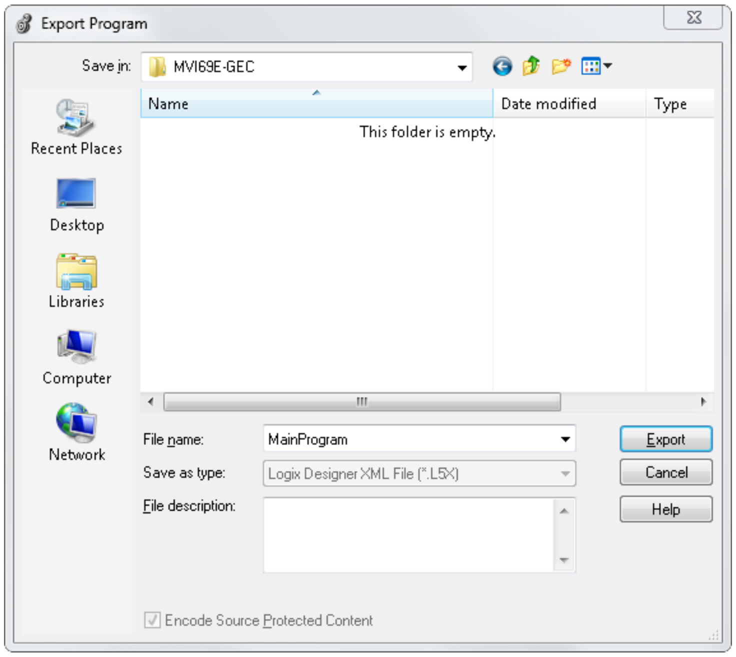
The MVI69E-GEC ladder logic, Controller Tags, and UDT's must be imported into your existing Studio 5000 project. To do this, the sample program can be exported to a .L5X file and imported into the existing Studio 5000 project.
You may use the sample ladder logic as-is; or import the project components into your existing applications.

2.2.1 Exporting the MVI69E-GEC .L5X File

1 Download the sample ladder logic program (*MVI69EGEC.ACD*) from www.prosoft-technology.com
2 Open the *MVI69EGEC.ACD* file.
3 In the *Controller Organizer* window, right-click on the *Tasks > Main Task > MainProgram* folder. Click on the **EXPORT PROGRAM** option.

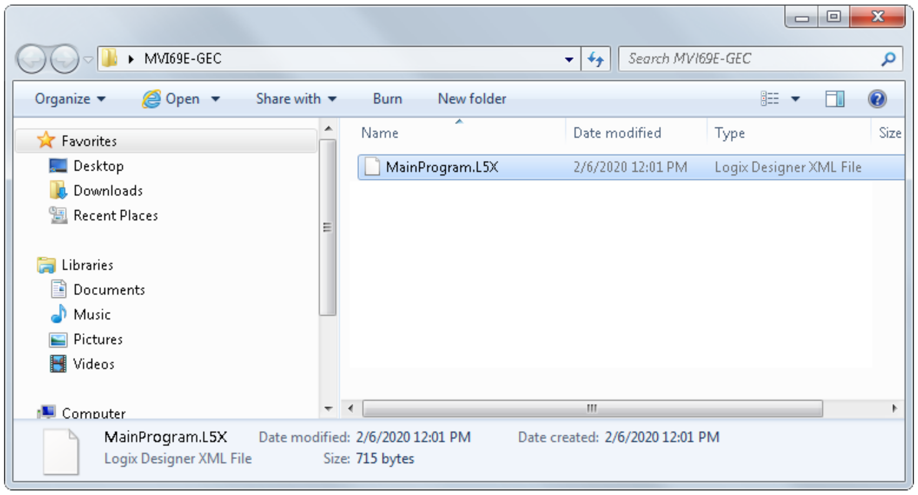


4 Select a storage location to save the *MainProgram.L5X* file. Click the **EXPORT** button.



File name: MainProgram
Save as type: Logix Designer XML File (*.L5X)
File description:
Encode Source Protected Content

5 Verify the *MainProgram.L5X* file has been successfully exported.

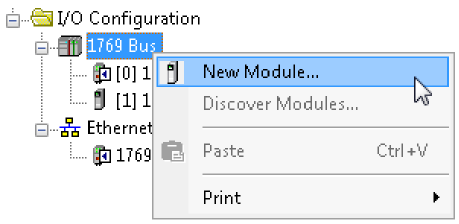


Name	Date modified	Type	Size
MainProgram.L5X	2/6/2020 12:01 PM	Logix Designer XML File	

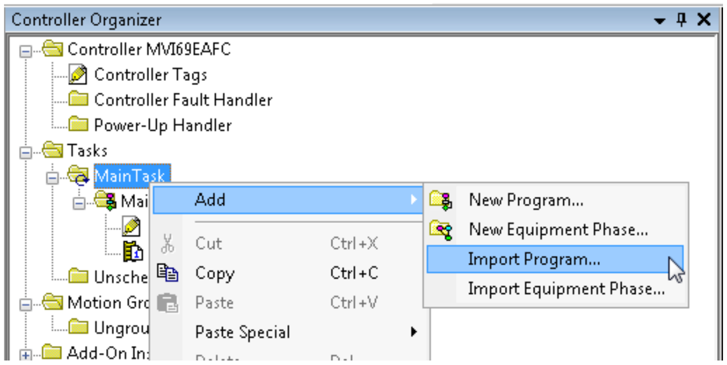
MainProgram.L5X Date modified: 2/6/2020 12:01 PM Date created: 2/6/2020 12:01 PM
Logix Designer XML File Size: 715 bytes

2.2.2 Importing the MVI69E-GEC .L5X file Into an Existing Studio 5000 Project

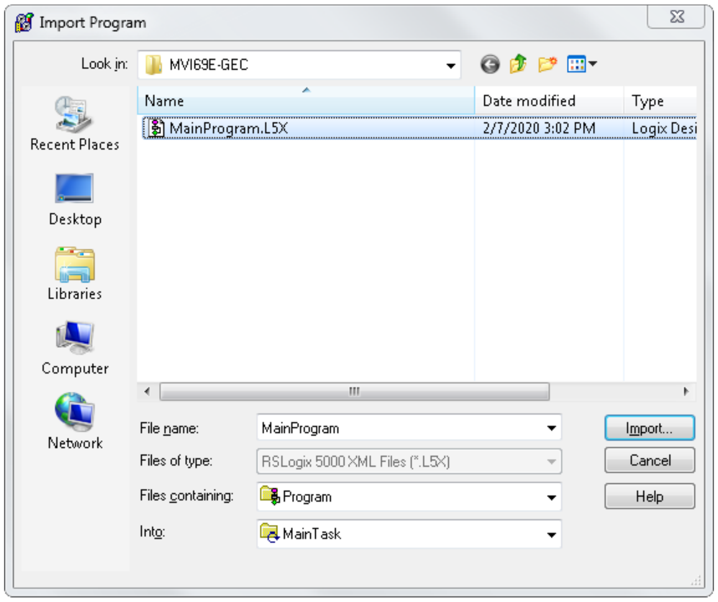
1 Within the existing Studio 5000 Project, add the MVI69E-GEC module to the *I/O Configuration*. (For further instructions, see Configuring the MVI69E-GEC in Studio 5000 on page 11).



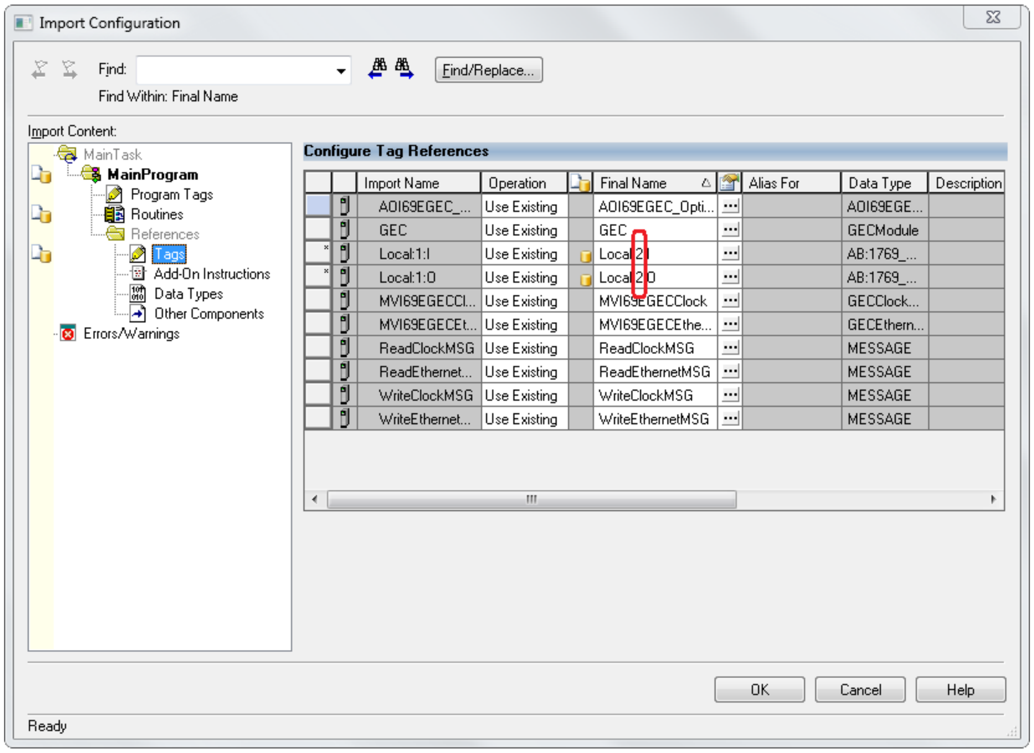
2 In the *Controller Organizer* window, right-click on the *Tasks > Main Task* folder. Click on *Add* > ***Import Program*** option.



3 Select the (previously exported) MVI69E-GEC *MainProgram.L5X* file and click the **IMPORT** button.



4 In the *Import Configuration* dialog, edit the **Local:x:I** and **Local:x:O** tags to correspond to the slot position of the MVI69E-GEC. Then click **OK**.


Import Configuration

Find:

Find Within: Final Name

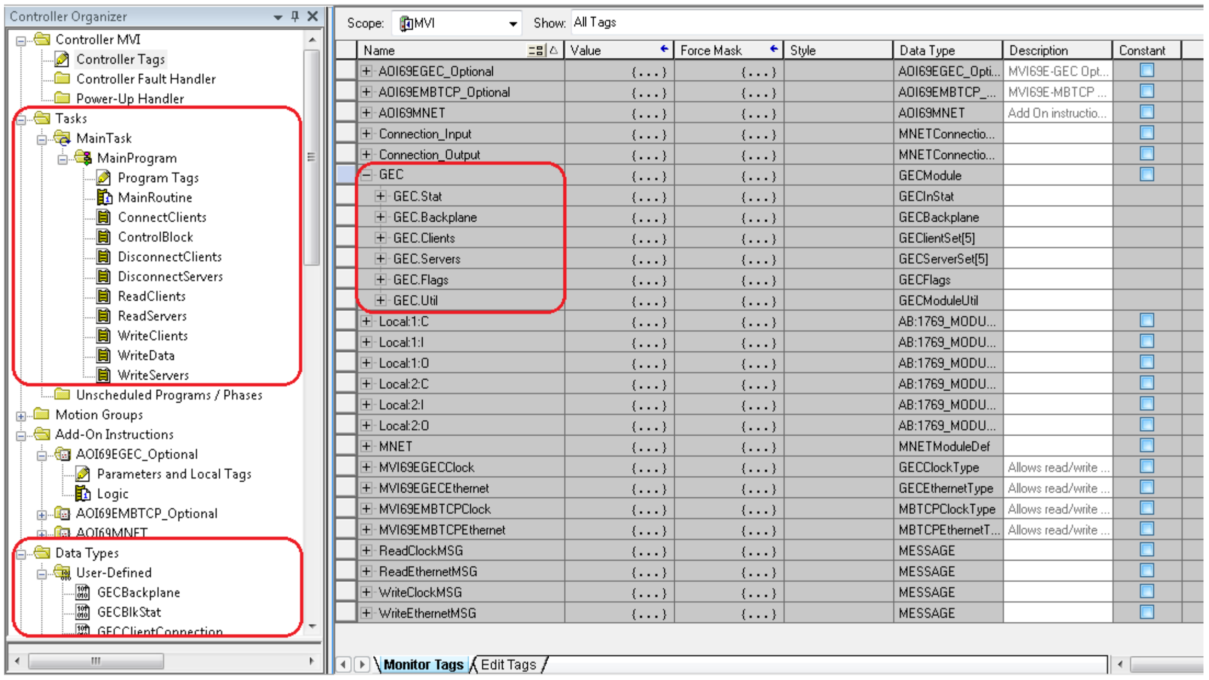
Import Content:

- MainTask
  - MainProgram
    - Program Tags
    - Routines
    - References
      - Tags
      - Add-On Instructions
      - Data Types
      - Other Components
- Errors/Warnings

Configure Tag References

Import Name	Operation	Final Name	Alias For	Data Type	Description
AOI69EGEC_...	Use Existing	AOI69EGEC_Opti...		AOI69EGE...	
GEC	Use Existing	GEC		GECModule	
Local:1:I	Use Existing	Local:2:I		AB:1769_...	
Local:1:O	Use Existing	Local:2:O		AB:1769_...	
MVI69EGECCl...	Use Existing	MVI69EGECClock		GECClock...	
MVI69EGECEt...	Use Existing	MVI69EGECEthe...		GECEthern...	
ReadClockMSG	Use Existing	ReadClockMSG		MESSAGE	
ReadEthernet...	Use Existing	ReadEthernetMSG		MESSAGE	
WriteClockMSG	Use Existing	WriteClockMSG		MESSAGE	
WriteEthernet...	Use Existing	WriteEthernetMSG		MESSAGE	

5 Upon successful import, the MVI69E-GEC ladder logic, Controller Tags, and UDT's are now visible in the Studio 5000 project.


Controller Organizer

- Controller MVI
  - Controller Tags
  - Controller Fault Handler
  - Power-Up Handler
- Tasks
  - MainTask
    - MainProgram
      - Program Tags
      - MainRoutine
      - ConnectClients
      - ControlBlock
      - DisconnectClients
      - DisconnectServers
      - ReadClients
      - ReadServers
      - WriteClients
      - WriteData
      - WriteServers
  - Unscheduled Programs / Phases
- Motion Groups
- Add-On Instructions
  - AOI69EGEC_Optional
    - Parameters and Local Tags
    - Logic
  - AOI69EMBTCP_Optional
  - AOI69MNET
- Data Types
  - User-Defined
    - GECBackplane
    - GECBlkStat
    - GECClientConnection

Scope: MVI Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description	Constant
+ AOI69EGEC_Optional	{...}	{...}		AOI69EGEC_Opti...	MVI69E-GEC Opt...	☐
+ AOI69EMBTCP_Optional	{...}	{...}		AOI69EMBTCP_...	MVI69E-MBTCP ...	☐
+ AOI69MNET	{...}	{...}		AOI69MNET	Add On instructio...	☐
+ Connection_Input	{...}	{...}		MNETConnectio...		☐
+ Connection_Output	{...}	{...}		MNETConnectio...		☐
− GEC	{...}	{...}		GECModule		☐
+ GEC.Stat	{...}	{...}		GECInStat		
+ GEC.Backplane	{...}	{...}		GECBackplane		
+ GEC.Clients	{...}	{...}		GEClientSet[5]		
+ GEC.Servers	{...}	{...}		GECServerSet[5]		
+ GEC.Flags	{...}	{...}		GECFlags		
+ GEC.Util	{...}	{...}		GECModuleUtil		
+ Local:1:C	{...}	{...}		AB:1769_MODU...		☐
+ Local:1:I	{...}	{...}		AB:1769_MODU...		☐
+ Local:1:O	{...}	{...}		AB:1769_MODU...		☐
+ Local:2:C	{...}	{...}		AB:1769_MODU...		☐
+ Local:2:I	{...}	{...}		AB:1769_MODU...		☐
+ Local:2:O	{...}	{...}		AB:1769_MODU...		☐
+ MNET	{...}	{...}		MNETModuleDef		☐
+ MVI69EGECClock	{...}	{...}		GECClockType	Allows read/write ...	☐
+ MVI69EGECEthernet	{...}	{...}		GECEthernetType	Allows read/write ...	☐
+ MVI69EMBTCPClock	{...}	{...}		MBTCPClockType	Allows read/write ...	☐
+ MVI69EMBTCPEthernet	{...}	{...}		MBTCPEthernetT...	Allows read/write ...	☐
+ ReadClockMSG	{...}	{...}		MESSAGE		☐
+ ReadEthernetMSG	{...}	{...}		MESSAGE		☐
+ WriteClockMSG	{...}	{...}		MESSAGE		☐
+ WriteEthernetMSG	{...}	{...}		MESSAGE		☐

2.3 Optional Add-On Instruction

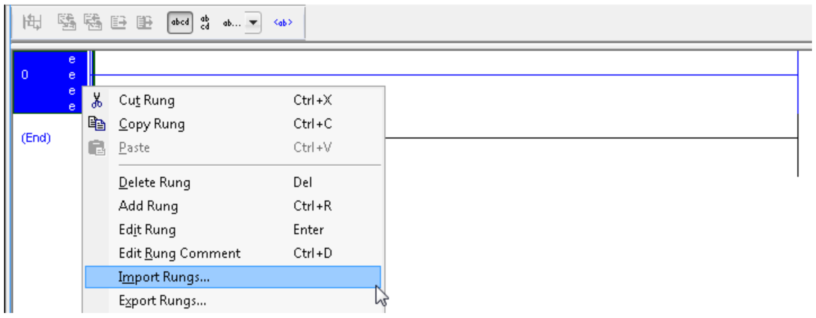
The Optional AOI supports the following optional features:

- Read/Write IP Address
- Read/Write Date Time

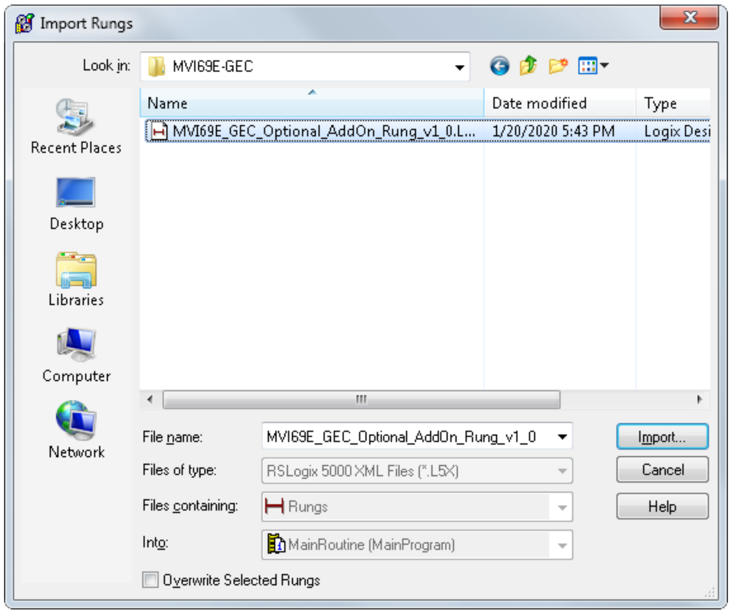
Using controller tags, the Optional AOI allows you to request and set the module's IP address, date, and time. These optional features are not supported by the MVI69-GEC legacy module.

Note: The Optional AOI may be added to an existing legacy MVI69-GEC application to add the new functionality during a module replacement.

1 Add a new rung to the existing processor ladder logic. Right-click on the new rung and select *Import Rungs…*



2 Select the Optional AOI file: *MVI69E_GEC_Optional_AddOn_Rung.L5X*

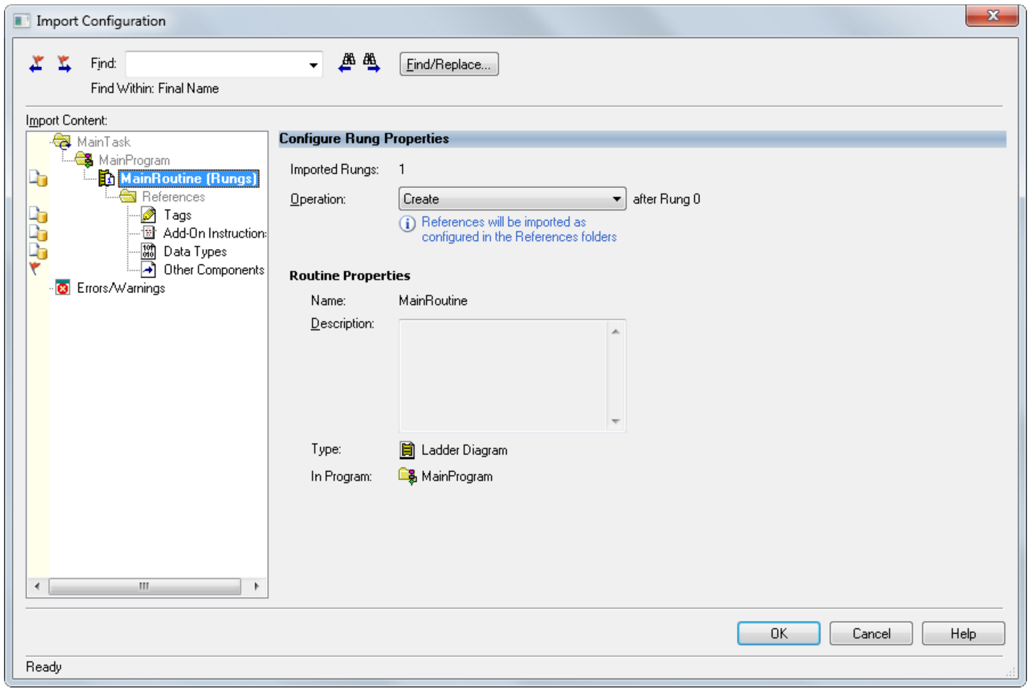

Import Rungs

Look in: MVI69E-GEC

Name	Date modified	Type
MVI69E_GEC_Optional_AddOn_Rung_v1_0.L...	1/20/2020 5:43 PM	Logix Desi

File name: MVI69E_GEC_Optional_AddOn_Rung_v1_0
Files of type: RSLogix 5000 XML Files (*.L5X)
Files containing: Rungs
Into: MainRoutine (MainProgram)
Overwrite Selected Rungs

3 At the *Import Configuration* window, select the *Operation* parameter to **CREATE**. Then click **OK**.


Import Configuration

Find:
Find Within: Final Name

Import Content:
MainTask
MainProgram
MainRoutine (Rungs)
References
Tags
Add-On Instruction
Data Types
Other Components
Errors/Warnings

Configure Rung Properties

Imported Rungs: 1

Operation: Create after Rung 0

References will be imported as configured in the References folders

Routine Properties

Name: MainRoutine

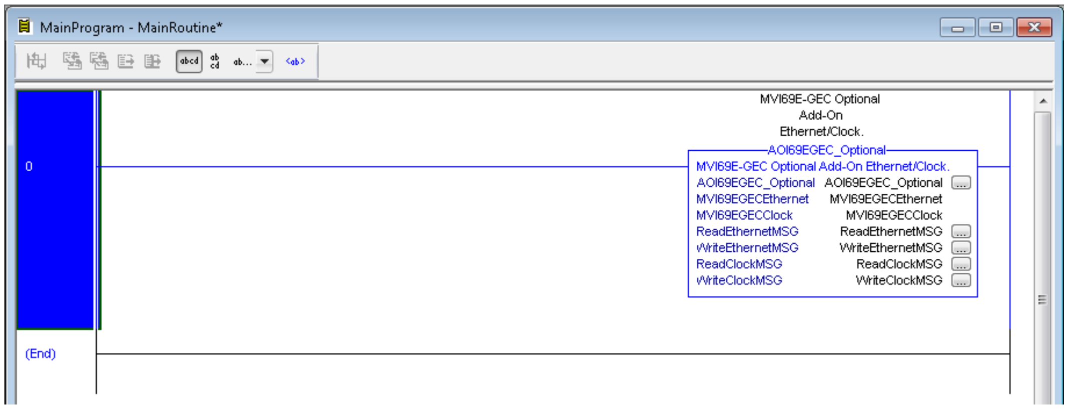
Description:

Type: Ladder Diagram

In Program: MainProgram

Ready

4 The imported AOI rung is now in place.


MainProgram - MainRoutine*

0

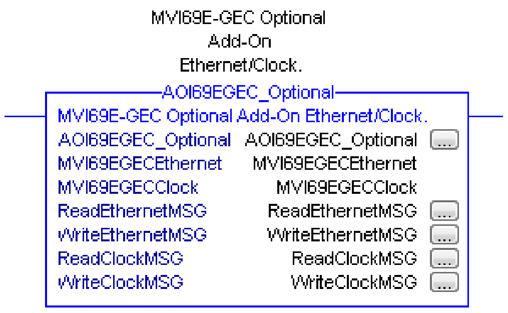
MVI69E-GEC Optional Add-On Ethernet/Clock.

AOI69EGEC_Optional
MVI69E-GEC Optional Add-On Ethernet/Clock.
AOI69EGEC_Optional AOI69EGEC_Optional
MVI69EGECEthernet MVI69EGECEthernet
MVI69EGECClock MVI69EGECClock
ReadEthernetMSG ReadEthernetMSG
WriteEthernetMSG WriteEthernetMSG
ReadClockMSG ReadClockMSG
WriteClockMSG WriteClockMSG

(End)

2.3.1 *Setting Up the Optional AOI*

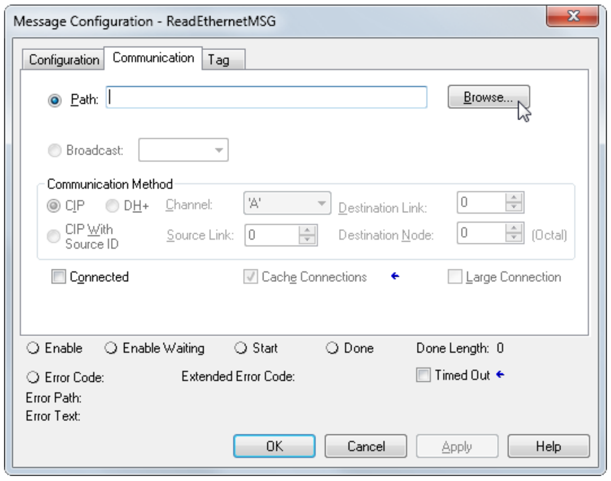
1 Click on the *ReadEthernetMSG* [...] icon to configure the message route:


MVI69E-GEC Optional
Add-On
Ethernet/Clock.

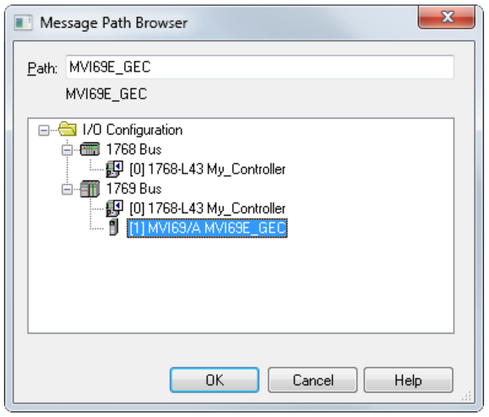
AOI69EGEC_Optional
MVI69E-GEC Optional Add-On Ethernet/Clock.

AOI69EGEC_Optional	AOI69EGEC_Optional
MVI69EGECEthernet	MVI69EGECEthernet
MVI69EGECClock	MVI69EGECClock
ReadEthernetMSG	ReadEthernetMSG
WriteEthernetMSG	WriteEthernetMSG
ReadClockMSG	ReadClockMSG
WriteClockMSG	WriteClockMSG

2 In the *Message Configuration* dialog, under the *Communication* tab, select the **BROWSE** button.


Message Configuration - ReadEthernetMSG

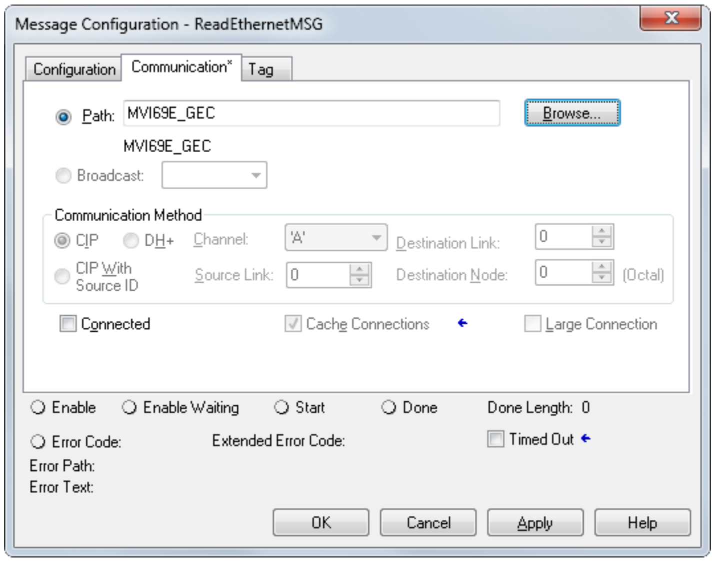
3 In the *Message Path Browser* dialog, select the MVI69E-GEC module under the *1769 Bus* and click at **OK**.


Message Path Browser

Path: MVI69E_GEC
MVI69E_GEC

```
I/O Configuration
  1768 Bus
    [0] 1768-L43 My_Controller
  1769 Bus
    [0] 1768-L43 My_Controller
    [1] MVI69/A MVI69E_GEC
```

4 The module name is displayed in the *Path* field. Click **OK** to confirm the route configuration.


Message Configuration - ReadEthernetMSG

Configuration | Communication* | Tag

Path: MVI69E_GEC
MVI69E_GEC
Broadcast:

Communication Method
CIP DH+ Channel: 'A' Destination Link: 0
CIP With Source ID Source Link: 0 Destination Node: 0 (Octal)

Connected Cache Connections Large Connection

Enable Enable Waiting Start Done Done Length: 0
Error Code: Extended Error Code: Timed Out
Error Path:
Error Text:

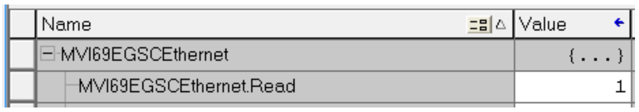
5 Repeat the same procedure to set the route for the remaining messages:

- *WriteEthernetMSG* ...
- *ReadClockMSG* ...
- *WriteClockMSG* ...

2.4 Synchronizing the IP Settings from the MVI69E-GEC to the Processor

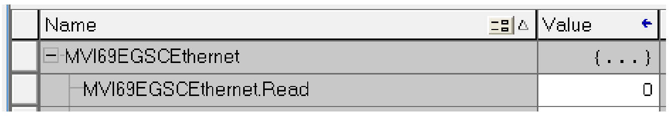
This section covers the process to read the IP settings from the MVI69E-GEC, and implement them in the processor.

1 To trigger the IP settings read operation, set the *MVI69EGECEthernet.Read* bit to ‘**1**’.



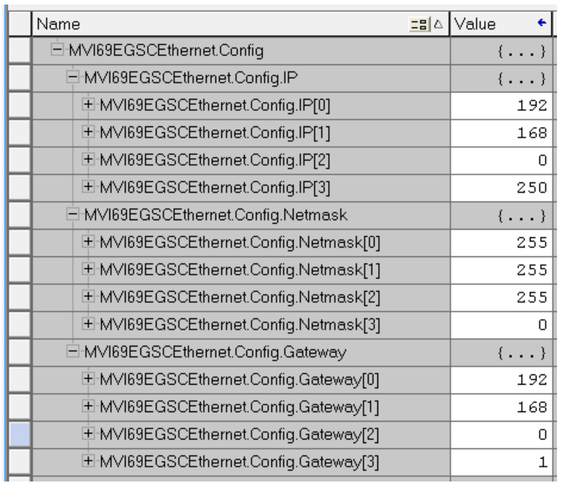
Name	Value
⊟ MVI69EGSCEthernet	{...}
MVI69EGSCEthernet.Read	1

2 Once the operation is concluded, the tag will automatically reset to ‘**0**’.



Name	Value
⊟ MVI69EGSCEthernet	{...}
MVI69EGSCEthernet.Read	0

3 The data is stored in the *MVI69EGECEthernet.Config* tags (IP, Netmask, Gateway) as follows:

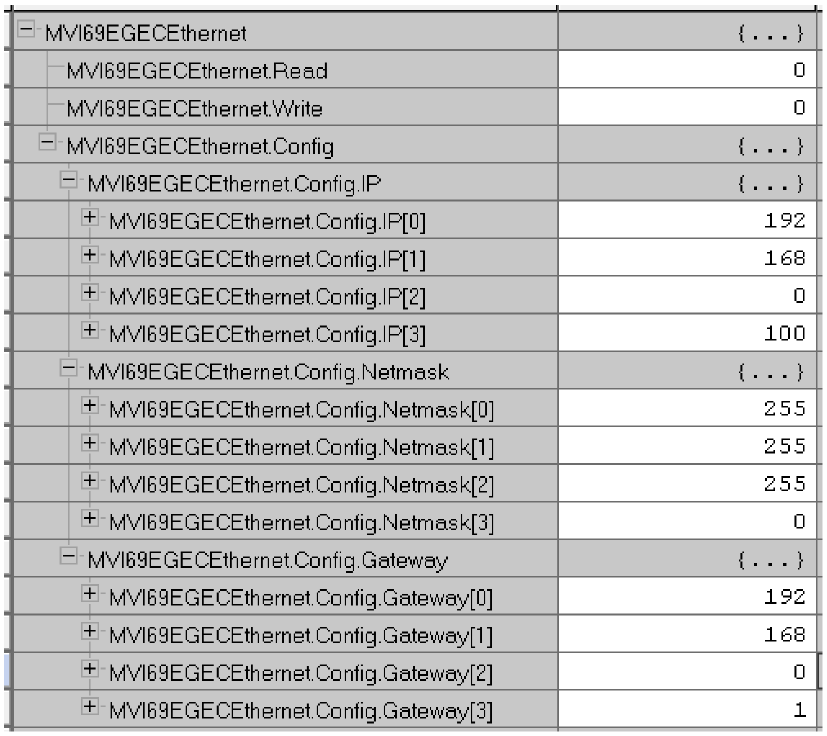


Name	Value
⊟ MVI69EGSCEthernet.Config	{...}
⊟ MVI69EGSCEthernet.Config.IP	{...}
⊞ MVI69EGSCEthernet.Config.IP[0]	192
⊞ MVI69EGSCEthernet.Config.IP[1]	168
⊞ MVI69EGSCEthernet.Config.IP[2]	0
⊞ MVI69EGSCEthernet.Config.IP[3]	250
⊟ MVI69EGSCEthernet.Config.Netmask	{...}
⊞ MVI69EGSCEthernet.Config.Netmask[0]	255
⊞ MVI69EGSCEthernet.Config.Netmask[1]	255
⊞ MVI69EGSCEthernet.Config.Netmask[2]	255
⊞ MVI69EGSCEthernet.Config.Netmask[3]	0
⊟ MVI69EGSCEthernet.Config.Gateway	{...}
⊞ MVI69EGSCEthernet.Config.Gateway[0]	192
⊞ MVI69EGSCEthernet.Config.Gateway[1]	168
⊞ MVI69EGSCEthernet.Config.Gateway[2]	0
⊞ MVI69EGSCEthernet.Config.Gateway[3]	1

2.5 Synchronizing the IP Settings from the Processor to the MVI69E-GEC

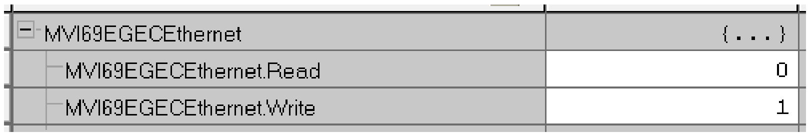
This section covers the process to send the IP settings from the processor to the MVI69E-GEC.

1 Populate the IP settings in the *MVI69EGECEthernet.Config* tag:



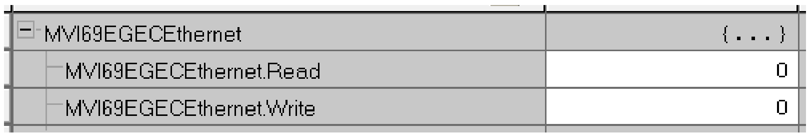
MVI69EGECEthernet	{...}
MVI69EGECEthernet.Read	0
MVI69EGECEthernet.Write	0
MVI69EGECEthernet.Config	{...}
MVI69EGECEthernet.Config.IP	{...}
MVI69EGECEthernet.Config.IP[0]	192
MVI69EGECEthernet.Config.IP[1]	168
MVI69EGECEthernet.Config.IP[2]	0
MVI69EGECEthernet.Config.IP[3]	100
MVI69EGECEthernet.Config.Netmask	{...}
MVI69EGECEthernet.Config.Netmask[0]	255
MVI69EGECEthernet.Config.Netmask[1]	255
MVI69EGECEthernet.Config.Netmask[2]	255
MVI69EGECEthernet.Config.Netmask[3]	0
MVI69EGECEthernet.Config.Gateway	{...}
MVI69EGECEthernet.Config.Gateway[0]	192
MVI69EGECEthernet.Config.Gateway[1]	168
MVI69EGECEthernet.Config.Gateway[2]	0
MVI69EGECEthernet.Config.Gateway[3]	1

2 Set the *MVI69EGECEthernet.Write* bit to ‘**1**’ to trigger the IP settings write operation.



MVI69EGECEthernet	{...}
MVI69EGECEthernet.Read	0
MVI69EGECEthernet.Write	1

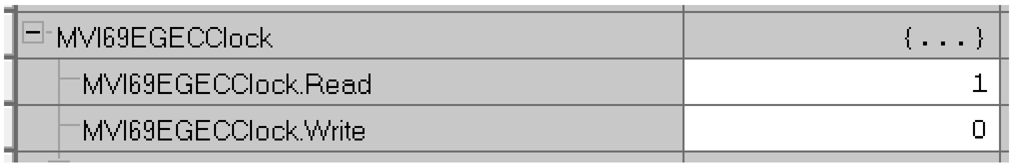
3 The *MVI69EGECEthernet.Write* bit will automatically reset to ‘**0**’ once the operation is concluded.



MVI69EGECEthernet	{...}
MVI69EGECEthernet.Read	0
MVI69EGECEthernet.Write	0

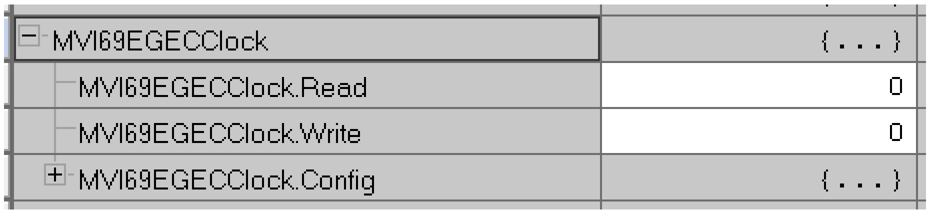
2.6 Reading the Date/Time from the MVI69E-GEC to the Processor

1 Toggle the *MVI69EGECClock.Read* bit to ‘**1**’ to toggle the date/time read operation.



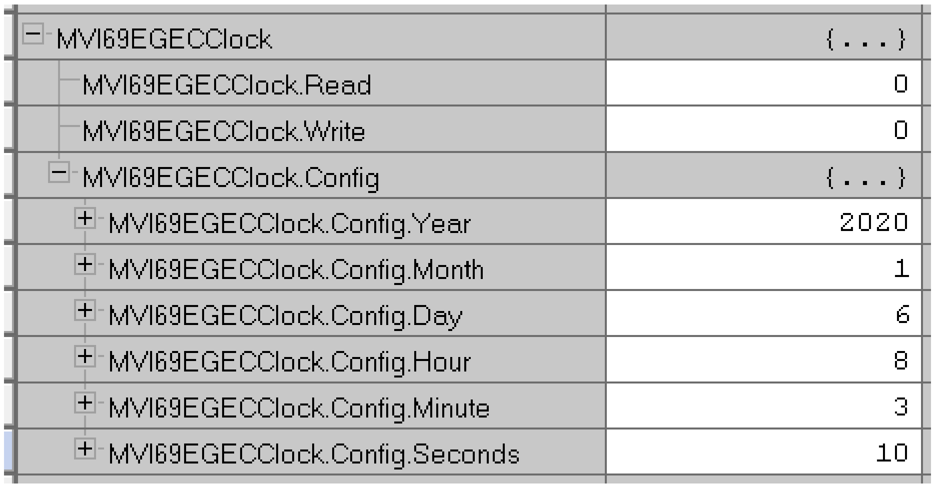
⊟ MVI69EGECClock	{...}
MVI69EGECClock.Read	1
MVI69EGECClock.Write	0

2 The *MVI69EGECClock.Read* bit will automatically reset to ‘**0**’ once the operation is concluded.



⊟ MVI69EGECClock	{...}
MVI69EGECClock.Read	0
MVI69EGECClock.Write	0
⊞ MVI69EGECClock.Config	{...}

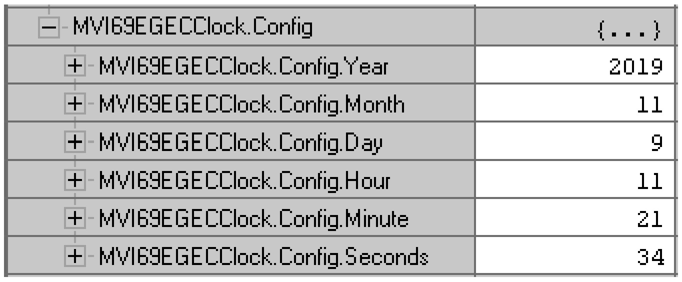
3 The date and time read from the MVI69E-GEC is stored at the *MVI69EGECClock.Config* tag.



⊟ MVI69EGECClock	{...}
MVI69EGECClock.Read	0
MVI69EGECClock.Write	0
⊟ MVI69EGECClock.Config	{...}
⊞ MVI69EGECClock.Config.Year	2020
⊞ MVI69EGECClock.Config.Month	1
⊞ MVI69EGECClock.Config.Day	6
⊞ MVI69EGECClock.Config.Hour	8
⊞ MVI69EGECClock.Config.Minute	3
⊞ MVI69EGECClock.Config.Seconds	10

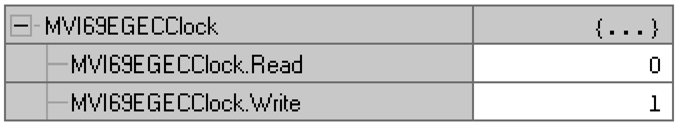
2.7 Writing the Date/Time from the Processor to the MVI69E-GEC

1 Populate date and time values in the *MVI69EGECClock.Config* tag.



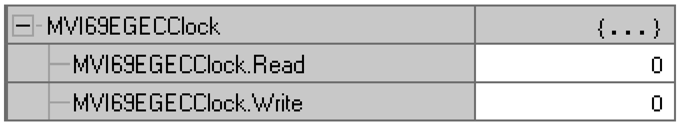
MVI69EGECClock.Config	{...}
MVI69EGECClock.Config.Year	2019
MVI69EGECClock.Config.Month	11
MVI69EGECClock.Config.Day	9
MVI69EGECClock.Config.Hour	11
MVI69EGECClock.Config.Minute	21
MVI69EGECClock.Config.Seconds	34

2 Toggle the *MVI69EGECClock.Write* bit to '**1**' to trigger the write date/time operation.



MVI69EGECClock	{...}
MVI69EGECClock.Read	0
MVI69EGECClock.Write	1

3 The *MVI69EGECClock.Write* tag will be automatically reset to '**0**' once the write date/time operation is concluded.



MVI69EGECClock	{...}
MVI69EGECClock.Read	0
MVI69EGECClock.Write	0

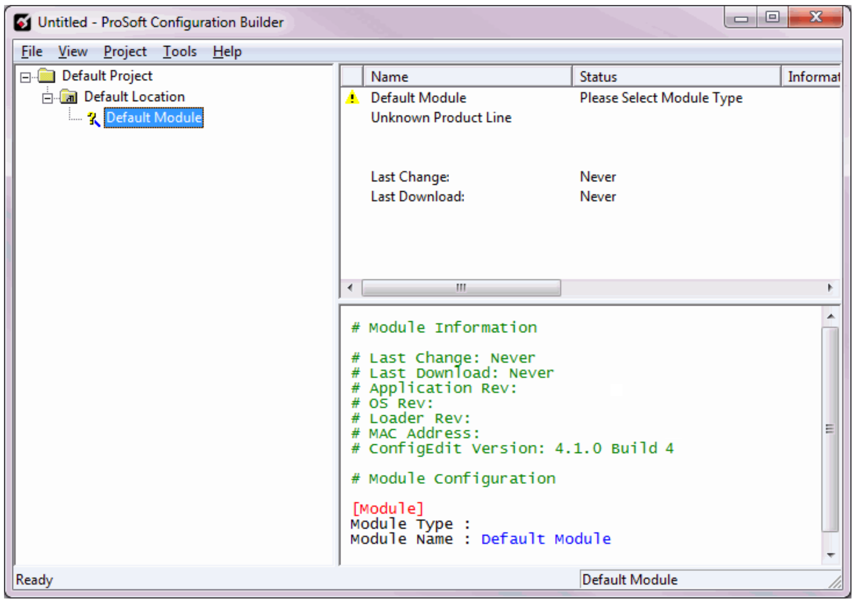
3 ProSoft Configuration Builder

3.1 Using ProSoft Configuration Builder

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

3.1.1 Setting Up the Project

1 Start *ProSoft Configuration Builder*. The following illustration shows the *ProSoft Configuration Builder* window with a new project.

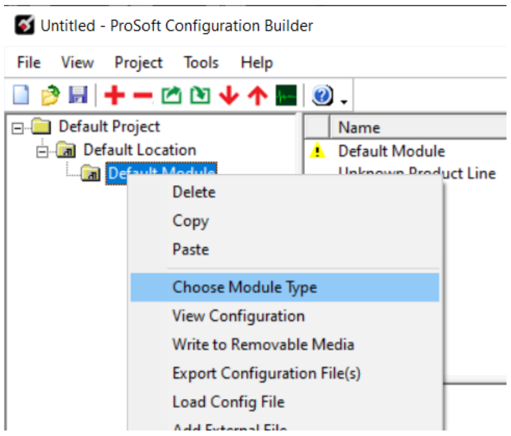


Default Module — Please Select Module Type
Unknown Product Line

Last Change: Never
Last Download: Never

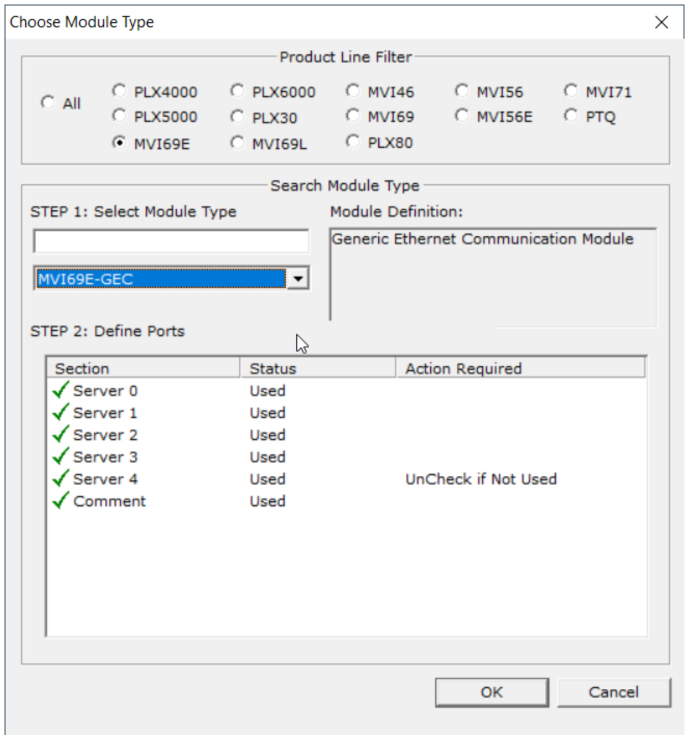
```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 4.1.0 Build 4

# Module Configuration

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

2 Use the mouse to select **DEFAULT MODULE** in the tree view, and then click the right mouse button to open a shortcut menu. Select **CHOOSE MODULE TYPE**.



3 This action opens the *Choose Module Type* dialog box.

4 In the *Product Line Filter* area of the dialog box, select **MVI69E**. In the *Select Module Type* dropdown list, select **MVI69E-GEC**, and then click **OK** to save your settings and return to the *ProSoft Configuration Builder* window.



Product Line Filter: [✓ MVI69E]

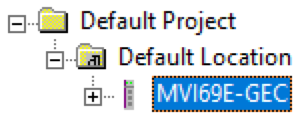
STEP 1: Select Module Type: MVI69E-GEC

Module Definition: Generic Ethernet Communication Module

Section	Status	Action Required
✓ Server 0	Used	
✓ Server 1	Used	
✓ Server 2	Used	
✓ Server 3	Used	
✓ Server 4	Used	UnCheck if Not Used
✓ Comment	Used	

3.1.2 Renaming PCB Objects

You can rename objects such as the *Default Project* and *Default Location* folders in the tree view. You can also rename the Module icon to customize the project.



1. Right-click the object you want to rename and choose **RENAME.**
2. Type the new name for the object and press **Enter**.

Configuring Module Parameters

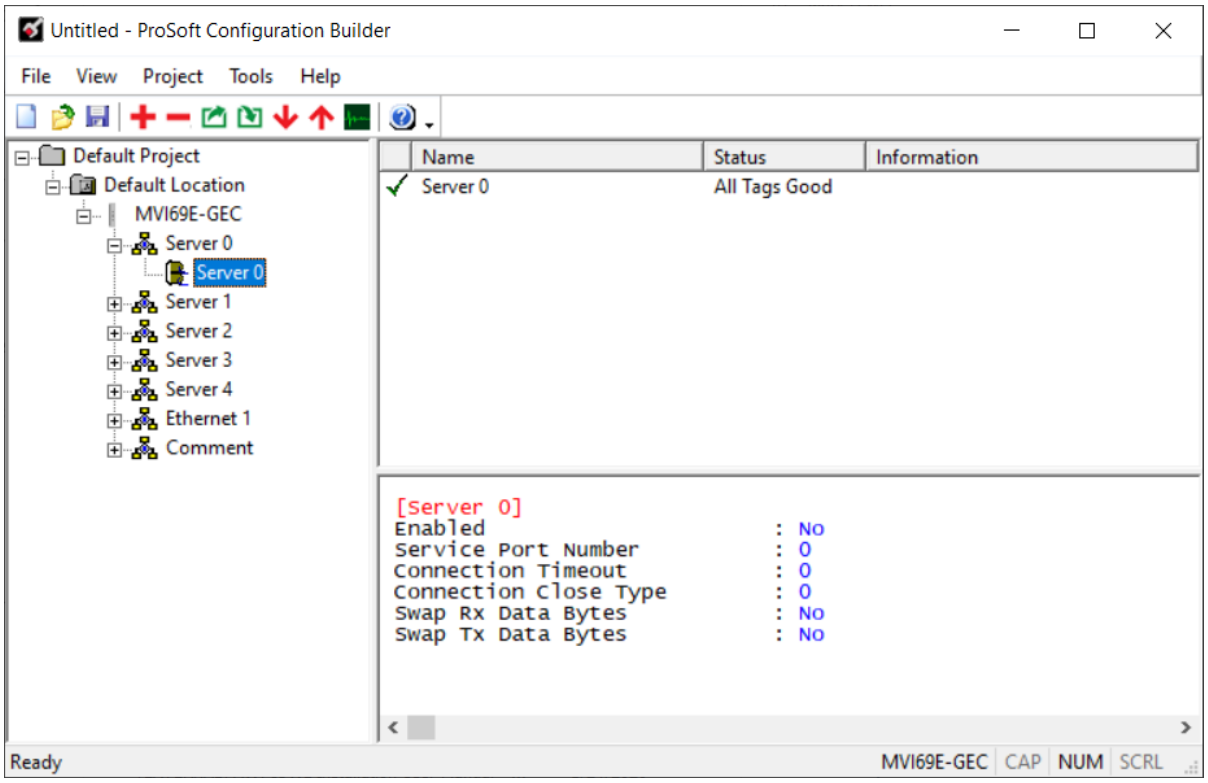
1. Click the **[+]** sign next to the module icon to expand module information.
2. Click the **[+]** sign next to any icon to view module information and configuration options.
3. Double-click any icon to open an *Edit* dialog box.
4. To edit a parameter, select the parameter in the left pane and make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

Printing a Configuration File

1. In the main PCB window, right-click the **MODULE** icon and choose **VIEW CONFIGURATION**.
2. In the *View Configuration* dialog box, open the **FILE** menu, and choose **PRINT**.
3. In the *Print* dialog box, choose the printer to use from the drop-down list, select the printing options, and then click **OK**.

3.2 Server Configuration

You can configure up to five servers ([Server 0] through [Server 4]). The configuration section for each server contains the same set of parameters. You can configure the parameters for each server to meet the requirements of your application.



Name	Status	Information
✓ Server 0	All Tags Good	

```
[Server 0]
Enabled                     : No
Service Port Number         : 0
Connection Timeout          : 0
Connection Close Type       : 0
Swap Rx Data Bytes          : No
Swap Tx Data Bytes          : No
```

3.2.1 Enabled

Yes or No

This parameter determines if the server will be utilized by the module. If a value of "Yes" is entered, the server will be used. Any other value will disable the server.

3.2.2 Service Port Number

1 to 65535

This parameter sets the TCP/IP service port for this server. Each server can have its own unique service port or can share the same number with other servers.

3.2.3 Connection Timeout

0 or 5000 to 65535

This parameter specifies the number of milliseconds the server will permit the server to be inactive after a connection is made before closing the socket. This timeout period is reset on each read or write packet. If the parameter is set to 0, the connection will not timeout.

3.2.4 Connection Close Type

0, 1 or 2

This coded parameter defines the personality of the server after a connection is made. If the parameter is set to 0, the socket will only be closed when a request from the client is received or the connection timeout is exceeded. If a value of 1 is selected, the server will close the socket after it transmits a single message. If a value of 2 is selected, the server will close the socket after it receives a message.

3.2.5 Swap Rx Data Bytes

Yes or No

This parameter determines if the data received by the server will have the byte order of the data swapped. If the parameter is set to No, no byte swapping will occur. If the parameter is set to Yes, the odd byte will be swapped with the even byte in each word of data received.

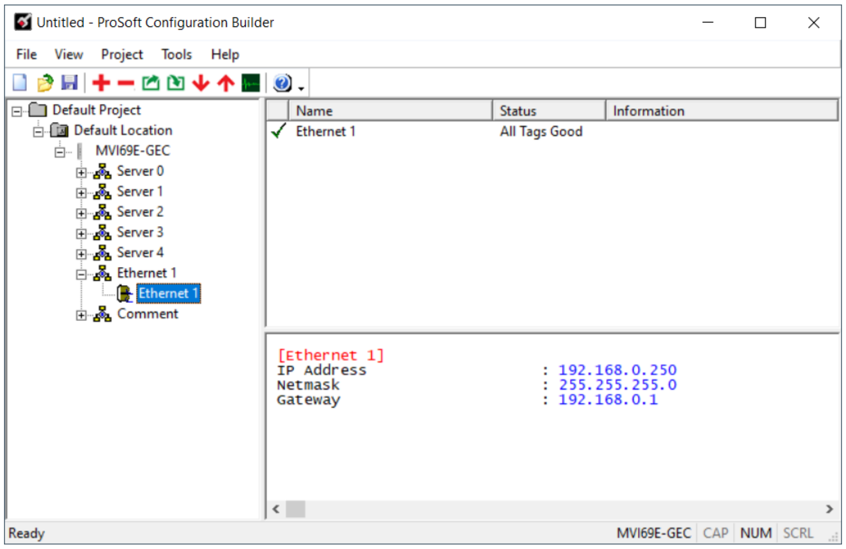
3.2.6 Swap Tx Data Bytes

Yes or No

This parameter determines if the data to be transmitted by the module will have the byte order of the data swapped. If the parameter is set to No, no byte swapping will occur. If the parameter is set to Yes, the odd byte will be swapped with the even byte in each word of data received.

3.3 Ethernet 1 (IP Address) Configuration

Use this procedure to configure the Ethernet settings for your module. You must assign an IP address, subnet mask and gateway address.



Name	Status	Information
✓ Ethernet 1	All Tags Good	

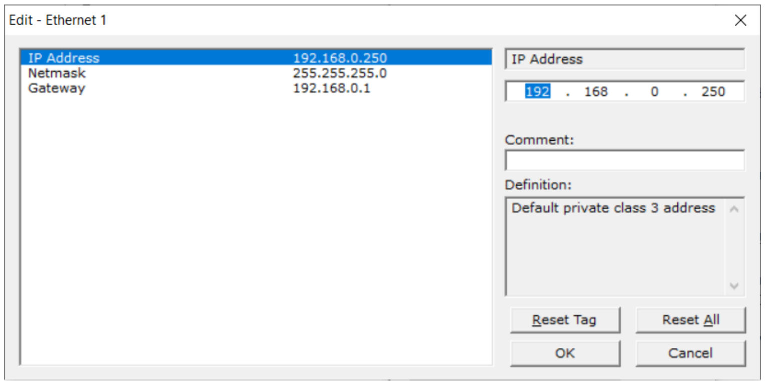
```
[Ethernet 1]
IP Address                    : 192.168.0.250
Netmask                       : 255.255.255.0
Gateway                       : 192.168.0.1
```

1 Determine the network settings for your module, with the help of your network administrator, if necessary. You will need the following information:

- 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 In PCB, double-click the **ETHERNET1** icon. This action opens the *Edit* dialog box.
3 Edit the values for *IP Address*, *Netmask* (subnet mask), and *Gateway* (default gateway).



IP Address	192.168.0.250
Netmask	255.255.255.0
Gateway	192.168.0.1

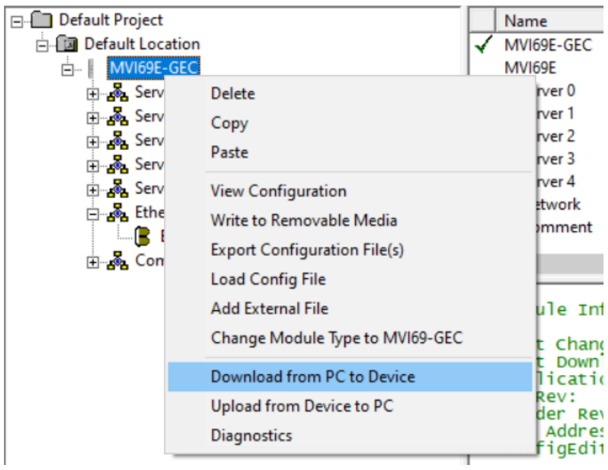
IP Address: 192 . 168 . 0 . 250

Definition: Default private class 3 address

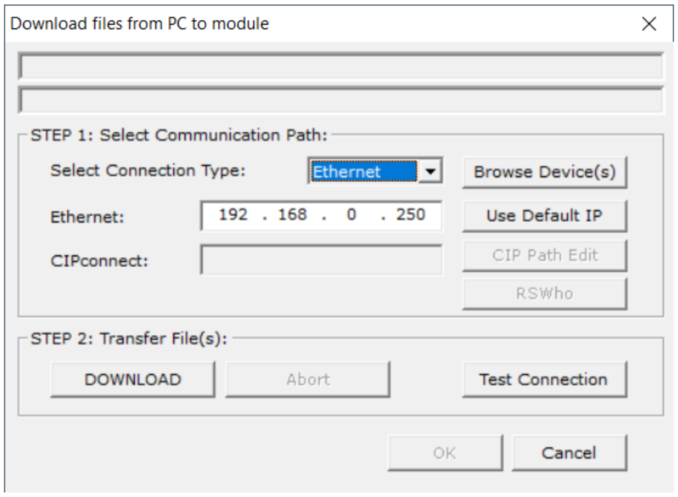
4 When you are finished editing, click **OK** to save your changes and return to the *ProSoft Configuration Builder* window.

3.4 Downloading the Configuration to the Module

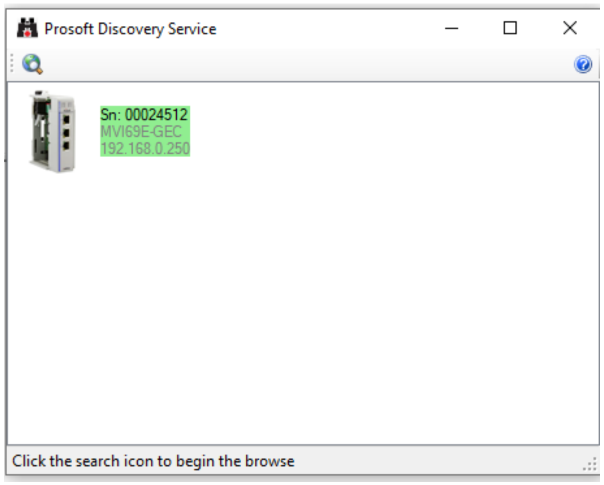
1 In the tree view in *ProSoft Configuration Builder*, right-click the module icon and select *Download from PC to Device*.



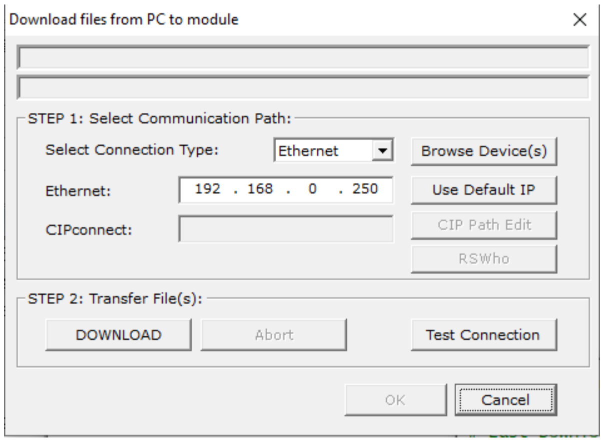
2 In the *Download files from PC to module* dialog, click **BROWSE DEVICE(S)** button.


Download files from PC to module
STEP 1: Select Communication Path:
Select Connection Type: Ethernet
Browse Device(s)
Ethernet: 192 . 168 . 0 . 250
Use Default IP
CIPconnect:
CIP Path Edit
RSWho
STEP 2: Transfer File(s):
DOWNLOAD
Abort
Test Connection
OK
Cancel

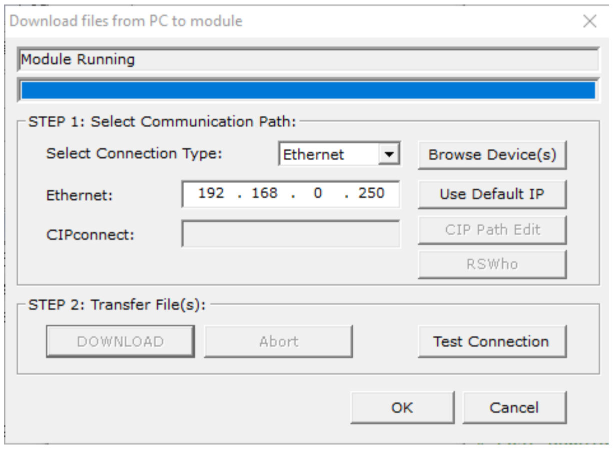
3 This launches the ProSoft Discovery Service utility to scan the network for ProSoft Technology devices. Once the module is located, double-click on the module icon.


Prosoft Discovery Service
Sn: 00024512
MVI69E-GEC
192.168.0.250
Click the search icon to begin the browse

4 In the *Transfer File(s)* section, click the **DOWNLOAD** button.

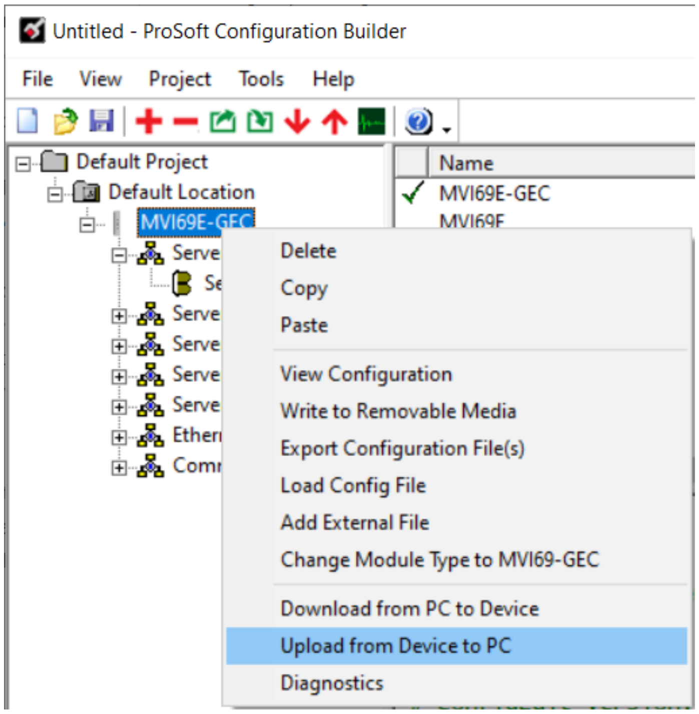

Download files from PC to module
STEP 1: Select Communication Path:
Select Connection Type: Ethernet
Browse Device(s)
Ethernet: 192 . 168 . 0 . 250
Use Default IP
CIPconnect:
CIP Path Edit
RSWho
STEP 2: Transfer File(s):
DOWNLOAD
Abort
Test Connection
OK
Cancel

5 When the download is complete, the **Module Running** message is displayed.

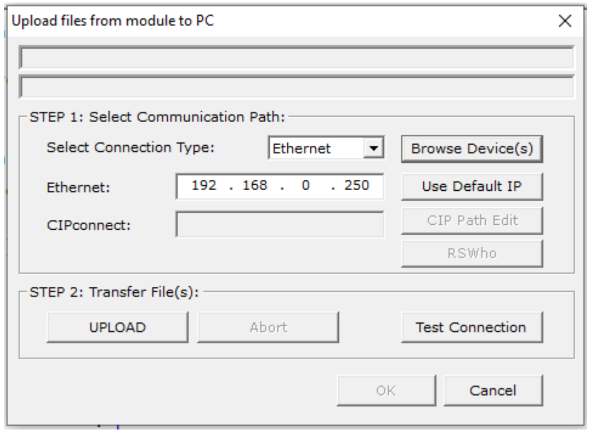

Download files from PC to module
Module Running
STEP 1: Select Communication Path:
Select Connection Type: Ethernet
Browse Device(s)
Ethernet: 192 . 168 . 0 . 250
Use Default IP
CIPconnect:
CIP Path Edit
RSWho
STEP 2: Transfer File(s):
DOWNLOAD
Abort
Test Connection
OK
Cancel

3.5 Uploading the Configuration File from the Module

1 In the ProSoft Configuration Builder tree view, right-click the **MVI69E-GEC** icon and choose **UPLOAD FROM DEVICE TO PC**.



2 Click the **BROWSE DEVICE(S)** button.


Upload files from module to PC

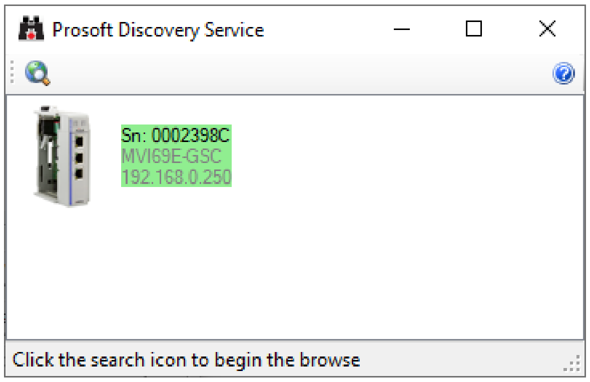
STEP 1: Select Communication Path:

Select Connection Type: Ethernet

Ethernet: 192 . 168 . 0 . 250

CIPconnect:

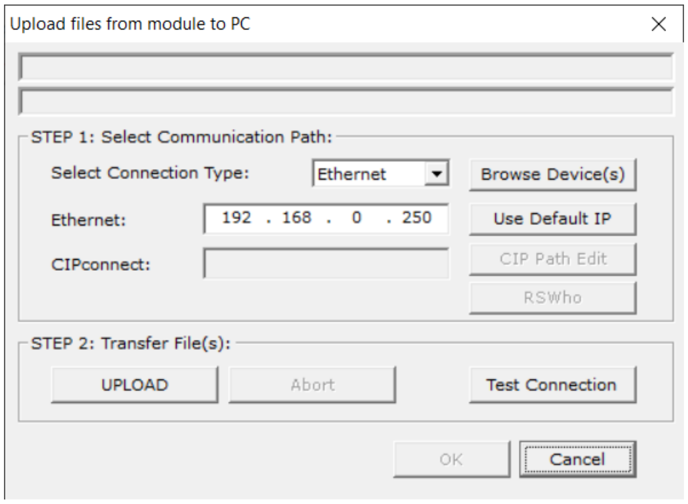
3 This launches the ProSoft Discovery Service utility to scan the network for ProSoft Technology devices. Once the module is located, double-click on the module icon.



Sn: 0002398C
MVI69E-GSC
192.168.0.250

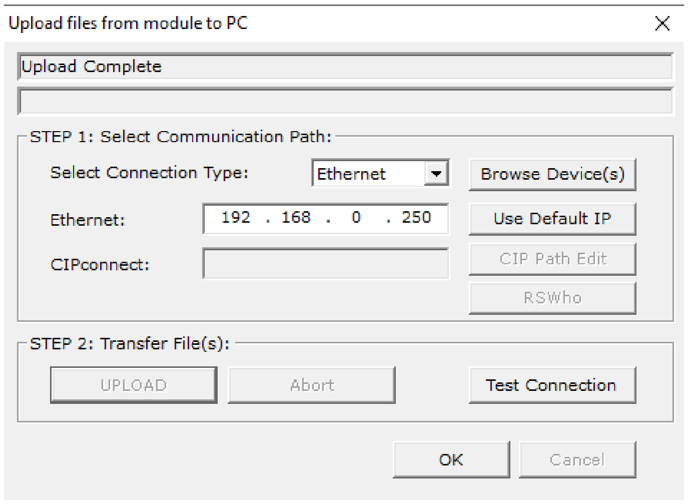
Click the search icon to begin the browse

4 In the *Transfer File(s)* section, click the **UPLOAD** button.


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

Select Connection Type: Ethernet
Ethernet: 192 . 168 . 0 . 250
CIPconnect:

5 Once the file is uploaded, the following message is displayed:


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

Upload Complete

Select Connection Type: Ethernet
Ethernet: 192 . 168 . 0 . 250
CIPconnect:

4 Using Controller Tags

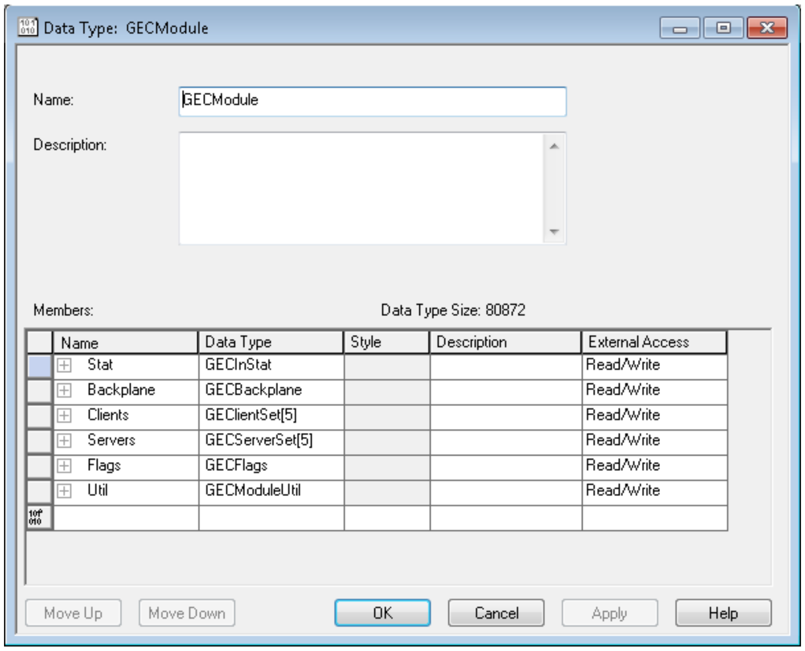
Ladder logic is required for the MVI69E-GEC module to operate. Tasks that must be handled by the ladder logic are module data transfer, special block handling, and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

The sample ladder logic, (download from www.prosoft-technology.com) is extensively commented, to provide information on the purpose and function of each rung. For most applications, the sample ladder will work without modification.

4.1 Module Data

All data related to the MVI69E-GEC is stored in a user defined data type.

The following table describes the structure of the object.



Name: GECModule

Description:

Members: Data Type Size: 80872

Name	Data Type	Style	Description	External Access
Stat	GECInStat			Read/Write
Backplane	GECBackplane			Read/Write
Clients	GECClientSet[5]			Read/Write
Servers	GECServerSet[5]			Read/Write
Flags	GECFlags			Read/Write
Util	GECModuleUtil			Read/Write

This object contains objects that define variables for the module and status data related to the module. Each of these object types is discussed in the following topics of the document.

4.1.1 GECInStat (Status Object)

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

Name	Data Type	Description
PassCnt	INT	Program cycle counter for module
Product	INT[2]	Product code for module (GEC)
Rev	INT[2]	Revision level of module's code
OP	INT[2]	Operating system version for module
Run	INT[2]	Run number for module
BlkErrs	GECBlkStat	Data block transfer statistics
Server	GECServerStat[5]	Status for each server
Client	GECClientStat[5]	Status for each client

Within the GECInStat objects are objects containing the status information for each server and the block transfer process. Refer to the Reference chapter for a complete listing of the data stored in this object.

4.1.2 GECServerStat (Server Status Object)

The GECServerStat object stores the status information related to each individual server on the module. All messages are counted for both the receive and transmit operations. Additionally, the object contains the CfgErrword member. This member is discussed in the following section. The following table describes the structure of the object.

Name	Data Type	Description
Enabled	INT	Flag to indicate if server is enabled (1=Yes,0=No)
State	INT	Current state of server
IP	INT[4]	IP address of host connected to server
Port	INT	TCP port for host connected to server
Open	INT	Number of times server performed an open
Est	INT	Number of times connection established
Close	INT	Number of times socket closed
Rx	INT	Number of messages received
RxOverflow	INT	Number of receive buffer overflows
Tx	INT	Number of messages transmitted
TxOverflow	INT	Number of transmit buffer overflows
Timeout	INT	Number of socket timeout conditions
CfgErrword	INT	Configuration error word value for server

4.1.3 GECBlkStat (Block Error Status Object)

The GECBlkStat object holds the status data related to the data transfer between the module and the controller. Each read and write block transferred between the module and the controller is counted in the Read and Write data members, respectively. Each write block that is parse by the module is counted in the Parse data member. The Err member is incremented each time a bad block is transferred between the two devices or there is an error in the backplane driver in the module. The following table describes the structure of the object.

Name	Data Type	Description
Read	INT	Number of blocks read by the module
Write	INT	Number of blocks written by the module
Parse	INT	Number of blocks parsed by the module
Err	INT	Number of block transfer errors

4.1.4 GECClientStat

This object stores the status information for a single client in the module. This data is received from the module in each new input image. The following table describes the structure of the object.

Name	Data Type	Description
Connected	INT	Connection state
State	INT	Socket state
IP	DINT	IP address of connected server
Port	INT	Service port of connected server
RxCount	INT	Number of receive messages
RxOverflow	INT	Number of times receive buffer overflowed
TxCount	INT	Number of transmit messages
TxOverflow	INT	Number of times the transmit buffer overflowed

The connected member of the object can have one of the values shown in the following table.

State Value	Definition
-3	Server closed connection for client or server is not available.
-2	Unable to open connection with specified server.
-1	Unable to open connection with specified server because of invalid IP address.
0	The client is idle and not connected.
1	The client set to connect to the server and waiting for the connection to establish.
2	The client is connected to the server and can transfer data.
3	The connection is being closed for the client.

A value less than one indicates that the client is not connected to a server and is available for use. If the client was previously used and an error condition existed relative to the socket, this parameter will be set to a value less than zero. If the client socket closed normally, the value will be set to 0.

When the ladder logic requests a new connection, it will set the parameter to a value of 1. The module will recognize this request and initiate the connection with the specified server. If the connection is established, the parameter will be set to two. Data may now be exchanged between the client and the server.

The parameter will be set to a value of 3 when the connection is being closed. This operation can be initiated from either the client using the Client control word in the output image or by the server.

The state member of the object can have one of the following values:

State Value	Definition
-1	Client is waiting for a connection request.
0	The client is waiting to establish the connection with the server.
1	The client has established a connection with the server and can send and receive data.
1000	The client has initiated a close operation on the connection.
1001	The client is waiting for the close on the connection to complete.
1002	The client is issuing an abort (reset) on the connection. The socket is forced closed.
3000	The client is issuing the ARP command request and waiting for the response.
3001	The client has received the ARP response and has opened the socket.

This member reports the current state of the client socket state machine in the module. It is preferred to use the Connected member of the object in the ladder logic instead of this member for control.

The next two members of the object are set by the ladder logic and correspond to the IP address of the server connected to the client and the service port in the server used for the connection.

The last four members of the object are statistics representing the transmit and receive activity of the client socket.

4.1.5 GECBackplane (Backplane Object)

The GECBackplane object stores all the variables required for the data transfer operation between the module and the controller. The LastRead data member is used as the handshaking byte to indicate the arrival of new data from the module. The following table describes the structure of the object.

Name	Data Type	Description
LastRead	INT	Sequence number of last block read
LastWriteCount	SINT[2]	Last number of bytes written
CurBlock	INT	Sequence number for current block
SourceIndex	INT	Current server or client in read block
RxLen	INT	Length of message received
TxServer	INT	Server number for current transmit
TxCount	INT	Number of bytes processed from last tx message

The other members of the object can be utilized in the ladder logic to assist in the data transfer operation.

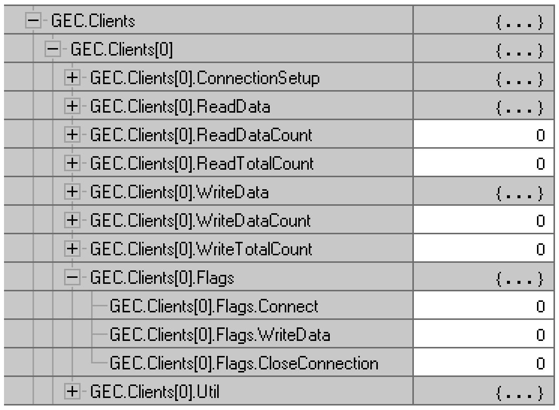
5 Sending and Receiving ASCII Data

5.1 Sending ASCII Data

The MVI69E-GEC, whether used as a client or server, can send ASCII texts to a remote device.

5.1.1 Sending ASCII Text as a Client

Use the following steps to configure the MVI69E-GEC as a client to send an ASCII string to a remote device (server). The MVI69E-GEC can simultaneously connect and send data to up to five servers.

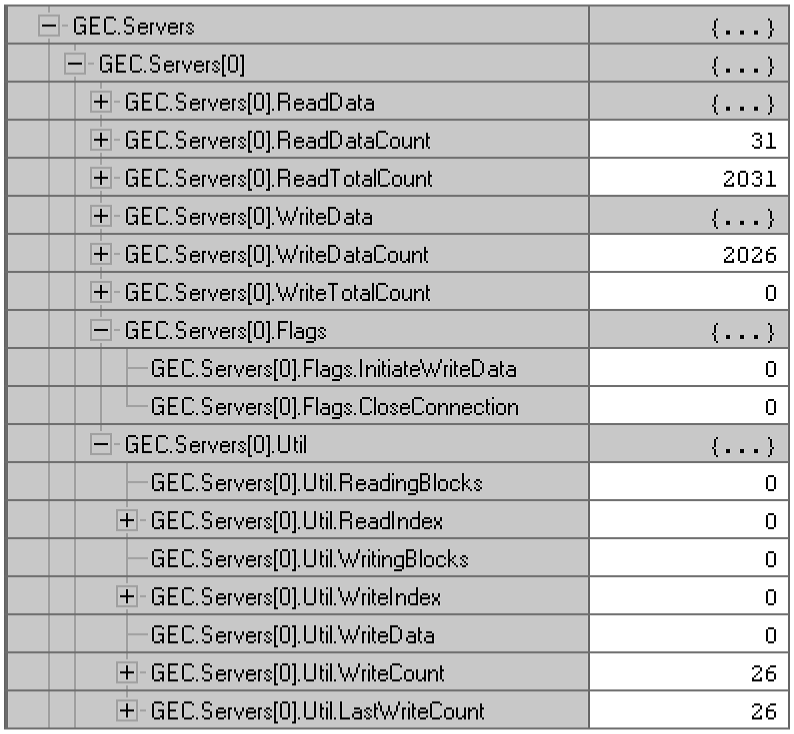


⊟ GEC.Clients	{...}
⊟ GEC.Clients[0]	{...}
⊞ GEC.Clients[0].ConnectionSetup	{...}
⊞ GEC.Clients[0].ReadData	{...}
⊞ GEC.Clients[0].ReadDataCount	0
⊞ GEC.Clients[0].ReadTotalCount	0
⊞ GEC.Clients[0].WriteData	{...}
⊞ GEC.Clients[0].WriteDataCount	0
⊞ GEC.Clients[0].WriteTotalCount	0
⊟ GEC.Clients[0].Flags	{...}
GEC.Clients[0].Flags.Connect	0
GEC.Clients[0].Flags.WriteData	0
GEC.Clients[0].Flags.CloseConnection	0
⊞ GEC.Clients[0].Util	{...}

1. Enter the IP address of the server at **GEC.Clients[x].ConnectionSetup.ServerIP**.
2. Enter the service port of the server at **GEC.Clients[x].ConnectionSetup.ServicePort**.
3. Enter the data to be sent at **GEC.Clients[x].WriteData**.
4. Enter the number of characters to be sent at **GEC.Clients[x].WriteDataCount**
5. Set the bit at **GEC.Clients[x].Flags.Connect** to open the connection to the server.
6. Set the bit at **GEC.Clients[x].Flags.WriteData** to send the message.
7. (Optional) Set the bit at **GEC.Clients[x].Flags.CloseConnection** to close the connection.
8. **GEC.Stat.Client[x].TxCount** increments by 1 on every transmission.

5.1.2 Sending ASCII Text as a Server

The remote Client must initiate a socket connection before the Server can transmit data to the Client.



GEC.Servers	{...}
GEC.Servers[0]	{...}
GEC.Servers[0].ReadData	{...}
GEC.Servers[0].ReadDataCount	31
GEC.Servers[0].ReadTotalCount	2031
GEC.Servers[0].WriteData	{...}
GEC.Servers[0].WriteDataCount	2026
GEC.Servers[0].WriteTotalCount	0
GEC.Servers[0].Flags	{...}
GEC.Servers[0].Flags.InitiateWriteData	0
GEC.Servers[0].Flags.CloseConnection	0
GEC.Servers[0].Util	{...}
GEC.Servers[0].Util.ReadingBlocks	0
GEC.Servers[0].Util.ReadIndex	0
GEC.Servers[0].Util.WritingBlocks	0
GEC.Servers[0].Util.WriteIndex	0
GEC.Servers[0].Util.WriteData	0
GEC.Servers[0].Util.WriteCount	26
GEC.Servers[0].Util.LastWriteCount	26

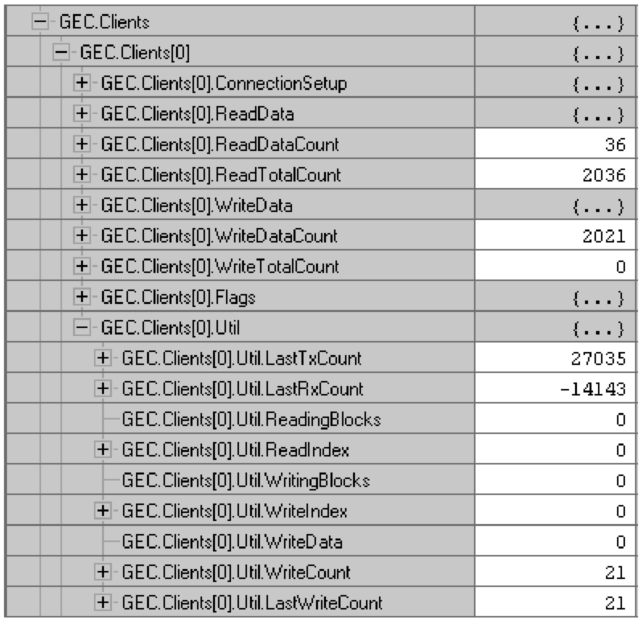
1. Enter the data to be sent at **GEC.SERVERS[X].WRITEDATA**.
2. Enter the number of characters to be sent at **GEC.SERVERS[X].WRITEDATACOUNT**.
3. Set the bit at **GEC.SERVERS[X].FLAGS.WRITEDATA** to send the message.
4. (Optional) Set the bit at **GEC.SERVERS[X].FLAGS.CLOSECONNECTION** to close the connection.
5. **GEC. SERVERS[X].UTIL.WRITECOUNT** increments by 1 on every transmission.

5.2 Receiving ASCII Data

The MVI69E-GEC, whether used as a client or server, can receive incoming ASCII texts from a remote device.

5.2.1 Receiving ASCII Text as a Client

The MVI69E-GEC can receive ASCII strings from the same server it sends to. Since the client socket connection has already been established with the server, the incoming data will be stored in the **GEC.CLIENTS[X].READDATA** array.



GEC.Clients	{...}
GEC.Clients[0]	{...}
GEC.Clients[0].ConnectionSetup	{...}
GEC.Clients[0].ReadData	{...}
GEC.Clients[0].ReadDataCount	36
GEC.Clients[0].ReadTotalCount	2036
GEC.Clients[0].WriteData	{...}
GEC.Clients[0].WriteDataCount	2021
GEC.Clients[0].WriteTotalCount	0
GEC.Clients[0].Flags	{...}
GEC.Clients[0].Util	{...}
GEC.Clients[0].Util.LastTxCount	27035
GEC.Clients[0].Util.LastRxCount	-14143
GEC.Clients[0].Util.ReadingBlocks	0
GEC.Clients[0].Util.ReadIndex	0
GEC.Clients[0].Util.WritingBlocks	0
GEC.Clients[0].Util.WriteIndex	0
GEC.Clients[0].Util.WriteData	0
GEC.Clients[0].Util.WriteCount	21
GEC.Clients[0].Util.LastWriteCount	21

1 When the MVI69E-GEC receives an ASCII string from a server, the **GEC.STAT.CLIENT[X].RXCOUNT** controller tag increments by 1. You will need to monitor this tag to determine a new message was received.

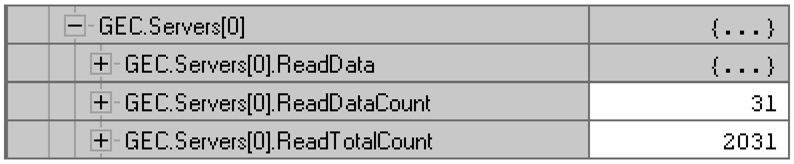
2 The **GEC.CLIENT[X].READDATA** array contains the ASCII text of the new message. This array is overwritten every time a new string is received. You will need to create logic that monitors when a new message is received (**GEC.STAT.CLIENT[X].RXCOUNT** increases by 1), and copies the text out of the **GEC.CLIENTS[X].READDATA ARRAY**.

3 The number of characters received in the new message is located at **GEC.CLIENTS[0].READDATACOUNT**.

4 The accumulated total number of characters received is located at **GEC.CLIENTS[0].READTOTALCOUNT**.

5.2.2 Receiving ASCII Text as a Server

When a server port of the MVI69E-GEC is set up, it will accept incoming ASCII text from a client only.



⊟ GEC.Servers[0]	{...}
⊞ GEC.Servers[0].ReadData	{...}
⊞ GEC.Servers[0].ReadDataCount	31
⊞ GEC.Servers[0].ReadTotalCount	2031

1 When the MVI69E-GEC receives an ASCII string from a client, the **GEC.STAT.SERVER[X].RX** controller tag increments by 1. You will need to monitor this tag to determine a new message was received.
2 The **GEC.SERVERS[X].READDATA** array contains the ASCII text of the new message. This array is overwritten every time a new string is received. You will need to create logic that monitors when a new message is received (**GEC.STAT.SERVER[X].RX** increases by 1), and copies the text out of the **GEC.SERVERS[X].READDATA** array.
3 The number of characters received in the new message is located at **GEC.SERVERS[0].READDATACOUNT**.
4 The accumulated total number of characters received is located at **GEC.SERVERS[0].READTOTALCOUNT**.

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

6.1 LED Status Indicators

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

LED	Color	Indication
ETH	Green	Application is running and Ethernet is ready
	Off	Application is not running
CLT	Red	Exception response received from the server; bad address, command, etc.
SRV	Red	Exception message received from the client
CFG	Red	Error in configuration
	Green	Configuration is OK
	Amber	Configuration state
	Off	Application is not running or backplane has failed
BP	Red	Processor is not in RUN mode
	Green	(Flashing) BP transfer is operational
	Amber	Initialization state
	Off	Application is not running
OK	Red	Application is not running
	Green	Application is running

6.1.1 Ethernet (ETH 1) LED Indicators

LED	State	Description
100 Mbit	Off	Ethernet connected at 10 Mbps duplex speed
	Amber (solid)	Ethernet connected at 100 Mbps duplex speed
LINK/ACT	Off	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	Green (solid or blinking)	Physical network connection detected. This LED must be On solid for Ethernet communication to be possible.

6.1.2 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 CompactLogix or MicroLogix 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.

6.1.3 Troubleshooting

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

Processor Errors

Problem Description	Steps to take
Processor Fault	Verify that the module is plugged into the slot that has been configured for the module. Verify that the slot in the rack configuration has been set up correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. Verify that all modules in the rack are configured in the ladder logic. The module has a power supply distance rating of 4 on CompactLogix, meaning that there must not be more than one other module between the MVI69E-GEC module and the power supply. If the module is used in a MicroLogix system, verify that the backplane can supply the 800 mA required by the module.

Module Errors

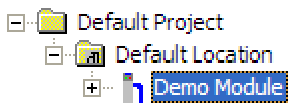
Problem Description	Steps to take
BP LED remains Off or blinks slowly	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in RUN mode ▪ The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

6.2 Using ProSoft Configuration Builder (PCB) for Diagnostics

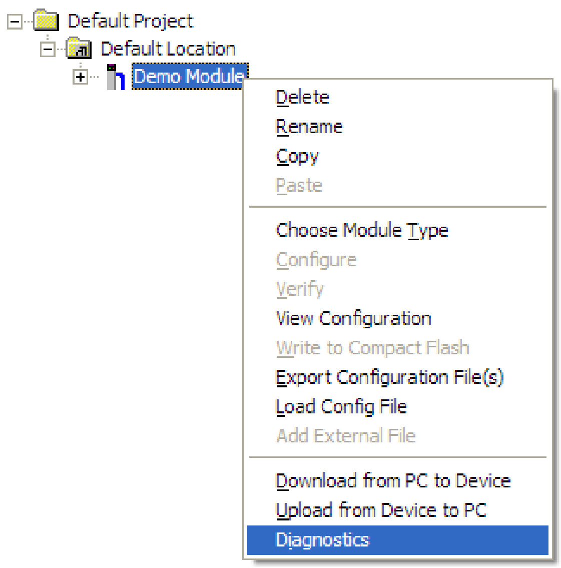
6.2.1 Using the Diagnostic Window in ProSoft Configuration Builder

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

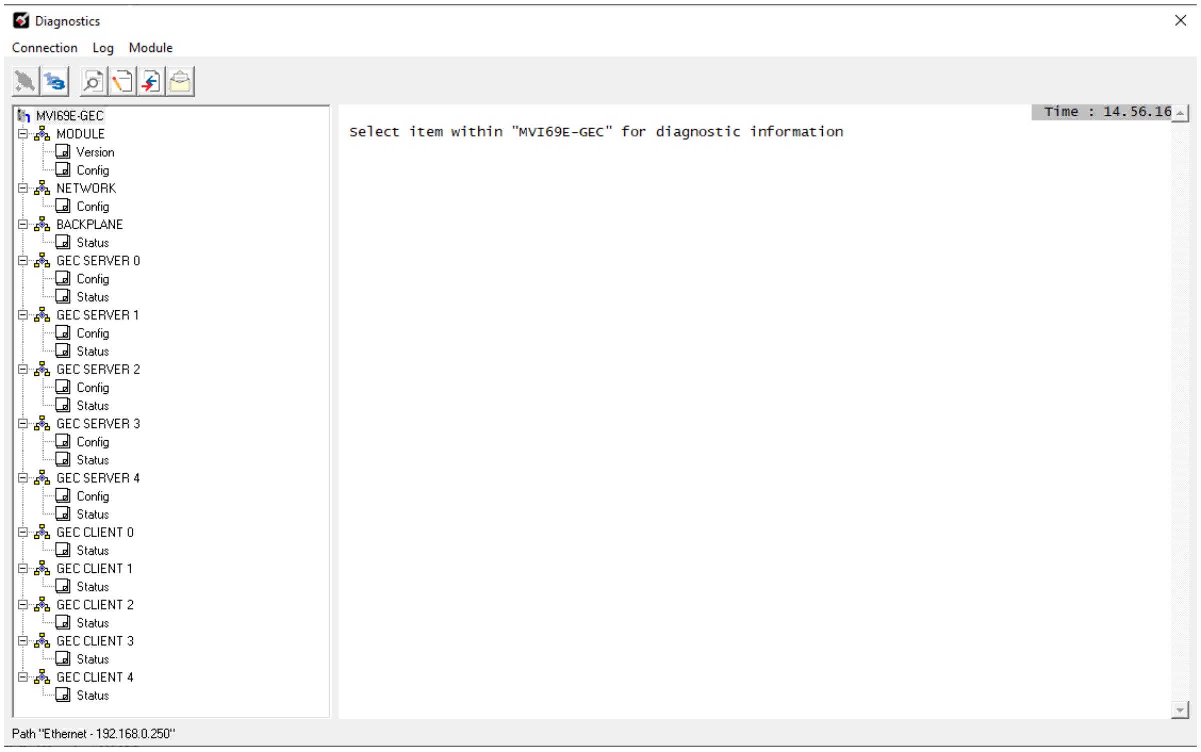
1 Start *PCB*, and then right-click the module icon.



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



This action opens the *Diagnostics* dialog box.



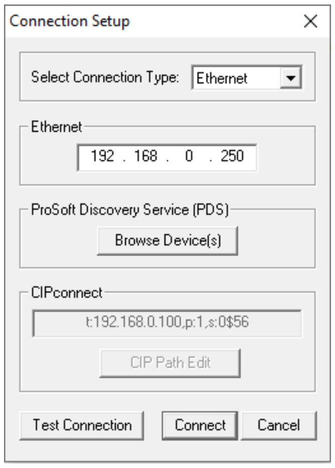
```
Time : 14.56.16
Select item within "MVI69E-GEC" for diagnostic information
```

Path "Ethernet - 192.168.0.250"

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

If there is no response from the module, try the following:

1 Click the **SETUP CONNECTION** button to configure the connection. On the *Connection Setup* dialog box, select a valid com port or other connection type supported by the module.



Select Connection Type: Ethernet

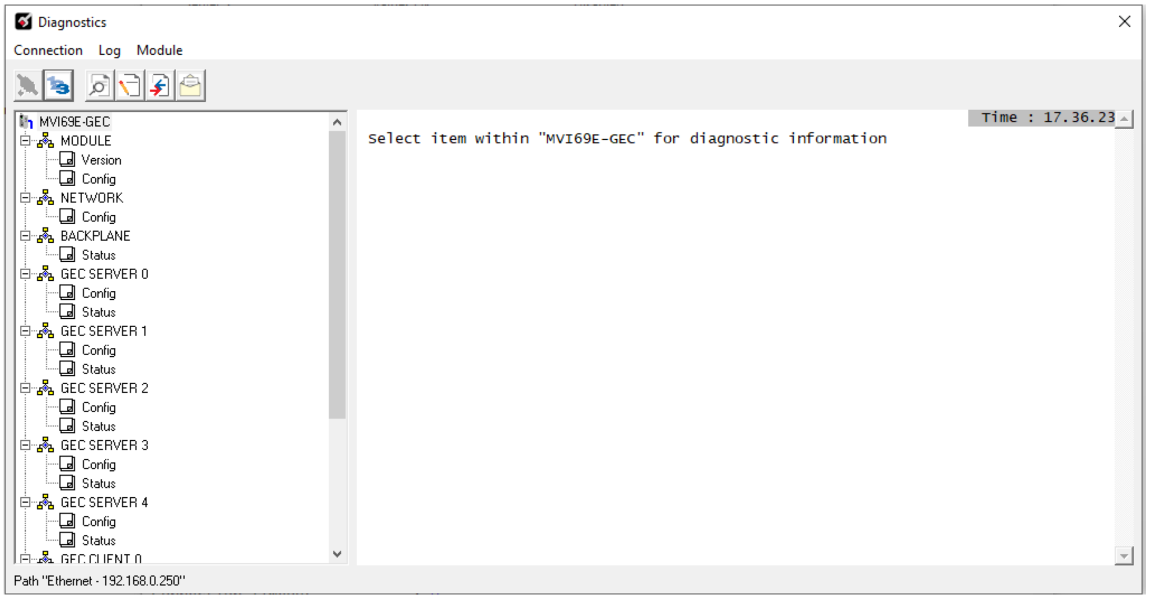
Ethernet: 192 . 168 . 0 . 250

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

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

6.2.2 Main Menu

When you first connect to the module from your computer, your terminal screen will be blank, you may refer to the menu items at the left showing the available diagnostics sections:

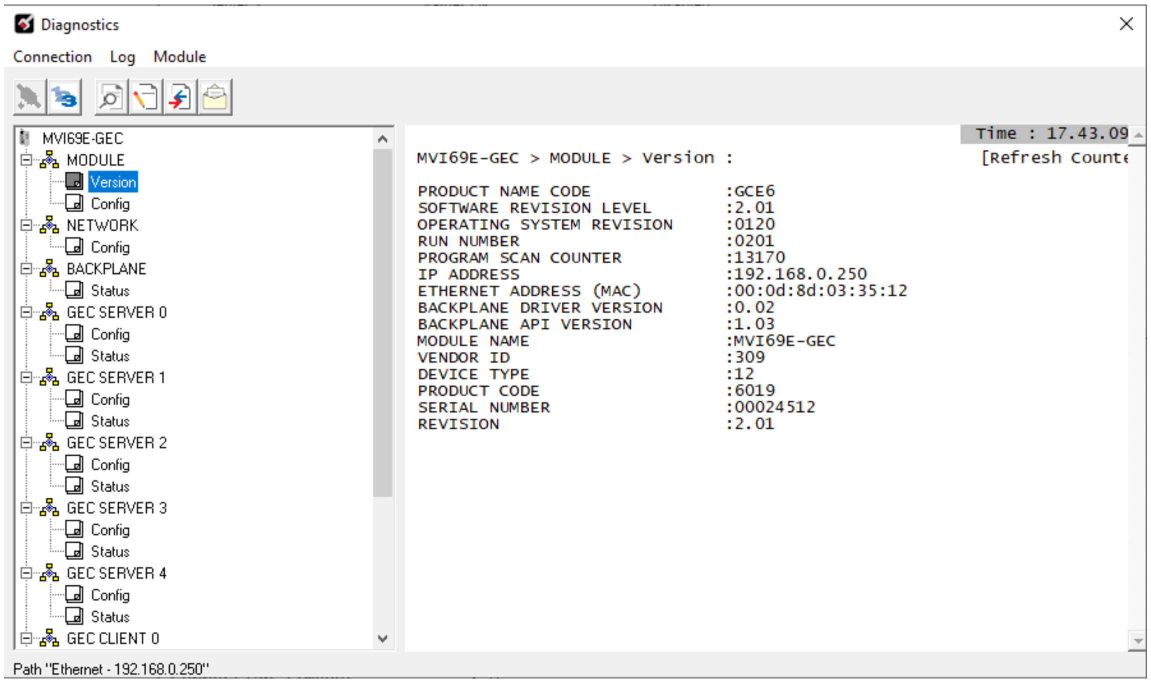


```
Time : 17.36.23
Select item within "MVI69E-GEC" for diagnostic information
```

Path "Ethernet - 192.168.0.250"

Viewing Module Version

Click at Version to display the module version, along with other relevant module information such as IP address, MAC address, serial number.

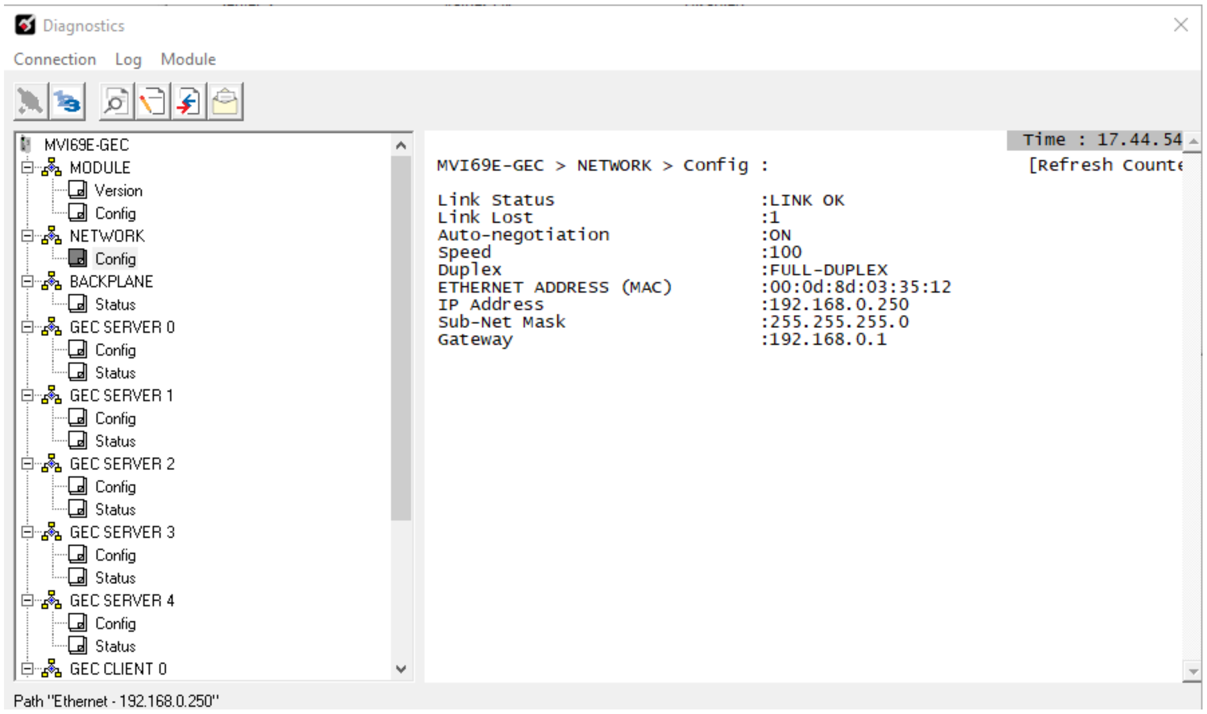


```
                                                  Time : 17.43.09
MVI69E-GEC > MODULE > Version :                  [Refresh Counte

PRODUCT NAME CODE              :GCE6
SOFTWARE REVISION LEVEL        :2.01
OPERATING SYSTEM REVISION      :0120
RUN NUMBER                     :0201
PROGRAM SCAN COUNTER           :13170
IP ADDRESS                     :192.168.0.250
ETHERNET ADDRESS (MAC)         :00:0d:8d:03:35:12
BACKPLANE DRIVER VERSION       :0.02
BACKPLANE API VERSION          :1.03
MODULE NAME                    :MVI69E-GEC
VENDOR ID                      :309
DEVICE TYPE                    :12
PRODUCT CODE                   :6019
SERIAL NUMBER                  :00024512
REVISION                       :2.01
```

Path "Ethernet - 192.168.0.250"

Viewing Ethernet Settings

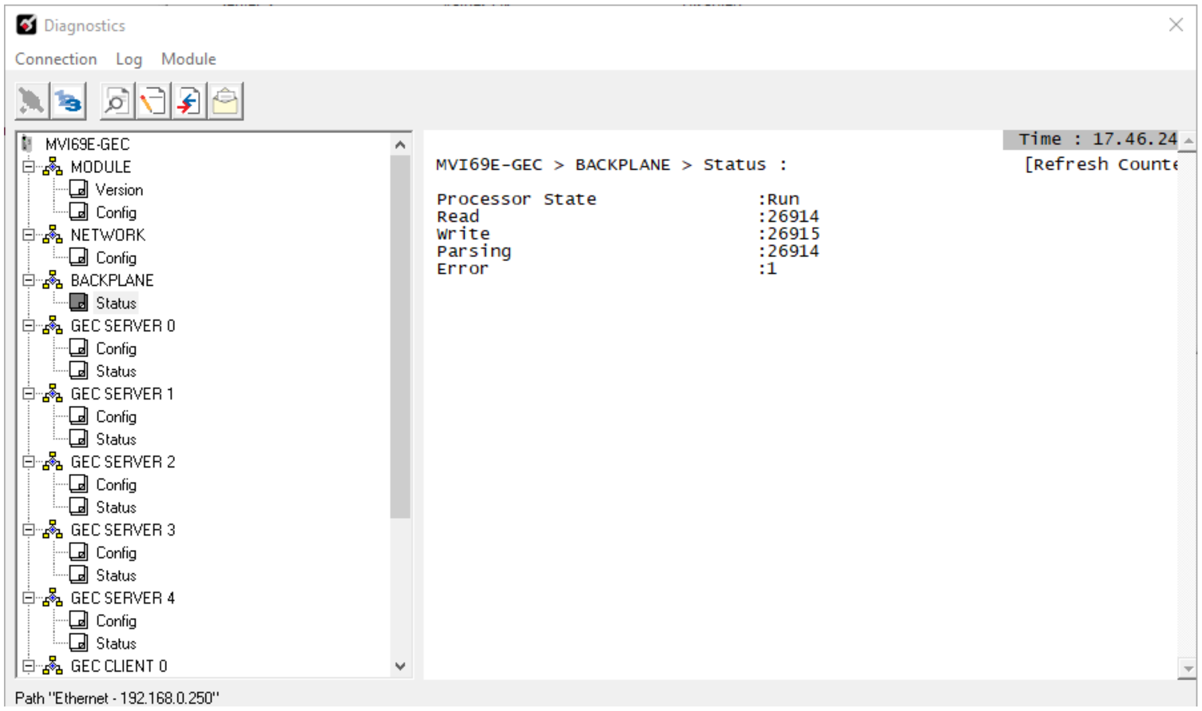
Click at *NETWORK* > *Config* to display the Ethernet configuration settings:



```
Time : 17.44.54
MVI69E-GEC > NETWORK > Config :                              [Refresh Counte

Link Status                     :LINK OK
Link Lost                       :1
Auto-negotiation                :ON
Speed                           :100
Duplex                          :FULL-DUPLEX
ETHERNET ADDRESS (MAC)          :00:0d:8d:03:35:12
IP Address                      :192.168.0.250
Sub-Net Mask                    :255.255.255.0
Gateway                         :192.168.0.1
```

Viewing the Backplane Status

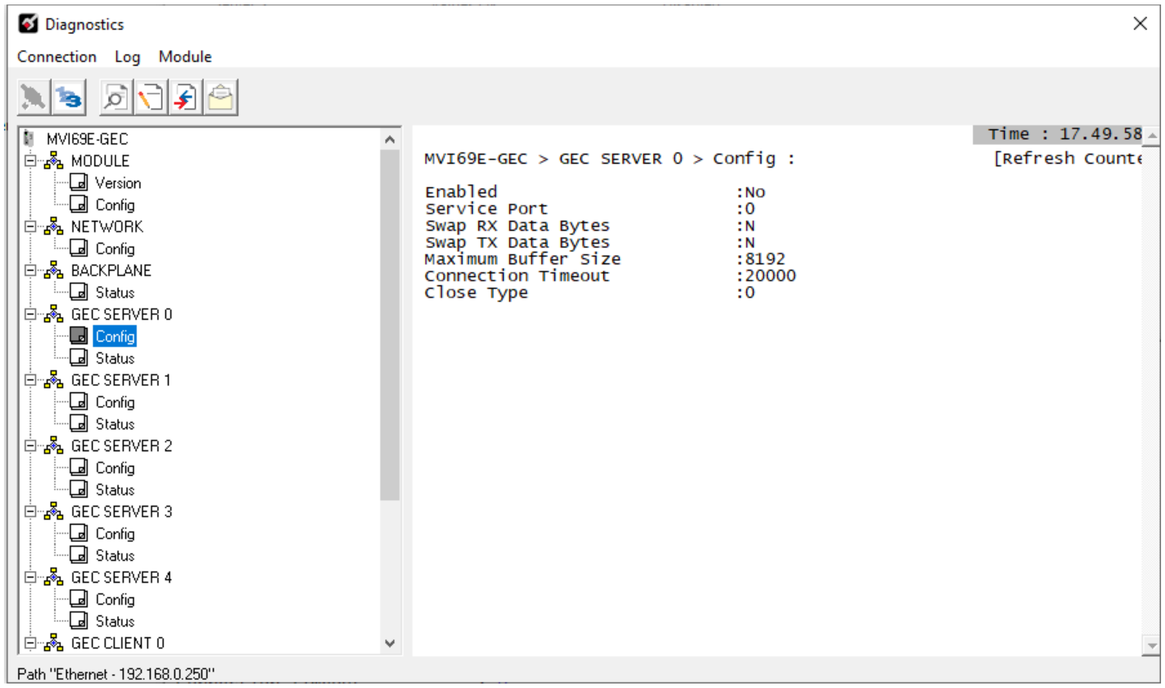
Click at *BACKPLANE* > *Status* to monitor the backplane communication status:



```
Time : 17.46.24
MVI69E-GEC > BACKPLANE > Status :                            [Refresh Counte

Processor State                 :Run
Read                            :26914
Write                           :26915
Parsing                         :26914
Error                           :1
```

Viewing the Backplane Status

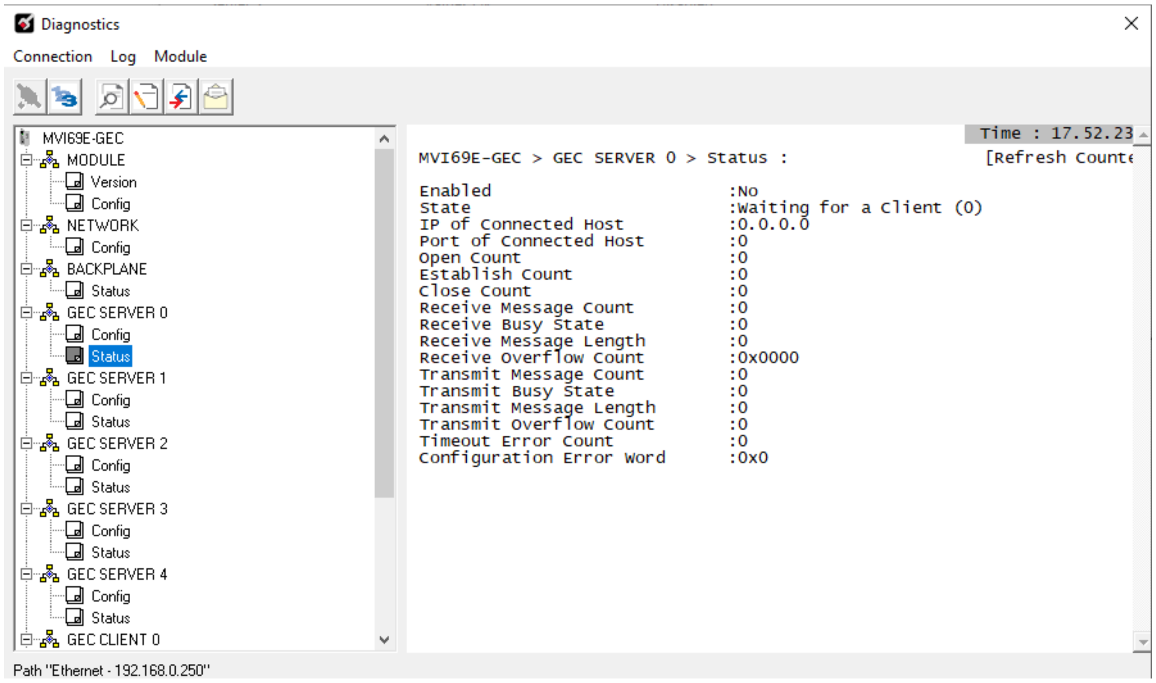
Click at *GEC SERVER X > Config* to display the server configuration settings:



```
Time : 17.49.58
MVI69E-GEC > GEC SERVER 0 > Config :                    [Refresh Counte

Enabled                        :No
Service Port                   :0
Swap RX Data Bytes             :N
Swap TX Data Bytes             :N
Maximum Buffer Size            :8192
Connection Timeout             :20000
Close Type                     :0
```

Viewing the Server Status

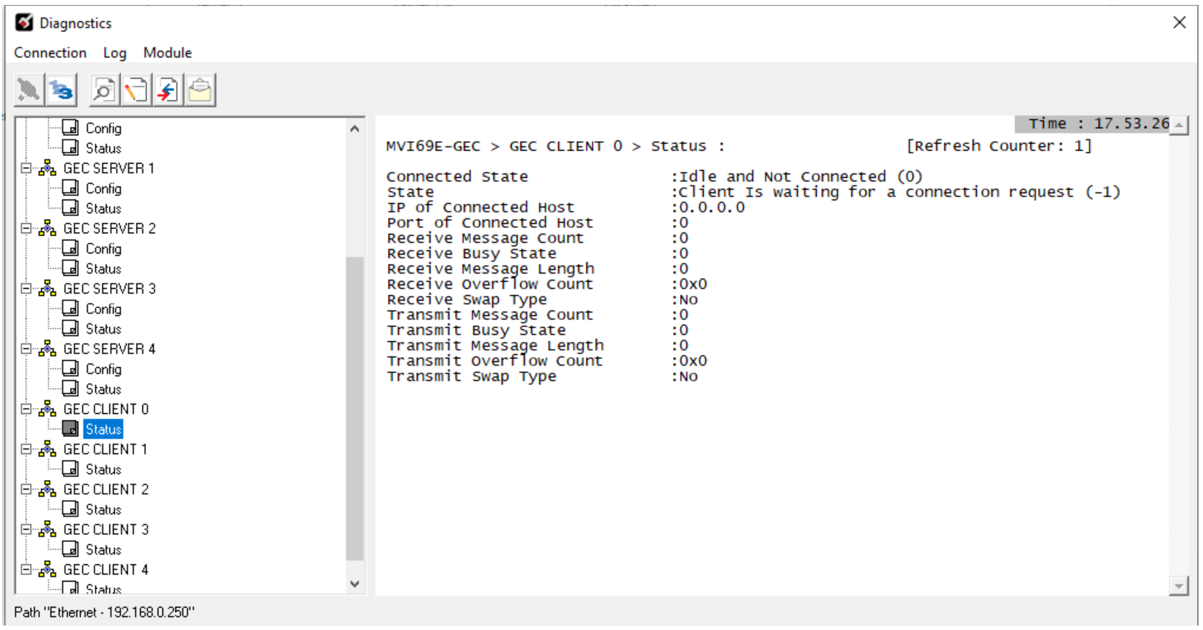
Click at *GEC SERVER X > Status* to display the server status:



```
Time : 17.52.23
MVI69E-GEC > GEC SERVER 0 > Status :                    [Refresh Counte

Enabled                        :No
State                          :Waiting for a Client (0)
IP of Connected Host           :0.0.0.0
Port of Connected Host         :0
Open Count                     :0
Establish Count                :0
Close Count                    :0
Receive Message Count          :0
Receive Busy State             :0
Receive Message Length         :0
Receive Overflow Count         :0x0000
Transmit Message Count         :0
Transmit Busy State            :0
Transmit Message Length        :0
Transmit Overflow Count        :0
Timeout Error Count            :0
Configuration Error Word       :0x0
```

Viewing the Client Status

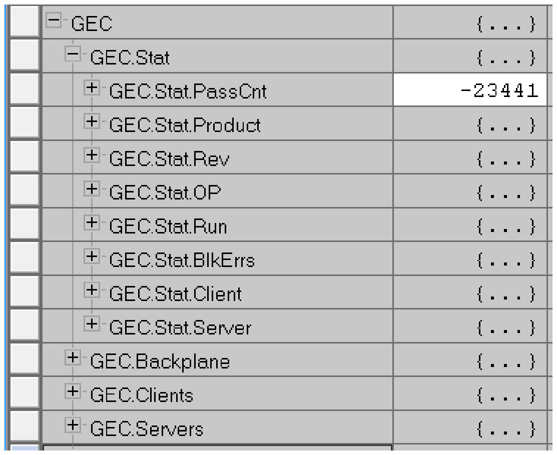
Click at *GEC CLIENT X* > *Status* to display the client status:



```
                                                            Time : 17.53.26
MVI69E-GEC > GEC CLIENT 0 > Status :                     [Refresh Counter: 1]

Connected State              :Idle and Not Connected (0)
State                        :Client Is waiting for a connection request (-1)
IP of Connected Host         :0.0.0.0
Port of Connected Host       :0
Receive Message Count        :0
Receive Busy State           :0
Receive Message Length       :0
Receive Overflow Count       :0x0
Receive Swap Type            :No
Transmit Message Count       :0
Transmit Busy State          :0
Transmit Message Length      :0
Transmit Overflow Count      :0x0
Transmit Swap Type           :No
```

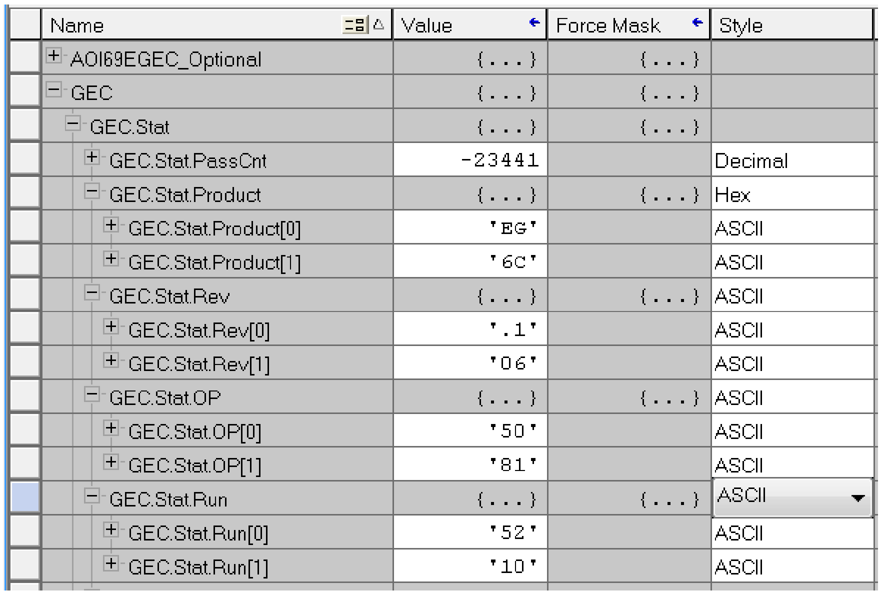
6.3 Diagnostics Using the GEC.Stat Controller Tags

Refer to *GEC.Stat* controller tag for the module status:



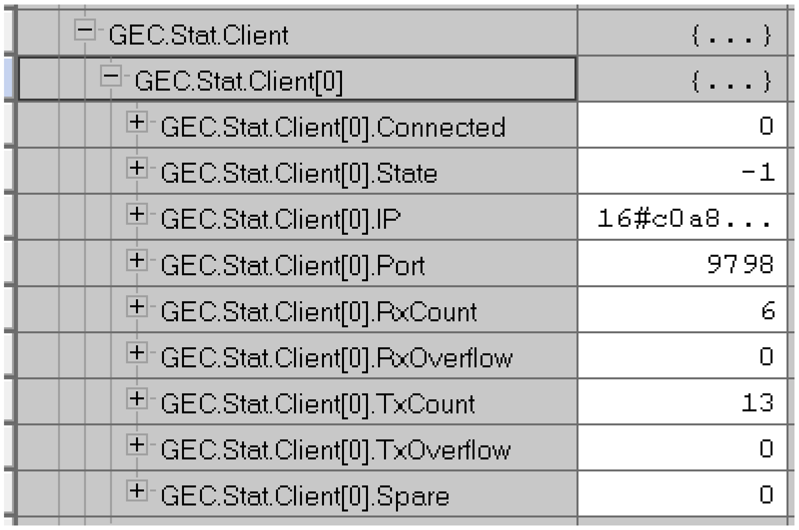
⊟ GEC	{...}
⊟ GEC.Stat	{...}
⊞ GEC.Stat.PassCnt	-23441
⊞ GEC.Stat.Product	{...}
⊞ GEC.Stat.Rev	{...}
⊞ GEC.Stat.OP	{...}
⊞ GEC.Stat.Run	{...}
⊞ GEC.Stat.BlkErrs	{...}
⊞ GEC.Stat.Client	{...}
⊞ GEC.Stat.Server	{...}
⊞ GEC.Backplane	{...}
⊞ GEC.Clients	{...}
⊞ GEC.Servers	{...}

These status tags contain the program scan counter, product name, firmware revision, operating system revision, run number:



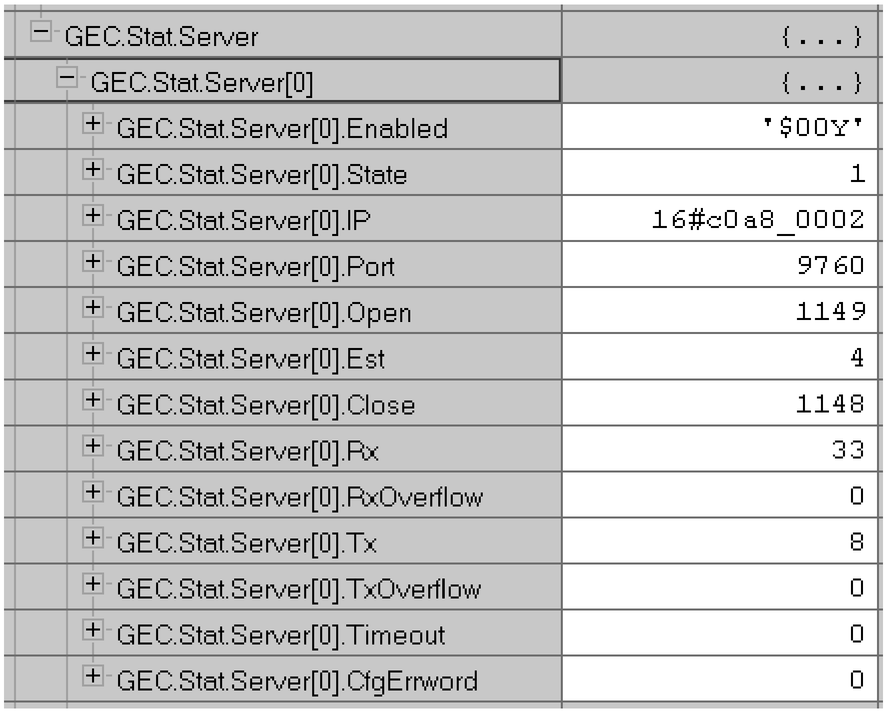
Name	Value	Force Mask	Style
⊞ AOI69EGEC_Optional	{...}	{...}	
⊟ GEC	{...}	{...}	
⊟ GEC.Stat	{...}	{...}	
⊞ GEC.Stat.PassCnt	-23441		Decimal
⊟ GEC.Stat.Product	{...}	{...}	Hex
⊞ GEC.Stat.Product[0]	'EG'		ASCII
⊞ GEC.Stat.Product[1]	'6C'		ASCII
⊟ GEC.Stat.Rev	{...}	{...}	ASCII
⊞ GEC.Stat.Rev[0]	'.1'		ASCII
⊞ GEC.Stat.Rev[1]	'06'		ASCII
⊟ GEC.Stat.OP	{...}	{...}	ASCII
⊞ GEC.Stat.OP[0]	'50'		ASCII
⊞ GEC.Stat.OP[1]	'81'		ASCII
⊟ GEC.Stat.Run	{...}	{...}	ASCII
⊞ GEC.Stat.Run[0]	'52'		ASCII
⊞ GEC.Stat.Run[1]	'10'		ASCII

Refer to *GEC.Stat.Client[x]* array to monitor each client operation:



GEC.Stat.Client	{...}
GEC.Stat.Client[0]	{...}
GEC.Stat.Client[0].Connected	0
GEC.Stat.Client[0].State	-1
GEC.Stat.Client[0].IP	16#c0a8...
GEC.Stat.Client[0].Port	9798
GEC.Stat.Client[0].RxCount	6
GEC.Stat.Client[0].RxOverflow	0
GEC.Stat.Client[0].TxCount	13
GEC.Stat.Client[0].TxOverflow	0
GEC.Stat.Client[0].Spare	0

Refer to *GEC.Stat.Server[x]* array for each server status:

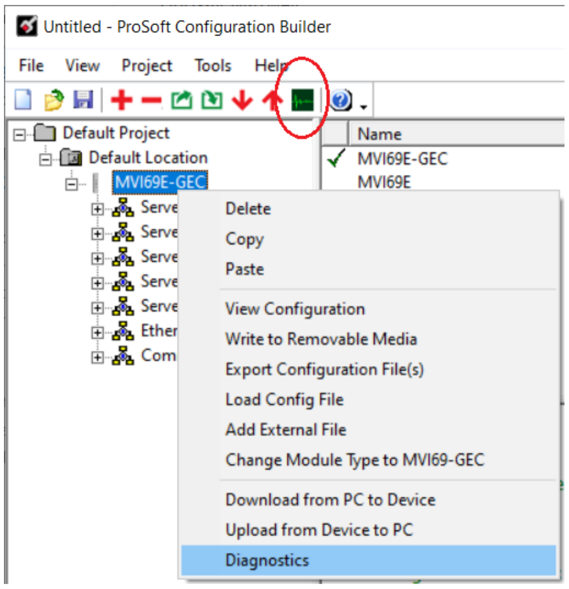


GEC.Stat.Server	{...}
GEC.Stat.Server[0]	{...}
GEC.Stat.Server[0].Enabled	'$00Y'
GEC.Stat.Server[0].State	1
GEC.Stat.Server[0].IP	16#c0a8_0002
GEC.Stat.Server[0].Port	9760
GEC.Stat.Server[0].Open	1149
GEC.Stat.Server[0].Est	4
GEC.Stat.Server[0].Close	1148
GEC.Stat.Server[0].Rx	33
GEC.Stat.Server[0].RxOverflow	0
GEC.Stat.Server[0].Tx	8
GEC.Stat.Server[0].TxOverflow	0
GEC.Stat.Server[0].Timeout	0
GEC.Stat.Server[0].CfgErrword	0

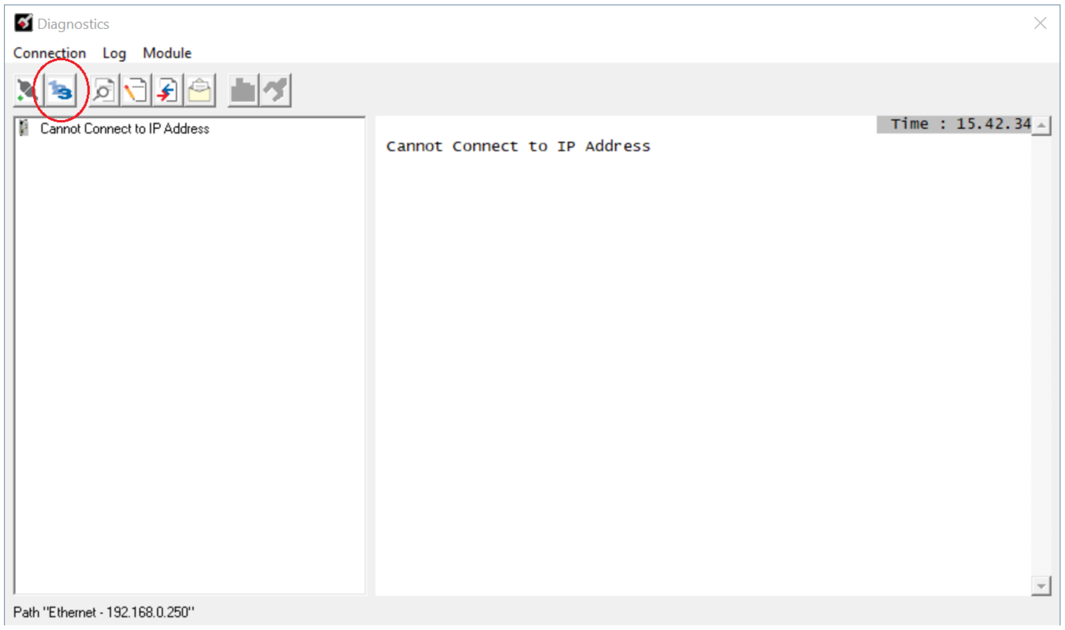
6.4 Assigning a Temporary IP Address Using ProSoft Discovery Service (PDS)

You can assign a temporary IP address to the module using the ProSoft Discovery Service utility. This utility is installed within ProSoft Configuration Builder (PCB).

1 In PCB, click on the *Diagnostics* icon in the shortcuts menu, or right-click on the module's icon and select **DIAGNOSTICS**.



2 The *Diagnostics* dialog opens. Click on the *Connection Setup* icon, or select **CONNECTION > CONNECTION SETUP**.

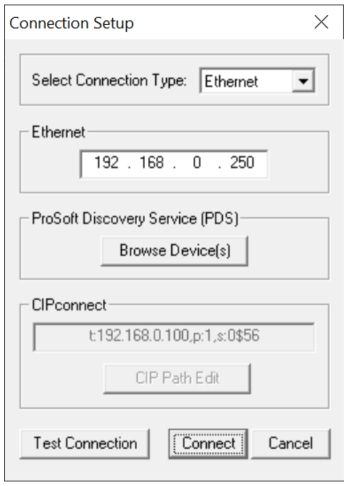


Cannot Connect to IP Address

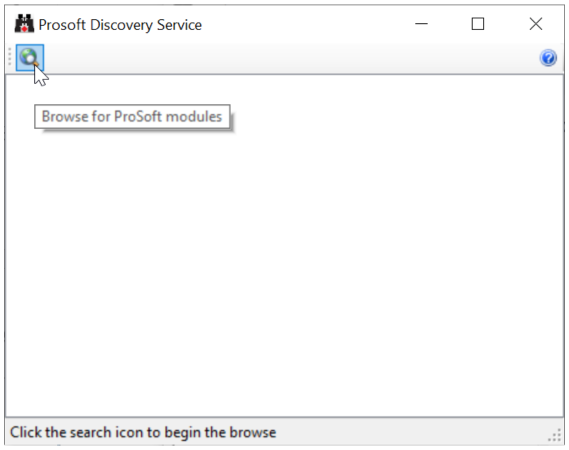
```
Time : 15.42.34
Cannot Connect to IP Address
```

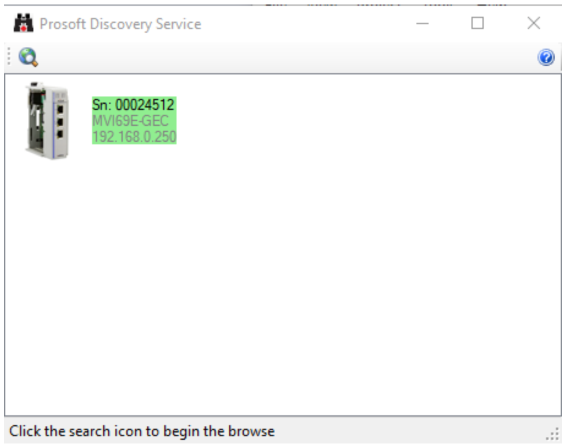
Path "Ethernet - 192.168.0.250"

3 In the *Connection Setup* dialog box, click **BROWSE DEVICE(S)** to start *ProSoft Discovery Service*.



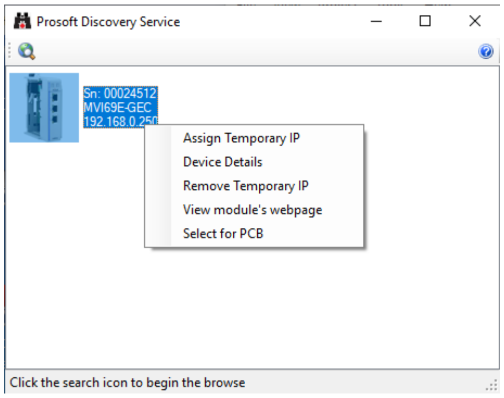
4 ProSoft Discovery Service scans the network for ProSoft devices. Click on the icon to manually initiate a search.


Prosoft Discovery Service
Browse for ProSoft modules
Click the search icon to begin the browse

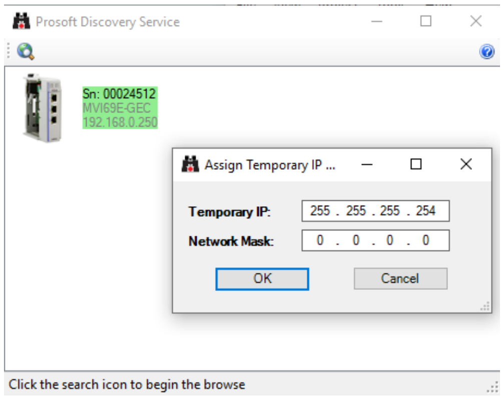

Prosoft Discovery Service
Sn: 00024512
MVI69E-GEC
192.168.0.250
Click the search icon to begin the browse

5 Right-click the module icon and choose **ASSIGN TEMPORARY IP**.

The *Assign Temporary IP feature* might be used to set a second IP address in the same subnet as your PC IP address. The temporary IP address is reset after the module is rebooted.

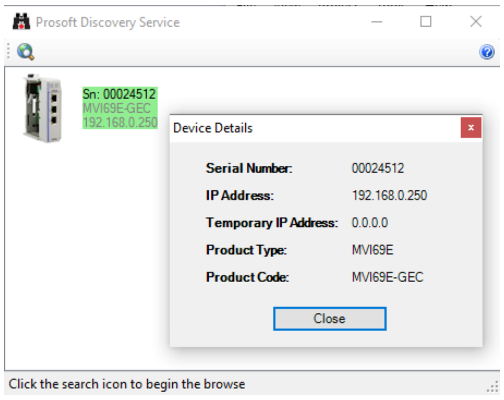


6 Enter the *Temporary IP* and *Network Mask*.


Prosoft Discovery Service
Sn: 00024512
MVI69E-GEC
192.168.0.250
Assign Temporary IP ...
OK
Cancel
Click the search icon to begin the browse

Temporary IP:	255 . 255 . 255 . 254
Network Mask:	0 . 0 . 0 . 0

7 To view the current parameters of the MVI69E-GEC, right-click on the module icon and select *Device Details*.

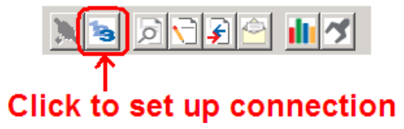

Prosoft Discovery Service
Sn: 00024512
MVI69E-GEC
192.168.0.250
Device Details
Close
Click the search icon to begin the browse

Serial Number:	00024512
IP Address:	192.168.0.250
Temporary IP Address:	0.0.0.0
Product Type:	MVI69E
Product Code:	MVI69E-GEC

6.5 Connecting to the Module’s Webpage

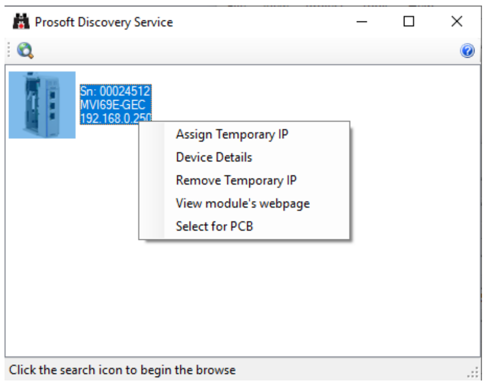
The module’s internal web server provides access to module version and status information, as well as the ability to set the date and time, reboot the module, and download firmware upgrade to the module. Enter the assigned IP address of the module into a web browser or use the following steps in PCB.

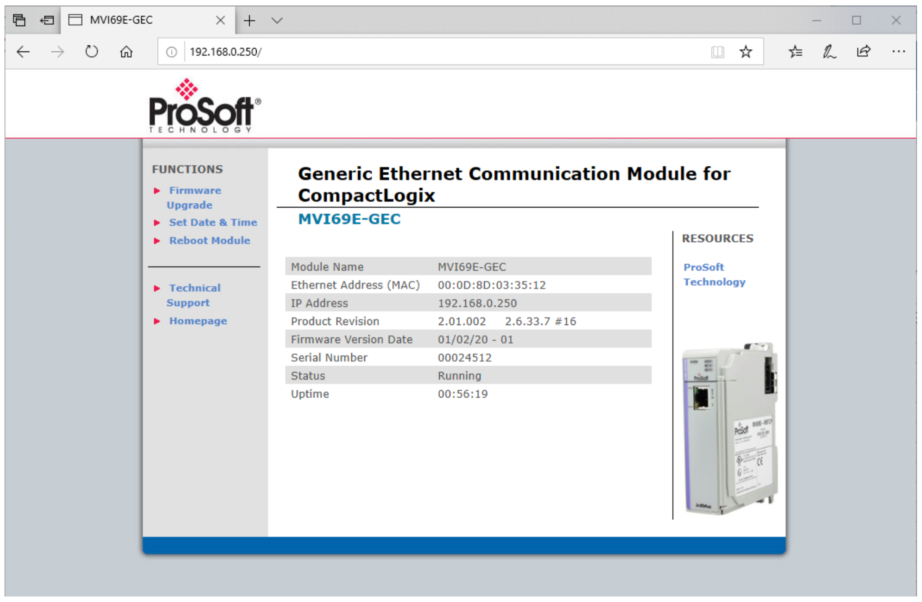
1 In the PCB Diagnostics window, click the **SET UP CONNECTION** button.



2 In the *Connection Setup* dialog box, click **BROWSE DEVICE(S)** to start *ProSoft Discovery Service*.

3 Right-click the module icon and choose **VIEW MODULE’S WEBPAGE** to launch your default browser and display the module’s webpage.





Generic Ethernet Communication Module for CompactLogix

MVI69E-GEC

Module Name	MVI69E-GEC
Ethernet Address (MAC)	00:0D:8D:03:35:12
IP Address	192.168.0.250
Product Revision	2.01.002 2.6.33.7 #16
Firmware Version Date	01/02/20 - 01
Serial Number	00024512
Status	Running
Uptime	00:56:19

7 Reference

7.1 Product Specifications

The MVI69E-GEC Generic ASCII Ethernet Interface module is designed to allow CompactLogix™ / MicroLogix™ processors to interface easily with ASCII devices using the TCP/IP protocol. Compatible devices may be either ASCII instruments with built-in Ethernet or Ethernet connection via a thin server to the existing ASCII device.

Five servers and Clients are present on the module permitting both the reception and transmission of data between the Rockwell Automation processor and attached devices.

The MVI69E-GEC module is a powerful communication interface for CompactLogix / MicroLogix processors. Developed under license from Rockwell Automation, the module incorporates proprietary backplane technology that enables powerful data access between the module and the CompactLogix / MicroLogix processor.

7.1.1 General Specifications

- Single-slot, 1769 backplane-compatible
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module.
- Add-on Instruction creates UDTs, providing logical definitions for I/O, status, and control data.
- Supports CompactLogix processors with 1769 I/O bus capability version 16 and greater (MicroLogix 1500 not supported).

7.1.2 Hardware Specifications

Specification	**Description**
Dimensions	Standard 1769 Single-slot module
Current Load	500 mA max @ 5 VDC Power supply distance rating of 4
Operating Temp.	32° F to 140° F (0° C to 60°C)
Storage Temp.	-40° F to 185° F (-40° C to 85° C)
Relative Humidity	5% to 95% (with no condensation)
LED Indicators	Configuration Status Application Status Backplane Connectivity Status Module Status
Debug/App Port	10/100 Ethernet Port (Auto-negotiating)

7.1.3 Functional Specifications

- Five servers and Clients to receive and/or transmit data
- 10/100 Base-T Ethernet-compatible interface
- Configurable parameters
  - Service port number
  - Connection timeout
  - Close type
- Simple ladder logic operation
- Setup and monitoring through Studio 5000 (CompactLogix) or RS-Logix 500 (MicroLogix)
- CompactLogix backplane interface via I/O access
- Each server monitors
  - State
  - IP and port number of connected Client
  - Error codes
- Each Client monitors
  - State
  - IP and port number of connected server
  - Message-related parameters
- ASCII character strings up to 4096 characters in length supported
- Module error and status conditions returned to processor for diagnostic purposes
  - Module status
  - Port error status word (bitmapped)
  - Port receive state
  - Port receive character count
  - Port receive block count
  - Port transmit state
  - Port transmit character count
  - Port transmit block count
- All data related to the module is contained in a single controller tag with defined objects to simplify configuration, monitoring, and interfacing with the module
- Module configuration and communication configuration data is transferred to the MVI69E-GEC via a pre-defined user data type in the processor

7.2 Functional Overview

7.2.1 General Concepts

Module Power Up

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

1 Initialize hardware components
2 Initialize processor backplane driver
3 Test and clear all RAM
4 Initialize the TCP/IP stack and Ethernet interface
5 Initialize servers and clients

After the module has received the configuration, the module will begin receiving and transmitting messages with devices on the Ethernet network.

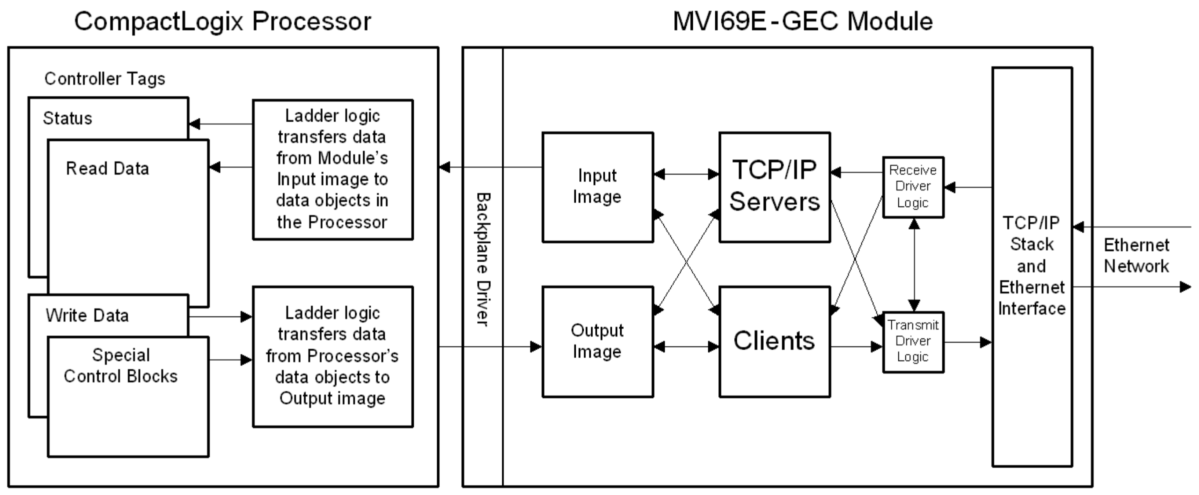
Backplane Data Transfer

The MVI69E-GEC module communicates directly over the CompactLogix or MicroLogix backplane. Data travels between the module and the processor across the backplane using the module's input and output images. The update frequency of the data is determined by the scan rate defined by the user for the module and the communication load on the module. Typical updates are in the range of 2 to 10 milliseconds.

Data received by the servers is placed in the module's input image. This data is processed by the ladder logic in the processor. The input image for the module is set to 60 words. This large data area permits fast throughput of data between the module and the processor.

The processor inserts data in the module's output image to transfer to the module. The module's program extracts the data and transmits the data out to the Ethernet network. Each message is directed to a server that is connected to a client in a remote host. This large data area permits fast throughput of data from the processor to the module.

The following illustration shows the data transfer method used to move data between the processor, the MVI69E-GEC module, and the Ethernet network.



As shown in the previous diagram, all data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the processor to interface the input and output image data defined in the controller tags. The user is responsible for handling and interpreting all data received on the application ports and transferred to the input image. Additionally, the user is responsible for constructing messages to be transferred out of the servers by building the messages in the output image of the module.

Normal Data Transfer

Normal data transfer includes the transferring of data received or to be transmitted on the servers and the status data. These data are transferred through read (input image) and write (output image) blocks. Refer to Module Configuration for a description of the data objects used with the blocks and the ladder logic required.

7.3 Status Data for Block Transfer

If word 1 of the Input Image block is set to -1, the data for the first three servers, the product and block transfer data is sent in the block. The format of this block is as follows:

Parameter	Block Offset Start	Description
Seq Number	0	Sequence number for this block.
Server Index	1	For this status data block, this word is set to a value of -1.
PassCnt	2	Program cycle counter
Product	3	Product name as ASCII string
Rev	5	Revision level as ASCII string
OP	7	Operating system level as ASCII string
Run	9	Run number as ASCII string
BlkErrs.Read	11	Number of blocks transferred from module to processor
BlkErrs.Write	12	Number of blocks transferred from processor to module
BlkErrs.Parse	13	Number of blocks parsed by module
BlkErrs.Err	14	Number of block errors in module
Server[0].Enabled	15	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[0].State	16	This flag defines the current state of the server.
Server[0].IP	17	This double-word value contains the IP address of the client connected to the server.
Server[0].Port	19	This word value contains the port address for the client connected to the server.
Server[0].Open	20	This status value contains the total number of times the server performed an open operation.
Server[0].Established	21	This status value contains the total number of times a connection was established on the socket.
Server[0].Closed	22	This status value contains the total number of times a close operation was performed on the socket.
Server[0].RxCount	23	This status value contains the total number of messages received by the server.
Server[0].RxOverflow	24	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[0].TxCount	25	This status value contains the total number of messages transmitted by the server.
Server[0].TxOverflow	26	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[0].Timeout	27	This status value contains the total number of times a connection timeout occurred on the socket.
Server[0].CfgErrWord	28	This bit mapped word defines the configuration errors for the server.
Server[1].Enabled	29	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[1].State	30	This flag defines the current state of the server.
Server[1].IP	31	This double-word value contains the IP address of the client connected to the server.
Server[1].Port	33	This word value contains the port address for the client connected to the server.
Server[1].Open	34	This status value contains the total number of times the server performed an open operation.

Parameter	**Block Offset Start**	**Description**
Server[1].Established	35	This status value contains the total number of times a connection was established on the socket.
Server[1].Closed	36	This status value contains the total number of times a close operation was performed on the socket.
Server[1].RxCount	37	This status value contains the total number of messages received by the server.
Server[1].RxOverflow	38	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[1].TxCount	39	This status value contains the total number of messages transmitted by the server.
Server[1].TxOverflow	40	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[1].Timeout	41	This status value contains the total number of times a connection timeout occurred on the socket.
Server[1].CfgErrWord	42	This bit mapped word defines the configuration errors for the server.
Server[2].Enabled	43	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[2].State	44	This flag defines the current state of the server.
Server[2].IP	45	This double-word value contains the IP address of the client connected to the server.
Server[2].Port	47	This word value contains the port address for the client connected to the server.
Server[2].Open	48	This status value contains the total number of times the server performed an open operation.
Server[2].Established	49	This status value contains the total number of times a connection was established on the socket.
Server[2].Closed	50	This status value contains the total number of times a close operation was performed on the socket.
Server[2].RxCount	51	This status value contains the total number of messages received by the server.
Server[2].RxOverflow	52	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[2].TxCount	53	This status value contains the total number of messages transmitted by the server.
Server[2].TxOverflow	54	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[2].Timeout	55	This status value contains the total number of times a connection timeout occurred on the socket.
Server[2].CfgErrWord	56	This bit mapped word defines the configuration errors for the server.
Reserved	57	This data area is reserved for future use.
Server States	58 to 62	State of each of the five servers.
Last Write Count	63	This word contains the number of characters written on server from last BTR block.

If word 1 of the Input Image block is set to -2, the data for the last two servers is passed to the processor. The format of this block is as follows:

Parameter	**Block Offset Start**	**Description**
Seq Number	0	Sequence number for this block.
Server Index	1	For this status data block, this word is set to a value of -2.
PassCnt	2	Program cycle counter
Product	3	Product name as ASCII string
Rev	5	Revision level as ASCII string
OP	7	Operating system level as ASCII string
Run	9	Run number as ASCII string
BlkErrs.Read	11	Number of blocks transferred from module to processor
BlkErrs.Write	12	Number of blocks transferred from processor to module
BlkErrs.Parse	13	Number of blocks parsed by module
BlkErrs.Err	14	Number of block errors in module
Server[3].Enabled	15	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[3].State	16	This flag defines the current state of the server.
Server[3].IP	17	This double-word value contains the IP address of the client connected to the server.
Server[3].Port	19	This word value contains the port address for the client connected to the server.
Server[3].Open	20	This status value contains the total number of times the server performed an open operation.
Server[3].Established	21	This status value contains the total number of times a connection was established on the socket.
Server[3].Closed	22	This status value contains the total number of times a close operation was performed on the socket.
Server[3].RxCount	23	This status value contains the total number of messages received by the server.
Server[3].RxOverflow	24	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[3].TxCount	25	This status value contains the total number of messages transmitted by the server.
Server[3].TxOverflow	26	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[3].Timeout	27	This status value contains the total number of times a connection timeout occurred on the socket.
Server[3].CfgErrWord	28	This bit mapped word defines the configuration errors for the server.
Server[4].Enabled	29	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[4].State	30	This flag defines the current state of the server.
Server[4].IP	31	This double-word value contains the IP address of the client connected to the server.
Server[4].Port	33	This word value contains the port address for the client connected to the server.
Server[4].Open	34	This status value contains the total number of times the server performed an open operation.
Server[4].Established	35	This status value contains the total number of times a connection was established on the socket.
Server[4].Closed	36	This status value contains the total number of times a close operation was performed on the socket.

Parameter	Block Offset Start	Description
Server[4].RxCount	37	This status value contains the total number of messages received by the server.
Server[4].RxOverflow	38	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[4].TxCount	39	This status value contains the total number of messages transmitted by the server.
Server[4].TxOverflow	40	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[4].Timeout	41	This status value contains the total number of times a connection timeout occurred on the socket.
Server[4].CfgErrWord	42	This bit mapped word defines the configuration errors for the server.
Reserved	43 to 57	This data area is reserved for future use.
Server States	58 to 62	State of each of the five servers.
Last Write Count	63	This word contains the number of characters written on server from last BTR block.

If word 1 of the Input Image block is set to -3, the data for the first three clients is passed to the processor. The format of this block is as follows:

Parameter	Block Offset Start	Description
Seq Number	0	Sequence number for this block.
Server Index	1	For this status data block, this word is set to a value of -3.
PassCnt	2	Program cycle counter
Product	3	Product name as ASCII string
Rev	5	Revision level as ASCII string
OP	7	Operating system level as ASCII string
Run	9	Run number as ASCII string
BlkErrs.Read	11	Number of blocks transferred from module to processor
BlkErrs.Write	12	Number of blocks transferred from processor to module
BlkErrs.Parse	13	Number of blocks parsed by module
BlkErrs.Err	14	Number of block errors in module
Client[0].Connected	15	This flag defines if the client is utilized and connected to a server. A value of 0 indicates the client is not connected and can be utilized for a connection. Any other value indicates the client is connected and being used.
Client[0].State	16	This flag defines the current state of the client.
Client[0].IP	17	This double-word value contains the IP address of the server connected to the client.
Client[0].Port	19	This word value contains the port address for the server connected to the client.
Client[0].RxCount	20	This status value contains the total number of messages received by the client.
Client[0].RxOverflow	21	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[0].TxCount	22	This status value contains the total number of messages transmitted by the client.
Client[0].TxOverflow	23	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.

Parameter	Block Offset Start	Description
Client[0].spare	24	Reserved for future use
Client[1].Connected	25	This flag defines if the client is utilized and connected to a server. A value of 0 indicates the client is not connected and can be utilized for a connection. Any other value indicates the client is connected and being used.
Client[1].State	26	This flag defines the current state of the client.
Client[1].IP	27	This double-word value contains the IP address of the server connected to the client.
Client[1].Port	29	This word value contains the port address for the server connected to the client.
Client[1].RxCount	30	This status value contains the total number of messages received by the client.
Client[1].RxOverflow	31	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[1].TxCount	32	This status value contains the total number of messages transmitted by the client.
Client[1].TxOverflow	33	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[1].spare	34	Reserved for future use
Client[2].Connected	35	This flag defines if the client is utilized and connected to a server. A value of 0 indicates the client is not connected and can be utilized for a connection. Any other value indicates the client is connected and being used.
Client[2].State	36	This flag defines the current state of the client.
Client[2].IP	37	This double-word value contains the IP address of the server connected to the client.
Client[2].Port	39	This word value contains the port address for the server connected to the client.
Client[2].RxCount	40	This status value contains the total number of messages received by the client.
Client[2].RxOverflow	41	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[2].TxCount	42	This status value contains the total number of messages transmitted by the client.
Client[2].TxOverflow	43	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[2].spare	44	Reserved for future use
Reserved	45 to 57	This data area is reserved for future use.
Server States	58 to 62	State of each of the five servers.
Last Write Count	63	This word contains the number of characters written on server from last BTR block.

If word 1 of the Input Image block is set to -4, the data for the last two clients is passed to the processor. The format of this block is as follows:

Parameter	Block Offset Start	Description
Seq Number	0	Sequence number for this block.
Server Index	1	For this status data block, this word is set to a value of -4.
PassCnt	2	Program cycle counter
Product	3	Product name as ASCII string
Rev	5	Revision level as ASCII string

Parameter	Block Offset Start	Description
OP	7	Operating system level as ASCII string
Run	9	Run number as ASCII string
BlkErrs.Read	11	Number of blocks transferred from module to processor
BlkErrs.Write	12	Number of blocks transferred from processor to module
BlkErrs.Parse	13	Number of blocks parsed by module
BlkErrs.Err	14	Number of block errors in module
Client[3].Connected	15	This flag defines if the client is utilized and connected to a server, with a non-zero value. A value of 0 indicates the client is not connected and can be utilized for a connection.
Client[3].State	16	This flag defines the current state of the client.
Client[3].IP	17	This double-word value contains the IP address of the server connected to the client.
Client[3].Port	19	This word value contains the port address for the server connected to the client.
Client[3].RxCount	20	This value contains the total number of messages received by the client.
Client[3].RxOverflow	21	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[3].TxCount	22	This status value contains the total number of messages transmitted by the client.
Client[3].TxOverflow	23	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[3].spare	24	Reserved for future use
Client[4].Connected	25	This flag defines if the client is utilized and connected to a server, with a non-zero value. A value of 0 indicates the client is not connected and can be utilized for a connection.
Client[4].State	26	This flag defines the current state of the client.
Client[4].IP	27	This double-word value contains the IP address of the server connected to the client.
Client[4].Port	29	This word value contains the port address for the server connected to the client.
Client[4].RxCount	30	This value contains the total number of messages received by the client.
Client[4].RxOverflow	31	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[4].TxCount	32	This status value contains the total number of messages transmitted by the client.
Client[4].TxOverflow	33	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[4].spare	34	Reserved for future use
Reserved	35 to 57	Reserved for future use.
Server States	58 to 62	State of each of the five servers.
Last Write Count	63	This word contains the number of characters written on server from last BTR block.

8 Support, Service & Warranty

8.1 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 serial, Ethernet or Fieldbus devices interfaced to the module, if any.

Note: *For technical support calls within the United States, ProSoft's 24/7 after-hours phone support is available for urgent plant-down issues. Detailed contact information for all our worldwide locations is available on the following page.*

Asia Pacific	**Europe / Middle East / Africa**
Regional Office Phone: +60.3.2247.1898 asiapc@prosoft-technology.com Languages spoken: Bahasa, Chinese, English, Japanese, Korean REGIONAL TECH SUPPORT support.ap@prosoft-technology.com **North Asia (China, Hong Kong)** Phone: +86.21.5187.7337 china@prosoft-technology.com Languages spoken: Chinese, English REGIONAL TECH SUPPORT support.ap@prosoft-technology.com **Southwest Asia (India, Pakistan)** Phone: +91.98.1063.7873 india@prosoft-technology.com Languages spoken: English, Hindi, Urdu **Australasia (Australia, New Zealand)** Phone: +60.3.7941.2888 pacific@prosoft-technology.com Language spoken: English **Southeast Asia (Singapore, Indonesia, Philippines)** Phone: +60.3.7941.2888 seasia@prosoft-technology.com Languages spoken: English, Bahasa, Tamil **Northeast & Southeast Asia (Japan, Taiwan, Thailand, Vietnam, Malaysia)** Phone: +60.3.7941.2888 neasia@prosoft-technology.com Languages spoken: English, Chinese, Japanese **Korea** Phone: +60.3.7941.2888 korea@prosoft-technology.com Languages spoken: English, Korean	Regional Office Phone: +33.(0)5.34.36.87.20 europe@prosoft-technology.com Languages spoken: French, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com **Middle East & Africa** Phone: +971.4.214.6911 mea@prosoft-technology.com Languages spoken: Hindi, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com **North Western Europe (UK, IE, IS, DK, NO, SE)** Phone: +44.(0)7415.864.902 nweurope@prosoft-technology.com Language spoken: English **Central & Eastern Europe, Finland** Phone: +48.22.250.2546 centraleurope@prosoft-technology.com Languages spoken: Polish, English **Russia & CIS** Phone: +7.499.704.53.46 russia@prosoft-technology.com Language spoken: Russian, English **Austria, Germany, Switzerland** Phone: +49.(0)1511.465.4200 germany@prosoft-technology.com Language spoken: German, English **BeNeLux, France, North Africa** Phone: +33(0)5.34.36.87.20 france@prosoft-technology.com Languages spoken: French, English **Mediterranean Countries** Phone: +39.342.8651.595 italy@prosoft-technology.com Languages spoken: Italian, English, Spanish

Latin and South America	North America
Brazil, Argentina, Uruguay Phone: +55.11.5084.5178 brasil@prosoft-technology.com Languages spoken: Portuguese, Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com **Mexico** Phone: +52.222.264.1814 mexico@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com **Andean Countries, Central America, Caribbean, Chile, Bolivia, Paraguay** Phone: +507.6427.48.38 andean@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com	Regional Office Phone: +1.661.716.5100 info@prosoft-technology.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT support@prosoft-technology.com

8.2 Warranty Information

For complete details regarding ProSoft Technology's TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS, go to www.prosoft-technology.com/legal

Documentation is subject to change without notice.

Index