Supplement for OCS Ethernet

SUP0740-12-EN_Ethernet

June 26, 2019

PREFACE

For user documentation updates, visit our website.

North America:
Tel: 317 916-4274
Fax: 317 639-4279
Web: http:/hornerautomation.com
Email: techsppt@heapg.com

Europe:
Tel: (+) 353-21-4321-266
Fax: (+) 353-21-4321826
Web: http://www.horner-apg.com
Email: tech.support@horner-apg.com

LIMITED WARRANTY AND LIMITATION OF LIABILITY

Horner APG, LLC ("HE-APG") warrants to the original purchaser that the ETN200 / 300 manufactured by HE-APG is free from defects in material and workmanship under normal use and service. The obligation of HE-APG under this warranty shall be limited to the repair or exchange of any part or parts which may prove defective under normal use and service within two (2) years from the date of manufacture or eighteen (18) months from the date of installation by the original purchaser whichever occurs first, such defect to be disclosed to the satisfaction of HE-APG after examination by HE-APG of the allegedly defective part or parts. THIS WARRANTY IS EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE AND OF ALL OTHER OBLIGATIONS OR LIABILITIES AND HE-APG NEITHER ASSUMES, NOR AUTHORIZES ANY OTHER PERSON TO ASSUME FOR HE-APG, ANY OTHER LIABILITY IN CONNECTION WITH THE SALE OF THIS MODULE. THIS WARRANTY SHALL NOT APPLY TO THIS MODULE OR ANY PART THEREOF THAT HAS BEEN SUBJECT TO ACCIDENT, NEGLIGENCE, ALTERATION, ABUSE, OR MISUSE. HE-APG MAKES NO WARRANTY WHATSOEVER IN RESPECT TO ACCESSORIES OR PARTS NOT SUPPLIED BY HE-APG. THE TERM "ORIGINAL PURCHASER", AS USED IN THIS WARRANTY, SHALL BE DEEMED TO MEAN THAT PERSON FOR WHOM THIS MODULE IS ORIGINALLY INSTALLED. THIS WARRANTY SHALL APPLY ONLY WITHIN THE BOUNDARIES OF THE CONTINENTAL UNITED STATES.

In no event, whether as a result of breach of contract, warranty, tort (including negligence) or otherwise, shall HE-APG or its suppliers be liable of any special, consequential, incidental or penal damages including, but not limited to, loss of profit or revenues, loss of use of the products or any associated equipment, damage to associated equipment, cost of capital, cost of substitute products, facilities, services or replacement power, down time costs, or claims of original purchaser's customers for such damages.

To obtain warranty service, return the product to your distributor with a description of the problem, proof of purchase, post paid, insured and in a suitable package.

ABOUT PROGRAMMING EXAMPLES

Any example programs and program segments in this manual or provided on accompanying media are included solely for illustrative purposes. Due to the many variables and requirements associated with any particular installation, Horner APG cannot assume responsibility or liability for actual use based on the examples and diagrams. It is the sole responsibility of the system designer utilizing this module to appropriately design the end system, to appropriately integrate this and to make safety provisions for the end equipment as is usual and customary in industrial applications as defined in any codes or standards which apply.

NOTE: The programming examples shown in this manual are for illustrative purposes only. Proper machine operation is the sole responsibility of the system integrator.

Visual Map of Key Chapters

FIRST STEP of ANY TASK: DATASHEET			
To obtain updates to datasheets, manuals and user documentation, visit a Horner website. North America: https://hornerautomation.com Europe: http://www.horner-apg.com			
PART ONE: Introduction & Configuration	**PART TWO:** Built-In Industrial Ethernet Protocols	**PART THREE:** Downloadable Industrial Protocol	**PART FOUR:** Non-Industrial Ethernet Protocols
Introduction page 8	CsCAN over Ethernet page 20	Modbus TCP Client Protocol page 88	Internet Control Message Protocol (ICMP) page 104
Installation page 11	Ethernet Global Data Protocol (EGD) page 31		FTP Server Protocol page 1071
General Configuration page 14	Service Request Transfer Protocol page 77		HTTP Server Protocol page 115
	Modbus TCP Server Protocol page 80		Email (SMTP Protocol) page 128
	Ethernet / IP Protocol page 83		ASCII over TCP/IP page 140
			WebMI page 143
			Safety Information page 145
			Technical Information page 146

TABLE OF CONTENTS

PART ONE: INTRODUCTION AND CONFIGURATION

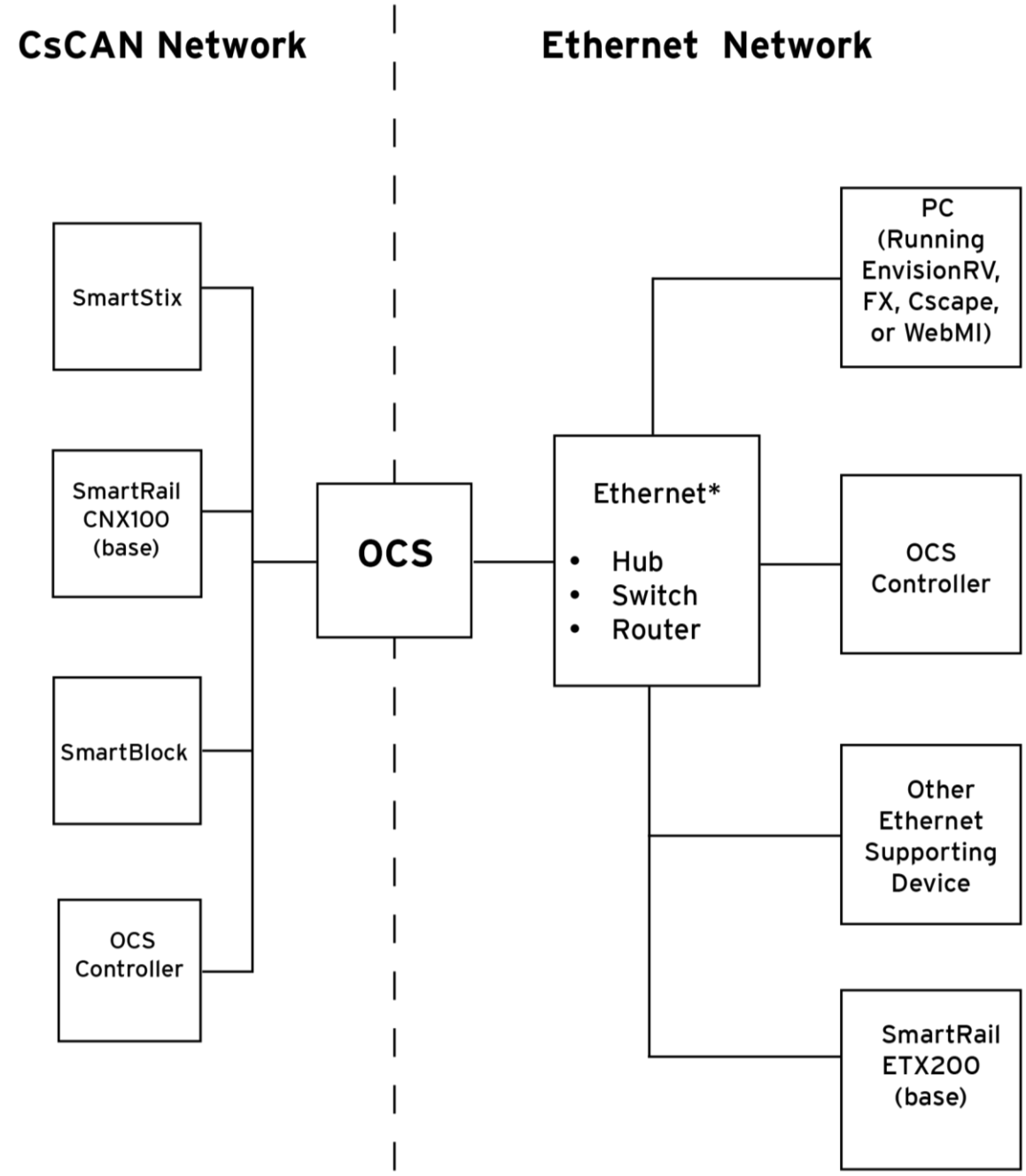
CHAPTER 1: INTRODUCTION

1.1 Ethernet Overview

Table 1.1 – Ethernet Protocols and Features	
ICMP Ping	Internet Control Message Protocol
EGD (Peer)	GE Fanuc Ethernet Global Data
SRTP Server	GE Fanuc Service Request Transfer Protocol
Modbus TCP	Modbus over Ethernet
CsCAN TCP Server	Horner APG CsCAN over Ethernet
Ethernet / IP Server	ODVA CIP over Ethernet
FTP Server	File Transfer Protocol
HTTP Server	Hypertext Transfer Protocol (Web Server)
Half/Full Duplex	Auto-Negotiated Ethernet Mode
SMTP	E-mail
10/100 Mbps	Auto-Negotiated Ethernet Speed
ASCII over TCP/IP	American Standard for Information Interchange over Ethernet
NTP Protocol	Obtain clock from web-based server
Extended Registers	Access to %R2049 - %R9999

Table 1.2 – Ethernet Protocols by Controller Series									
Controller	ICMP	EGD	SRTP	Modbus TCP	Ethernet/ IP	FTP	HTTP	ASCII over TCP/IP	NTP Protocol
XLEe/XLTe	✓	✓	✓	✓	✓	✓		✓	
XL4e	✓	✓	✓	✓	✓	✓	✓	✓	✓
XL7e	✓	✓	✓	✓	✓	✓	✓	✓	✓
EXL6e	✓	✓	✓	✓	✓	✓	✓	✓	✓
EXL10e	✓	✓	✓	✓	✓	✓	✓	✓	✓
XL+	✓	✓	✓	✓	✓	✓	✓	✓	
X5	✓	✓	✓	✓	✓	✓	✓	✓	✓
RCC Series	✓	✓	✓	✓	✓	✓	✓	✓	✓

1.2 Ethernet Network Wiring



* An Ethernet hub or multicast router is recommended for Ethernet Global Data (EGD) protocol because a switch might not be able to pass multicast messages.

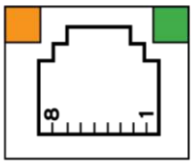
Figure 1.1 – Ethernet Network Wiring

Table 1.3 – Specifications	
General	
Ethernet Speeds	100 BaseTX Fast Ethernet (100Mbps) XL+ Only: 1000 BaseTX (1 Gigabit)
Ethernet Modes	Half or Full Duplex
Ethernet Auto-Negotiation	Both 100 / 1000Mbps and Half / Full Duplex
Ethernet Connector Type	RJ-45
Ethernet Cable Type Recommendation	CAT5 (or better) UTP
Ethernet Port (Applicable to QX351and XL6e)	Auto MDI/MDI-X
Application Protocols	
ICMP	Ping Only
CsCAN TCP Server	Maximum Connections = 8
EGD (Ethernet Global Data)	Maximum Exchanges = 127 Maximum Data Bytes per Exchange = 1,400
SRTP Server	Maximum Connections = 16
Modbus TCP	Maximum Connections = 16
Ethernet / IP Server	Maximum Connections = 2
FTP Server	Maximum Connections = 4
HTTP Server [non-WebMI]	Maximum Connections = 1
WebMI Server [non-HTTP]	Maximum Connections = 4
ASCII over TCP/IP	

CHAPTER 2: INSTALLATION

2.1 Connections for Ethernet

Table 2.1 – LAN Ports		
Device	One Ethernet Port	Two Ethernet Ports
XLEe/XLTe	✓	
XL4	✓	
XL7		✓
EXL6	✓	
EXL10		✓
XL+		✓
X5	✓	
RCC972	✓	
RCC1410	✓	
RCC2414		✓
RCC8842	✓	



Green LED indicates link – when illuminated, data communication is available.

Orange LED indicates activity – when flashing, data is in transmission.

LAN1 or LAN2 ports are available for Ethernet.

NOTE: Not all models use Green and Orange LED lights. Also, if a light is illumination (not flashing) this indicates that data communication. If a light is flashing, this indicates data is in transmission.

2.2 Network Administrator Installation Notes

When connecting an OCS to a local network, the following information is provided to the Network Administrator, as an aid in configuring Ethernet hubs, routers, switches, gateways and servers.

2.2.1 UDP AND TCP PORTS

Each protocol supported by the OCS (except ICMP) uses one or more UDP and/or TCP Ports as the destination port for all messaging, as shown in Table 2.1. Required port usage for the supported protocols should be considered when configuring Ethernet routers and gateways.

NOTE: Any port can be used as the source port.

Table 2.2 – Ethernet Protocol UDP and TCP Port Usage

Ethernet Protocol	UDP Port		TCP Port	
	Hexadecimal	Decimal	Hexadecimal	Decimal
EGD	4746	18246		
SRTP			4745	18245
Modbus			01F6	502
CsCAN over Ethernet			4845	18501
Ethernet / IP	08AE	2222	AF12	44818
FTP			0014 and 0015	20 and 21
HTTP			0050	80
HTTPS			01BB	443
SMTP	0035	53	0019	25
ICMP (Ping)	N/A		N/A	
ASCII over TCP/IP	N/A		User Configurable	

NOTE: SMTP TCP port is user configurable, but in **Cscape → Messaging → Email** feature if user selects '**Obtain SMTP Server IP Address from DNS Server**' then UDP connection will be used to obtain Server IP, in such case UDP port 53 (Decimal) will be used.

2.2.2 INTERNET CONNECTIVITY

Since the OCS uses a standard TCP/IP protocol stack (powered by NetX and ThreadX), it can communicate beyond the local network, and on the internet, for all protocols except EGD. To do so, the OCS must be configured with the IP Address of a network gateway server, which allows communication outside the local network. See **Default Gateway** configuration under Step 5 of Section 3.1 in this manual for details.

NOTE: As network complexity increases, due to Ethernet hubs, routers, switches, gateways, and the internet, the worst-case network delay increases. In many cases, the client software must be configured to account for this time lag. For example, Cscape's **Timeout** can be adjusted as shown under Step 2 of Section 4.2.

2.2.3 IMPORTANT: WIRELESS NETWORK CONSIDERATIONS

When using wireless equipment with industrial networks, make sure the system is designed and installed by personnel that have been trained to use wireless networks in industrial environments. Site surveys, selection of equipment, and installation can be critical in network performance.

In general, the 802.11**b**-based equipment is not a good choice in industrial environments. The frequencies and modulation techniques used in the "**b**" standard are very susceptible to multi-path interference in industrial environments (large metal objects, dense walls and floors, etc.) The 802.11**a** and 802.11**g** are less susceptible to this interference.

Using UDP based protocols, such as Ethernet Global Data (EGD), must be carefully considered when using wireless networks. Wireless networks are more likely to lose or damage communication packets. Many UDP based protocols, including EGD, do not detect and retransmit lost or damaged packets and depend on periodic data transmissions to compensate for this lost data. If your system requires a UDP protocol with a wireless network, make sure it is designed such that random periods without refreshed data do not adversely affect the operation of your system.

Depending on the architecture of the wireless network and the protocols used, wireless networks often produce collisions and extra data packets that are not experienced when using a traditional wire and switch-based network. The extra collisions and traffic coupled with the typically lower bandwidth and higher latency of wireless networks can cause degradation in performance.

CHAPTER 3: GENERAL CONFIGURATION

NOTE: The following configuration is required for all applications regardless of the protocols used. Additional configuration procedures must be performed for each protocol used as described in the configuration sections of the next several chapters.

3.1 Ethernet Configuration

To configure the OCS, use Cscape programming software to perform the following six steps:

1. On the main Cscape screen, select the **Controller** menu and its **Hardware Configure** sub-menu to open the Hardware Configuration dialog (Figure 3.1).

2. If configuring a different OCS model than the one shown in the Hardware Configuration dialog, click on the drop-down boxes for Series and Device Type and select the desired OCS model.

3. Click the **Config** button to the right of LAN1 for LAN 1 or LAN2 for LAN2, revealing the OCS Configuration dialog as shown in Figure 3.2.

NOTE: Figure 3.1 shows a typical Hardware Configuration dialog for OCS Models.

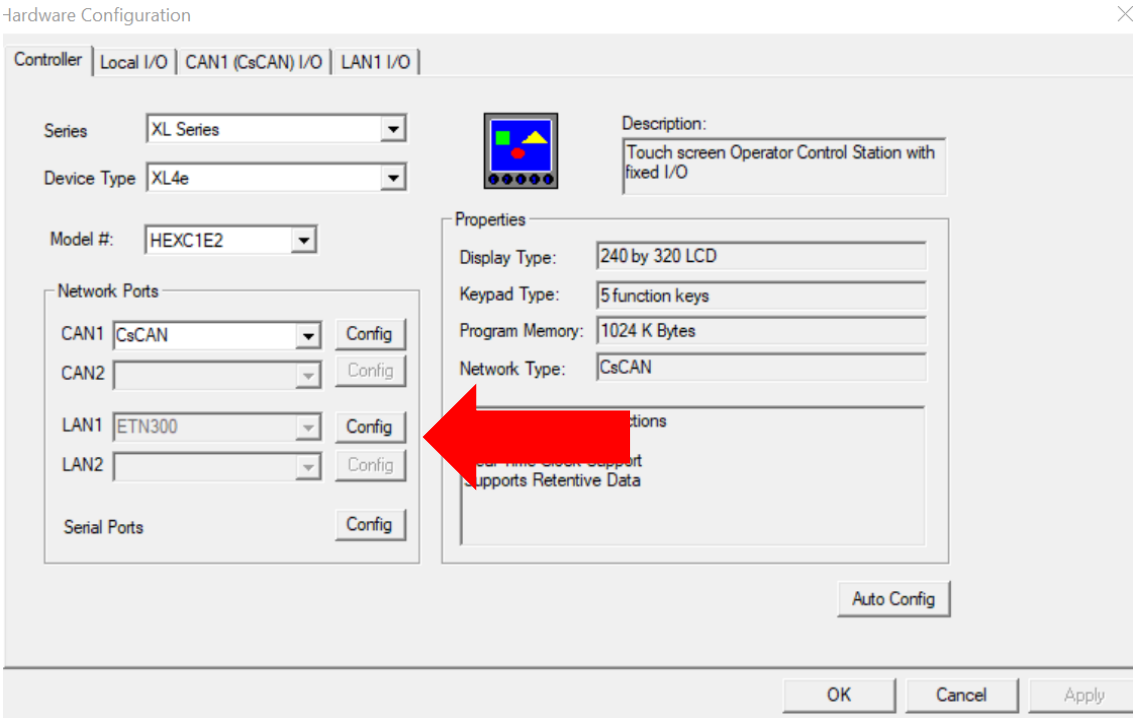


Figure 3-1 – Hardware Configuration Dialog – OCS Models with Built-In Ethernet

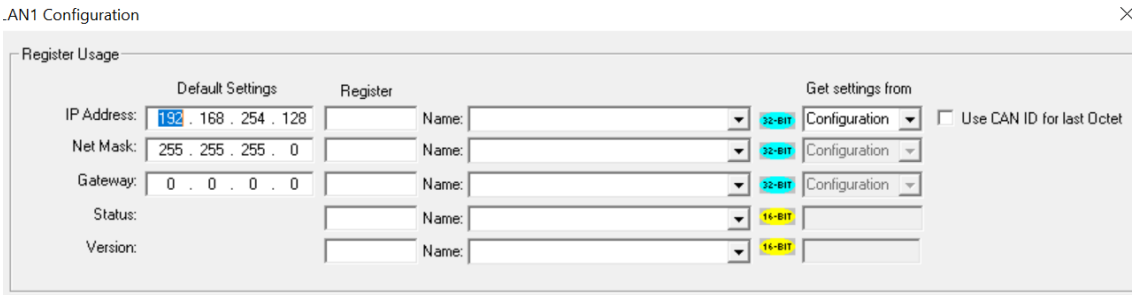


Figure 3.2– LAN1 Configuration Dialog – Register Usage

5. Configure the OCS parameters as follows using Figure 3.2 as a reference.

3.1.1 REGISTER USAGE:

- **IP Address:** Enter the static IP Address for the OCS being configured.

  NOTE: IP Addresses are entered as four numbers, each ranging from 0 to 255. These four numbers are called octets and they are always separated by decimal points.

- **Net Mask:** Enter the Net Mask (sometimes called Subnet Mask) being used by all nodes on the local network. Typical local networks use Class C IP Addresses, in which case the low octet (rightmost number) is used to uniquely identify each node on the local network. In this case, the default Net Mask value of 255.255.255.0 should be used.

- **Gateway:** Enter the IP Address of a Gateway Server on the local network that allows for communication outside of the local network. To prevent the OCS from communicating outside the local network, set the Default Gateway IP Address to 0.0.0.0 (the default setting).

- **Status Register:** Enter an OCS Register reference (such as %R100) to indicate which 16-bit OCS register will have the Ethernet Status word written to it. Table 3.1 shows how this register value is formatted and explains the meaning of each bit in the Status Word.

Table 3.1 - Ethernet Status Word Register Format															
High Byte								Low Byte							
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0	0	Dup	Spd	0	Rx	Tx	Link	TCP Connections							

Status Bit(s)	Status Indication	Status Values	
		Minimum	Maximum
0	Reserved	Always 0	
Dup	Link Duplex (Auto-Negotiated)	0 = Half Duplex	1 = Full Duplex
Spd	Link Speed (Auto-Negotiated)	0 = 10 Mbps	1 = 100 Mbps
Rx	Receive State	0 = Inactive	1 = Active
Tx	Transmit State	0 = Inactive	1 = Active
Link	Link State	0 = Down	1 = Up
TCP Connections	Total Number of Active TCP Connections (CsCAN, SRTP, Modbus, EIP, FTP, WebMI/HTTP, NTP)	0	40

- **Version Register:** Enter an OCS Register reference (such as %R101) to indicate which 16-bit OCS register will have the Ethernet Firmware Version written to it. The value stored in the Version Register is: (Ethernet Firmware Version * 100). For example, for Ethernet Firmware Version 4.30, the Version register will contain 430. For controllers with built-in Ethernet hardware, this Firmware Version is only changed when the Firmware for the OCS is updated

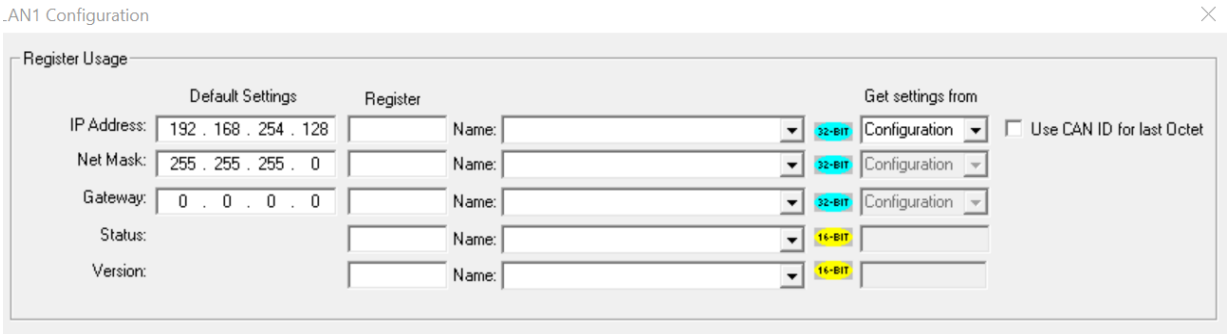
- For the Status and Version registers (if configured), the **Direction** settings are always **Read Only**

- **Use CAN ID for last Octet:** The **Use CAN ID for last Octet** checkbox does not affect Net Mask, Gateway, Status or Version configuration. If the checkbox is checked then it behaves as follows:

    a. If the **IP Address Direction** combo box is **Read / Write**, the **Use CAN ID for last Octet** checkbox will be unchecked and grayed.
    b. If the **IP Address Direction** combo box is empty or **Read Only**, the **Use CAN ID for last Octet** checkbox will be ungrayed and can then be unchecked or checked.
    c. If the **Use CAN ID for last Octet** checkbox is checked, the unit's 8-bit CAN Network ID replaces the last (rightmost) octet of the **Default IP Address**, and the combined result will be the unit's IP Address. In this case, if the **IP Address Register** edit box contains a valid OCS register, the indicated register will be loaded with the combined IP Address.

3.2 OCS IP Address

The OCS obtains its IP Address in one of three different ways, depending on how the **Use CAN ID for last Octet & IP Address direction** checkboxes are configured, as described in the following three sections.



3.2.1 STATIC IP ADDRESS WITH CAN ID

In this mode, the OCS's IP Address comes from a combination of the **IP Address** parameter and the OCS/RCS CAN Network ID. The most significant (leftmost) three octets of the IP Address come from the **IP Address** parameter. The least significant (rightmost) octet of the IP Address is taken from the OCS (or RCS) CAN Network ID. In this case the OCS writes the *adjusted* IP Address to the 32-bit OCS register indicated by the **IP Address Register** parameter.

NOTE: Every time a Hardware Configuration is successfully downloaded to an OCS with an OCS, the static **IP Address**, **Net Mask** and **Default Gateway** parameters are stored in non-volatile memory. In the event of a future unsuccessful I/O configuration download, the OCS will communicate using these 3 stored parameters. This is done to minimize potential loss of communication, which would require direct on-site intervention to correct.

3.3 Ethernet Configuration – IP Parameters

For primary operation, the IP address, Net Mask, and Gateway should be set in the LAN config of the **Cscape Hardware Configuration**. There are options to get IP parameters from the LAN Config or to get parameters from registers. It is possible to set the Ethernet IP parameters from the OCS System Menu, but only as a temporary measure. The following points on IP parameter configuration should be considered.

- **IP Parameters in Non-Volatile RAM:** The IP parameters of the Cscape LAN Config are written to non-volatile RAM on power down. IP parameter settings made in the System Menu are not written to non-volatile RAM. Any IP parameters settings made in the system menu will be lost after cycling power to the unit. It will revert to the last downloaded Cscape LAN Config that was loaded into non-volatile RAM at power down.

- **"Cscape LAN Config"/ "Get Settings from" Configuration**: When 'Get settings from' is set to Configuration, the IP parameters specified under 'Default Settings' is used after downloading to the controller. The IP parameters are represented in System Menu / Set Networks and can be edited. However, any edits made from System Menu / Set Networks is not retained through a power cycle. After power cycle, the unit reverts to the last downloaded Cscape LAN Config that was loaded into non-volatile RAM at power down.

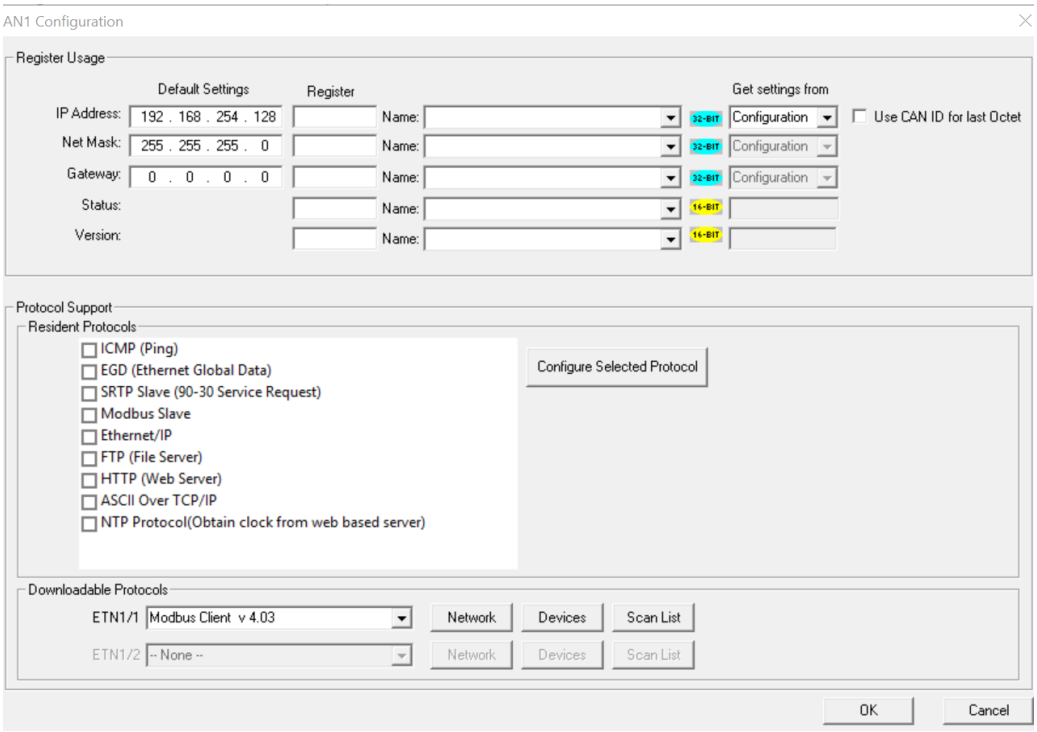
- **"Cscape LAN Config" / "Get Settings from" Register**: When 'Get settings from' is set to Register, the IP parameters are retrieved from the OCS registers assigned in LAN Config. Configured registers must be populated with the desired IP parameters.

  - The IP parameters are represented in System Menu / Set Networks.
  - The IP parameters cannot be edited from System Menu / Set Networks while the unit is in run mode.
  - The IP parameters always follow the values in the registers unless the OCS unit is placed in idle mode.

  Then the IP parameters can be edited in **System Menu / Set Networks**. When the OCS is placed back into run mode, it reverts to the registers for IP parameters.

3.4 Downloadable Protocols

The Downloadable Protocol (Protocol Support) area contains a list of all the protocols supported by the unit being configured. To activate a protocol, check its checkbox.



Click on a listed protocol to select it and then click the Configure Selected Protocol button. This will open a new dialog with configuration options for the selected protocol.

PART TWO: BUILT-IN (RESIDENT) INDUSTRIAL PROTOCOLS

CHAPTER 4: CSCAN OVER ETHERNET PROTOCOL

4.1 CsCAN over Ethernet Overview

This chapter describes CsCAN TCP Server Communication protocol, also known as CsCAN over Ethernet protocol.

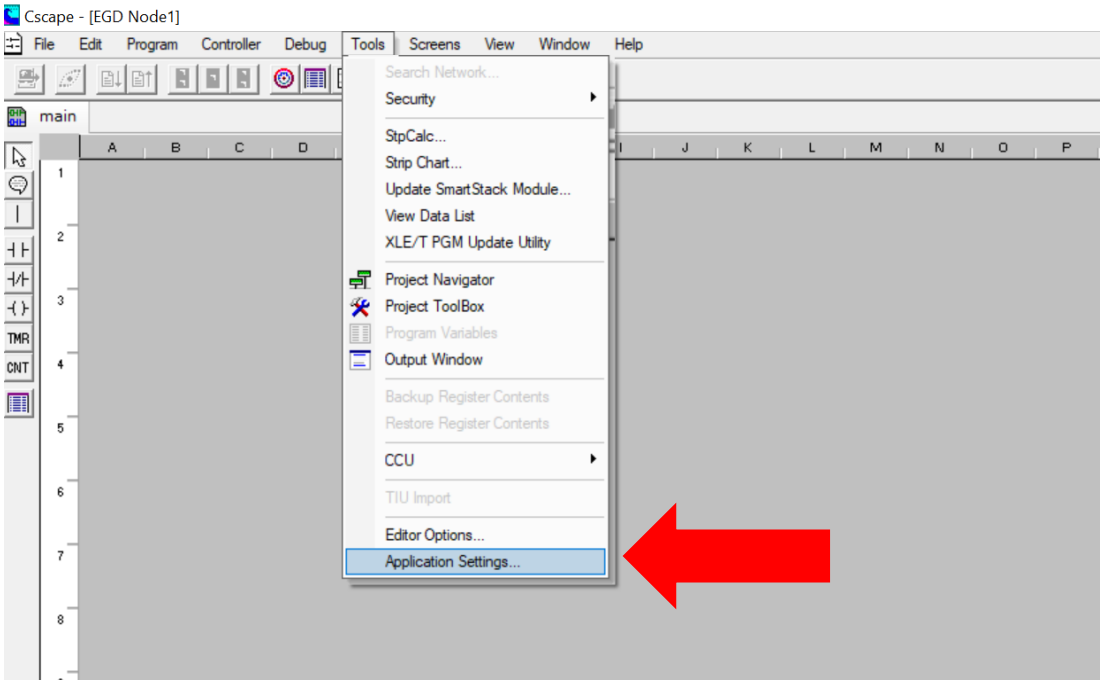
CsCAN over Ethernet protocol allows a CsCAN Host Programming Tool, such as Cscape, to access an OCS unit, as though it were connected directly to the OCS programming serial port.

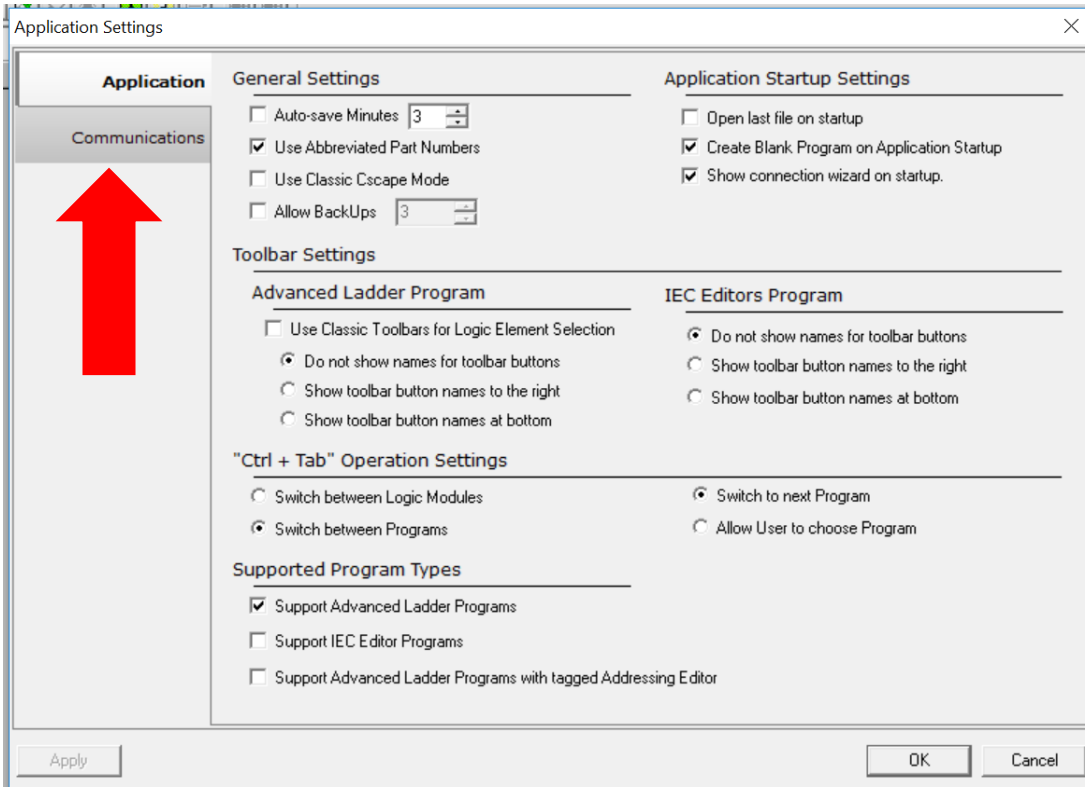
4.2 CsCAN over Ethernet Configuration

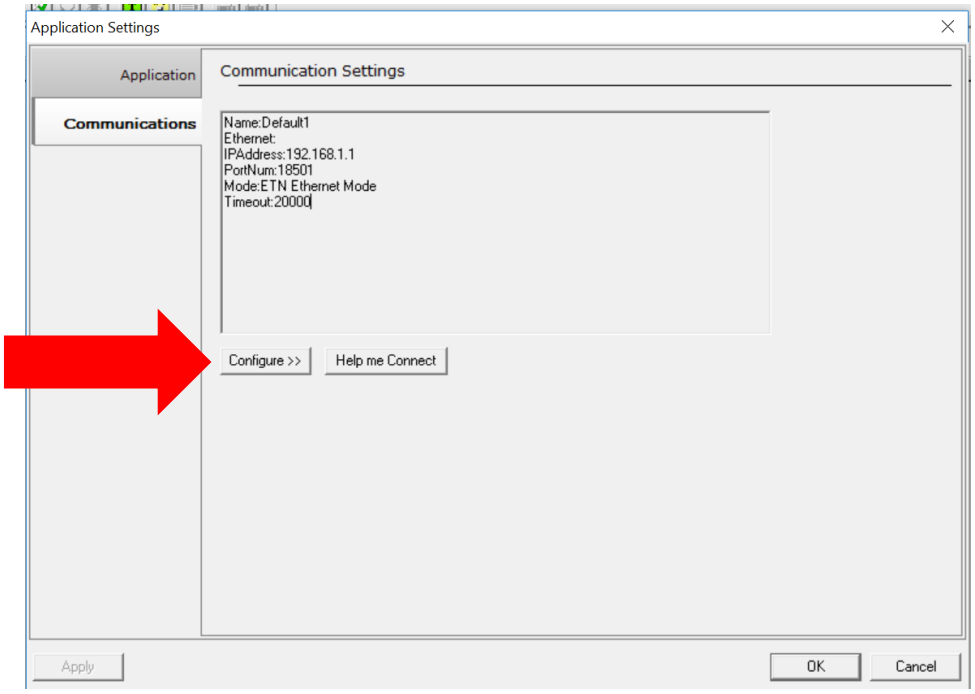
The OCS requires no protocol-specific configuration for CsCAN over Ethernet protocol. Only the general OCS configuration previously described in Chapter 3 is required. If the OCS has been assigned an IP Address and Net Mask, it will automatically respond to CsCAN over Ethernet messages sent to it by a PC running Cscape.

To connect using CsCAN over Ethernet protocol, perform the following these steps:

1. On the main Cscape screen, select the **Tools** menu → **Application Settings** → **Communications** and then click on **Configure>>** button to display Add Target dialog. See figures below.


Cscape - [EGD Node1]
File
Edit
Program
Controller
Debug
Tools
Screens
View
Window
Help
Search Network...
Security
StpCalc...
Strip Chart...
Update SmartStack Module...
View Data List
XLE/T PGM Update Utility
Project Navigator
Project ToolBox
Program Variables
Output Window
Backup Register Contents
Restore Register Contents
CCU
TIU Import
Editor Options...
Application Settings...
main
TMR
CNT


Application Settings
Application
Communications
General Settings
Auto-save Minutes
Use Abbreviated Part Numbers
Use Classic Cscape Mode
Allow BackUps
Application Startup Settings
Open last file on startup
Create Blank Program on Application Startup
Show connection wizard on startup.
Toolbar Settings
Advanced Ladder Program
Use Classic Toolbars for Logic Element Selection
Do not show names for toolbar buttons
Show toolbar button names to the right
Show toolbar button names at bottom
IEC Editors Program
Do not show names for toolbar buttons
Show toolbar button names to the right
Show toolbar button names at bottom
"Ctrl + Tab" Operation Settings
Switch between Logic Modules
Switch between Programs
Switch to next Program
Allow User to choose Program
Supported Program Types
Support Advanced Ladder Programs
Support IEC Editor Programs
Support Advanced Ladder Programs with tagged Addressing Editor
Apply
OK
Cancel



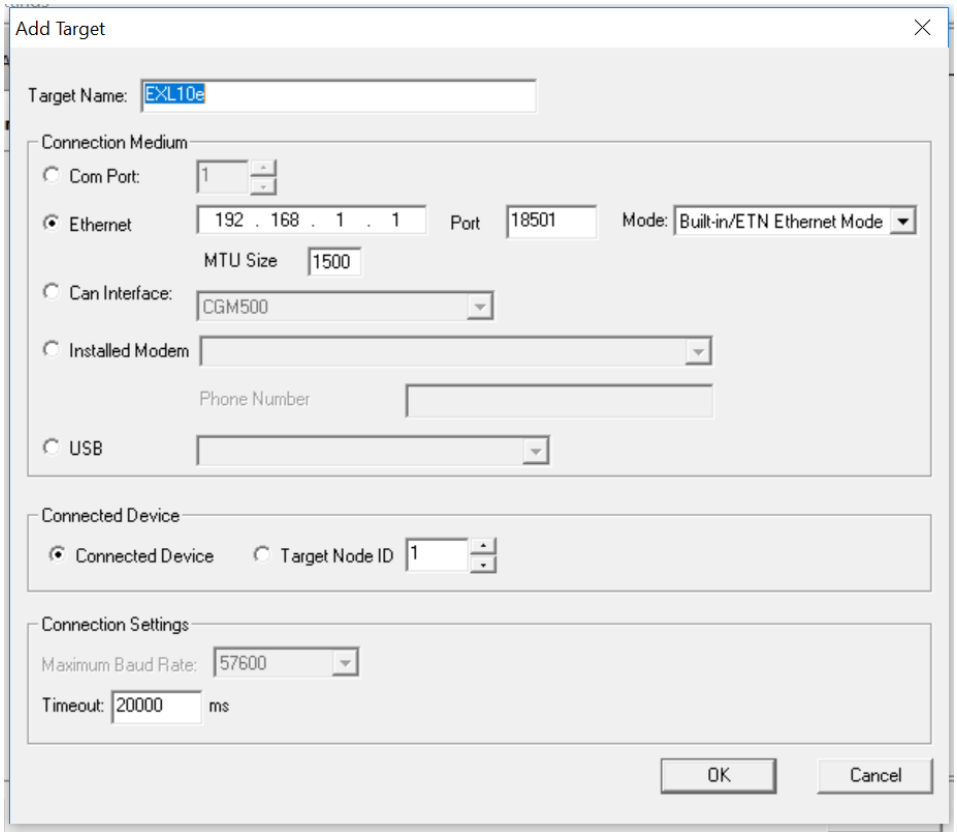


Figure 4.1 – Add Target Screen

2. Select the **Ethernet** from the **Connection Medium** list, and then set the **Target IP Address** and **Timeout** parameters as follows:

Target IP Address:
Enter the IP Address previously assigned to the target OCS. Please refer to Chapter 3 regarding how to assign an IP Address to an OCS.

MTU Size:
[Maximum Transmission Unit] Frequently networks are limited to MTU size of less than 1500 bytes. The MTU size should be set to match the network's MTU.

To check your network's MTU size, perform the following steps:

1. Click the Widows '**Start**' button.
2. In the search field type: '**cmd**' and press enter.
3. At the command prompt type: '**netsh interface ipv4 show subinterface**'

Timeout:
Enter a number between 1000 and 65,000 (in milliseconds) for the maximum expected network round-trip communication time. This value determines how long Cscape will wait for a response after it sends a CsCAN over Ethernet protocol command to the OCS.

NOTE: For most local network applications, the default **Timeout** value of 1000 is sufficient. However, there are some network considerations, which may require the **Timeout** value to be increased to facilitate reliable communication. This includes heavily loaded networks, complex networks with multiple levels of routers and switches, and Internet communication.

Also, because CsCAN communication is affected by ladder code scan rate, the **Timeout** value may have to be further increased to compensate for applications with very slow scan rates.

4.3 CsCAN over Ethernet Operation

As stated previously, the OCS allows a CsCAN Host Programming Tool, such as Cscape, to use CsCAN over Ethernet to perform all standard supervisory control, monitoring and programming functions with the OCS, as though it were connected directly to the OCS programming serial port.

These standard supervisory functions include the OCS's ability to handle pass-through communication with the CsCAN Nodes attached to the OCS CAN port. This feature is known as CsCAN single-point programming.

4.4 CsCAN over Ethernet Downloading Precautions

When downloading a new Ethernet configuration to a target OCS, using CsCAN over Ethernet protocol, extra care should be taken.

When downloading **I/O and Network Configuration**, the OCS configuration is also downloaded, which has the potential to change the target device's IP Address, or it could even remove the OCS configuration entirely, resulting in loss of communication and a failed download.

For this reason, when the Cscape user modifies the Hardware Configuration, and then starts to download it using CsCAN over Ethernet, Cscape issues a warning message, as shown in Figure 4.2.

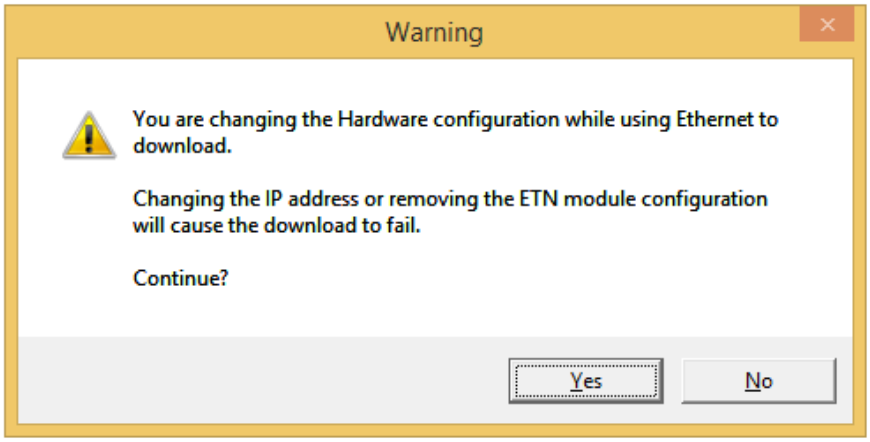
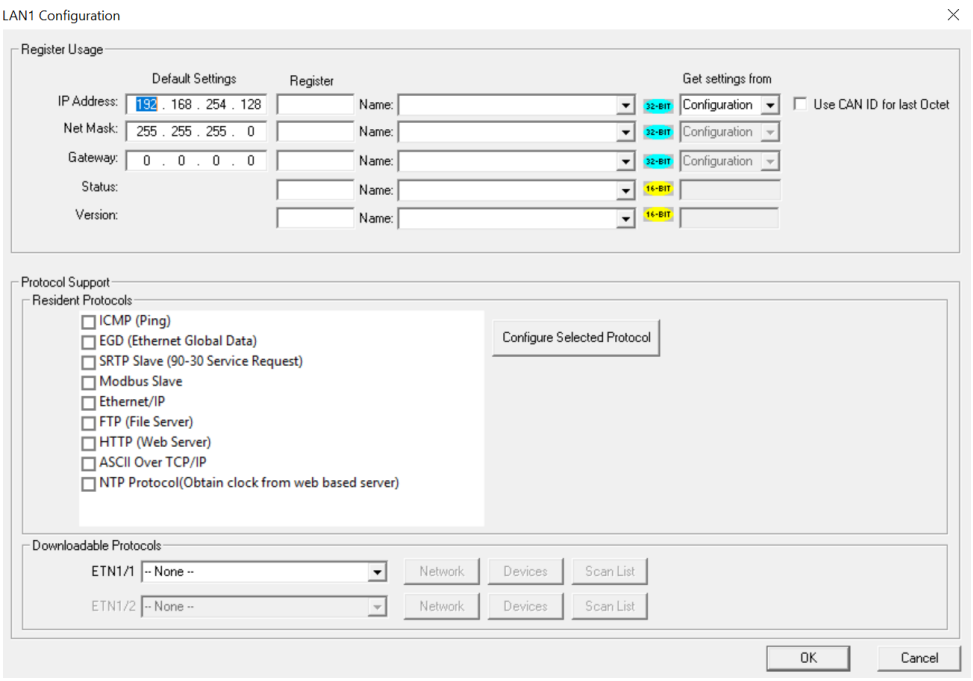


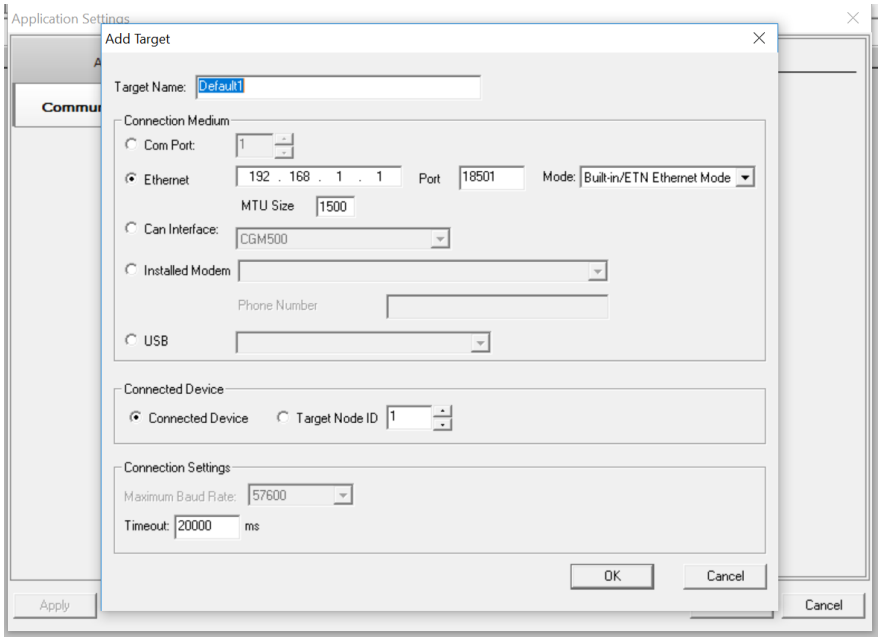
Figure 4.2 – I/O Configuration Download Warning

4.4.1 HOW TO PREVENT LOSING COMMUNICATION

Review Hardware Configuration, **Controller → Hardware Configuration → LAN1 / Config .**



Review Add Target: **Tools → Application Settings → Communications → Configure»**



Before using CsCAN over Ethernet to download a new **Hardware Configuration** to an OCS, the application programmer should:

1. Make sure the new **Hardware Configuration** contains an OCS configuration.
2. Make sure the new Ethernet configuration will not change the IP Address.
3. Refer to Chapter 3 on General Configuration and review all the configuration steps.

Table 4.1 - OCS IP Address	
Get Settings From: Configuration	In this mode, the OCS's IP Address comes from the **IP Address** parameter only, and does not use the CAN Network ID for the low octet and does not use **IP Address Register** to obtain an IP Address from an OCS register. In this case the OCS writes the static IP Address to the 32-bit OCS register indicated by the **IP Address Register** parameter
Static IP Address with CAN ID	In this mode, the OCS's IP Address comes from a combination of the **IP Address** parameter and the OCS/RCS CAN Network ID. The most significant (leftmost) three octets of the IP Address come from the **IP Address** parameter. The least significant (rightmost) octet of the IP Address is taken from the OCS (or RCS) CAN Network ID. In this case the OCS writes the *adjusted* IP Address to the 32-bit OCS register indicated by the **IP Address Register** parameter.
Get Settings From: Register	In this mode, the OCS's IP Address comes from an OCS register. The **IP Address Register** parameter indicates which 32-bit OCS registers to read the IP Address from. The static **IP Address** parameter is not used in this situation, except to set the **Default IP Address** in non-volatile memory.

- If the new Ethernet configuration specifies "**Get Settings From:**" make sure the **IP Address** parameter matches Cscape's **Target IP Address** (Figure 4.1).

- If the new Ethernet configuration specifies "**Get Settings From:**" CAN ID, make sure the IP Address, which will be built from the combination of the **IP Address** parameter and the OCS/RCS CAN Network ID, matches Cscape's **Target IP Address** (Figure 4.2).

- If the new Ethernet configuration specifies "**Get Settings From:**" OCS Register, make sure the OCS register indicated by the **IP Addr Register** parameter contains an IP Address, and that it matches Cscape's **Target IP Address** (Figure 4.2).

  NOTE: If necessary, use Cscape's Data Watch tool to set the OCS register to the correct IP Address before downloading. Refer to the Help file in Cscape for more information on Data Watch.

4.4.2 HOW TO RECOVER FROM LOST COMMUNICATION

If the OCS's IP Address changes because of using CsCAN over Ethernet to download a new user program, activity will halt at the end of the I/O Configuration download, and communication between Cscape and the target OCS will be lost. Then, after several seconds, Cscape will display a Communication Timeout error.

If this happens, it is possible to recover Cscape communication with the target OCS, as follows:

1. Referring to Chapter 3 (especially Section 3.2) and to Section 4.4.1 above, determine the OCS's new IP Address by using Cscape to re-examine the I/O Configuration just downloaded.
2. Referring to Figure 4.1 in Section 4.4.1, change Cscape's **Target IP Address** to match the OCS's new IP Address.
3. Try performing the download again.
4. If this fails, try changing Cscape's **Target IP Address** to match the static **IP Address** parameter (even if it is grayed out).
5. If all else fails, connect a PC running Cscape directly to the target OCS unit's programming serial port, repeat the download, and then use Data Watch to examine the register indicated by the **IP Addr Register** parameter in the Hardware Configuration dialog to discover the OCS's new IP Address.

NOTE: If the application programmer wants to use CsCAN over Ethernet to intentionally change an OCS's IP Address, do this by (1) starting the download, (2) waiting for Cscape to timeout and (3) performing steps 1, 2 and 3 of the recovery process, as described above.

4.5 CsCAN over Ethernet Security

To prevent the use of CsCAN over Ethernet protocol to gain unauthorized access to an OCS or RCC, Cscape Programming Software and OCS Ethernet Firmware can password-protect CsCAN over Ethernet communication. To implement CsCAN over Ethernet Security, use Cscape Programming Software, to perform the following nine steps:

1. Open the user program previously created for the target OCS or RCS controller.

2. **If the user program has already been set up with security passwords**, first log-in as the administrator. **Tools → Security →Log-in**, then enter the administrator password, and then click **OK**. Cscape will then acknowledge the administrator log-in; click **OK** again.

3. On the main Cscape screen, select the **Tools** menu and its **Security** sub-menu, and then click **Change Passwords**, then open the **Security Passwords** dialog, as shown in Figure 4.3.

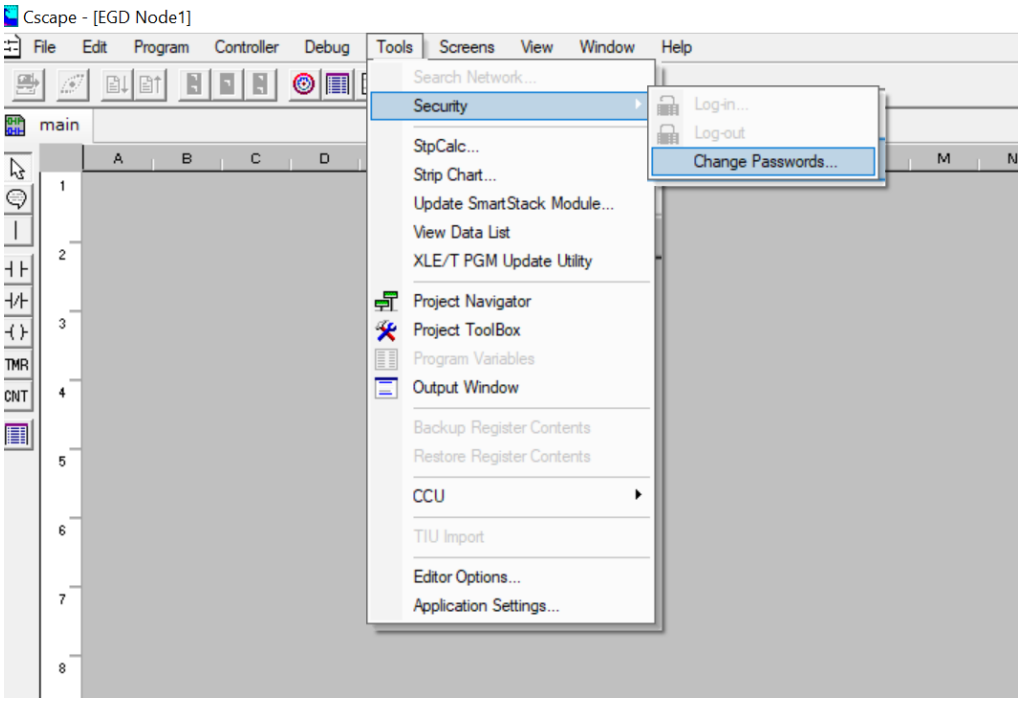


Figure 4.3 – Opening Change Passwords Dialog

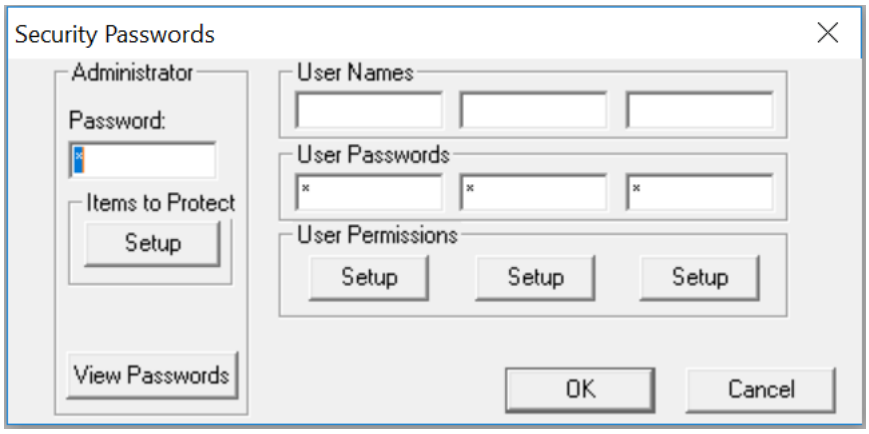


Figure 4.4 –Security Passwords Dialog

4. Click on the **View Passwords** button to view the Administrator and User passwords.

5. Change passwords and user names as desired for the application.

   NOTE: Passwords are numeric values between 1 and 999,999, while the user names can be any text from 1 to 15 characters long.

6. Click on the **Items to Protect Setup** button to open the Administrator's **Security Settings** dialog, as shown in Figure 4.5.

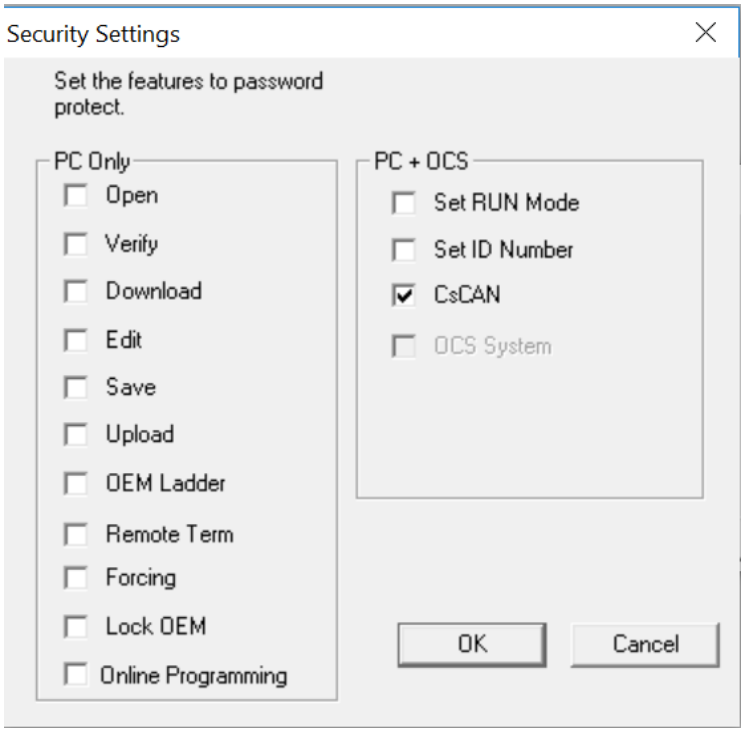


Figure 4.5 – Security Settings dialog

7. To password-protect CsCAN over Ethernet protocol, make sure the **CsCAN TCP (Ethernet)** checkbox is checked in the Administrator's **Security Settings** dialog, as shown in Figure 4.5. Then click **OK**.

8. The Administrator password is always authorized for all protected features. If desired, one or more of the user passwords can also be authorized for CsCAN over Ethernet communication. To do this, refer to Figure 4.4 and click one of the **User Permissions Setup** buttons to open that user's **Security Settings** dialog, which will be like Figure 4.5. Then check the **CsCAN TCP (Ethernet)** checkbox and click **OK**.

9. After downloading the user program to the target OCS or RCC, both Cscape and the OCS (or RCC) controller will enforce the new password security for CsCAN over Ethernet protocol.

CHAPTER 5: ETHERNET GLOBAL DATA PROTOCOL (EGD)

5.1 EGD Overview

Ethernet Global Data (EGD) protocol is a GE Fanuc Automation protocol, which is designed for simple, efficient data exchanges between peer devices on a network.

EGD protocol communicates using the UDP transport layer. Although this method of data transfer is very efficient, it has no specific way to detect and recover lost data packets. However, since all EGD data transfers are periodic, lost data packets will be repeated when their user-configured time periods expire.

CAUTION: EGD protocol is **not** intended for one-time event notification or for applications with critical data, which cannot withstand being delayed as described above.

Each device on an EGD network can be configured as a Producer, as a Consumer, or both.

A Producer is a device that transmits Exchanges (blocks of data) to one or more Consumers. A Producer can transmit Exchanges directly to a specific Consumer, by sending them to the Consumer's IP Address. (This is **Unicast** IP Addressing.) A Producer can also transmit Exchanges to a Group of Consumers, by sending them to a Group ID. (This is **Multicast** IP Addressing.) See Section 5.6 for more details regarding Unicast and Multicast IP Addressing.

A Consumer is a device that receives Exchanges from one or more Producers.

An OCS supports up to 127 concurrent Exchanges, each of which can be either a Producer or a Consumer of data.

5.2 EGD Terminology

Before configuring an OCS for EGD protocol, it is essential that the application programmer understand the key EGD terms, which are shown in Table 5.1.

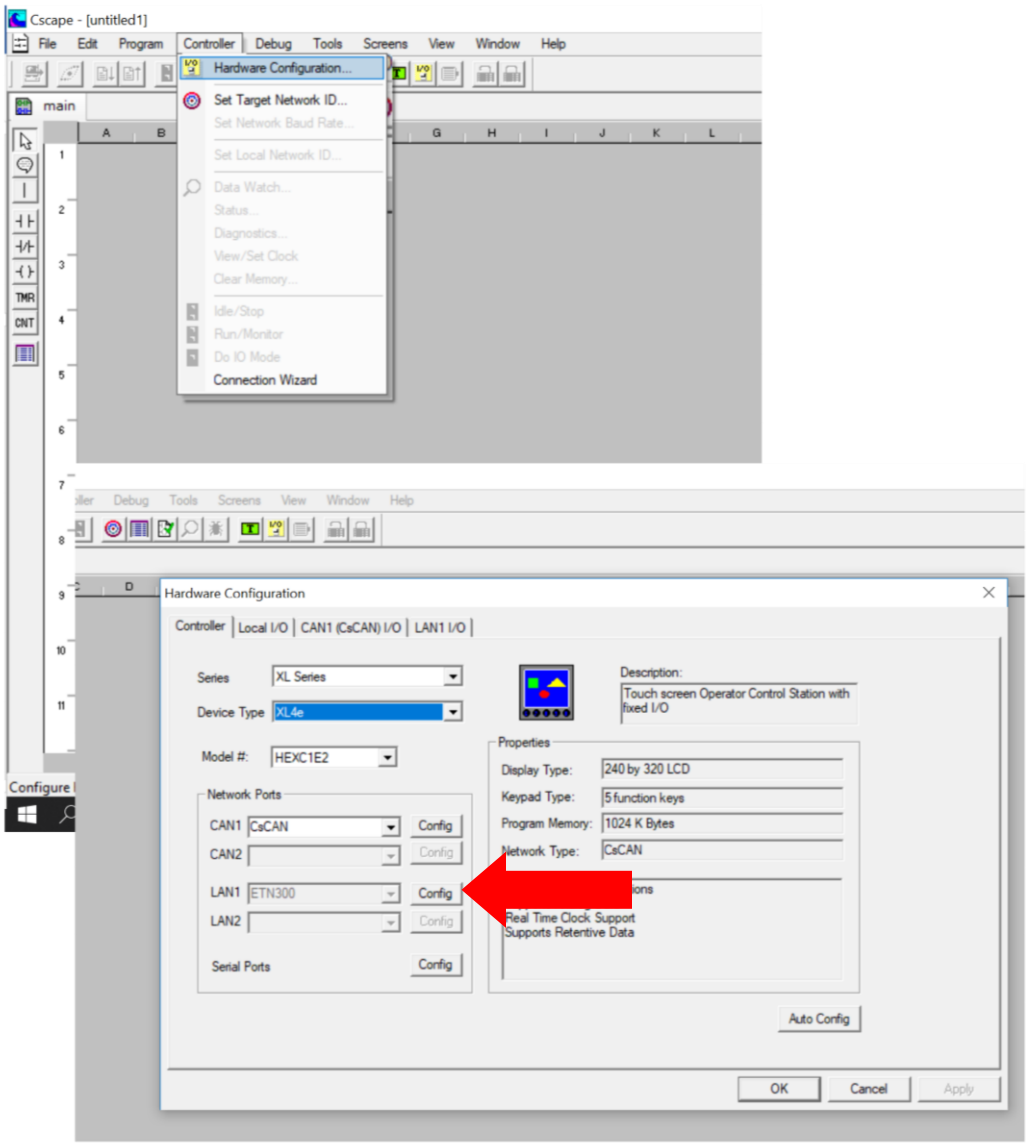
Table 5.1 – EGD Terminology

Term	Definition
Exchange	A block of data sent by a Producer and received by one or more Consumers
Exchange Number	A number (1 to 16,383), which along with the IP Address of the Producer, is used to uniquely identify an Exchange on an EGD network
Producer	An EGD network device configured to transmit one or more Exchanges
Consumer	An EGD network device configured to receive one or more Exchanges
Produced Exchange	A block of data that a Producer sends to a Consumer or to a Group of Consumers
Consumed Exchange	A block of data that a Consumer or Group of Consumers receives from a Producer
Group	One or more Consumers that are configured to receive Exchanges, which have been sent by a Producer to a specific Group ID
Group ID	A number (1 to 32), which is used to identify a Group of Consumers
Production Period	A value (in milliseconds) that specifies how often a Produced Exchange is transmitted to the network
Update Timeout	A value (in milliseconds) that specifies how long a Consumer will wait to receive an Exchange, before considering it late. **NOTE:** In general, a Consumed Exchange's Update Timeout is normally set to at least twice the corresponding Produced Exchange's Production Period, plus 10ms.
Unicast	Peer-to-peer communication in which data is sent to a SINGLE device at a specific IP Address.
Multicast	Peer-to-peer communication in which data is sent to a GROUP IP Address, and all devices in that group will receive the data.

5.3 EGD Configuration

If EGD protocol will be used in the application, EGD Configuration must be performed, in addition to the general OCS Configuration previously described in Chapter 3. To configure EGD protocol, use Cscape Programming Software to perform the following six steps:

1. Open the OCS Configuration dialog: **Controller → Hardware Configuration → Config (LAN1 or LAN2).**



2. Enable EGD by checking the **EGD (Ethernet Global Data)** checkbox in the LAN1 (or LAN2) Configuration dialog.

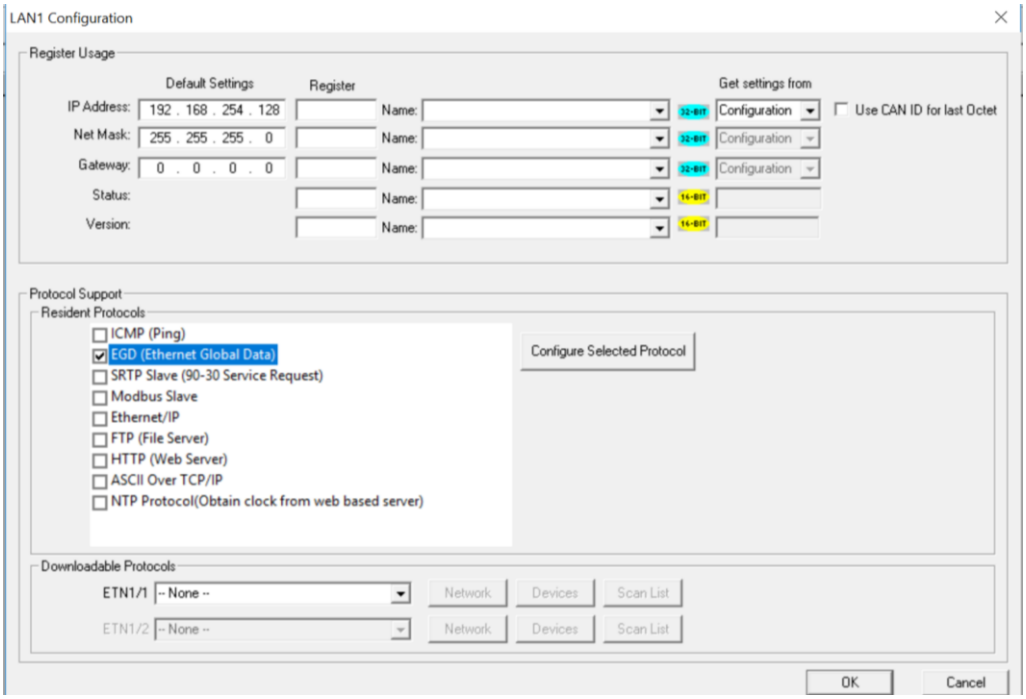


Figure 5.1 – Opening EGD Dialog

3. Click on the **Configure Selected Protocol** button after selecting **EGD (Ethernet Global Data)** to open the Ethernet Global Data dialog.

   NOTE: The Ethernet Global Data dialog allows EGD Produced and Consumed Exchanges to be configured. To configure Produced Exchanges, select the Produced Exchanges tab; to configure Consumed Exchanges, select the Consumed Exchanges tab. Refer to Figure 5.3 below.

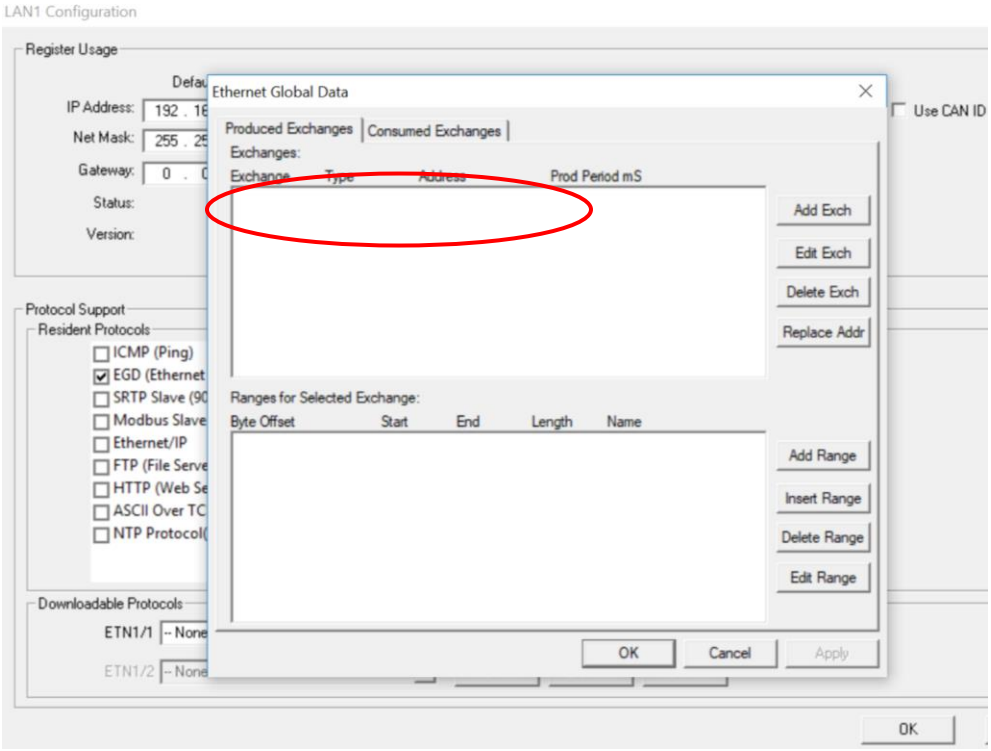


Figure 5.2– Ethernet Global Data (EGD) Configuration Dialog: Produced Exchange Tab Selected

4. Follow the steps in Section 5.4 to configure **Produced Exchanges**, as necessary for the application.

5. Follow the steps in Section 5.5 to configure **Consumed Exchanges**, as necessary for the application.

6. Click **OK** to accept the new EGD Configuration.

5.4 EGD Produced Exchange Configuration

Table 5.2 – Produced Exchange Unicast & Multicast	
Unicast	To configure a unicast communication, use the **IP address** of the other device.
Multicast	To configure a multicast communication, use a **Group Number**.

To configure EGD Produced Exchanges, open the Ethernet Global Data Configuration dialog (Figure 5.2) as described in Section 5.3, and select the **Produced Exchanges** tab, where:

1. In the upper window, one or more Produced Exchanges can be created.
2. In the lower window, I/O Blocks can be defined for each Produced Exchange.

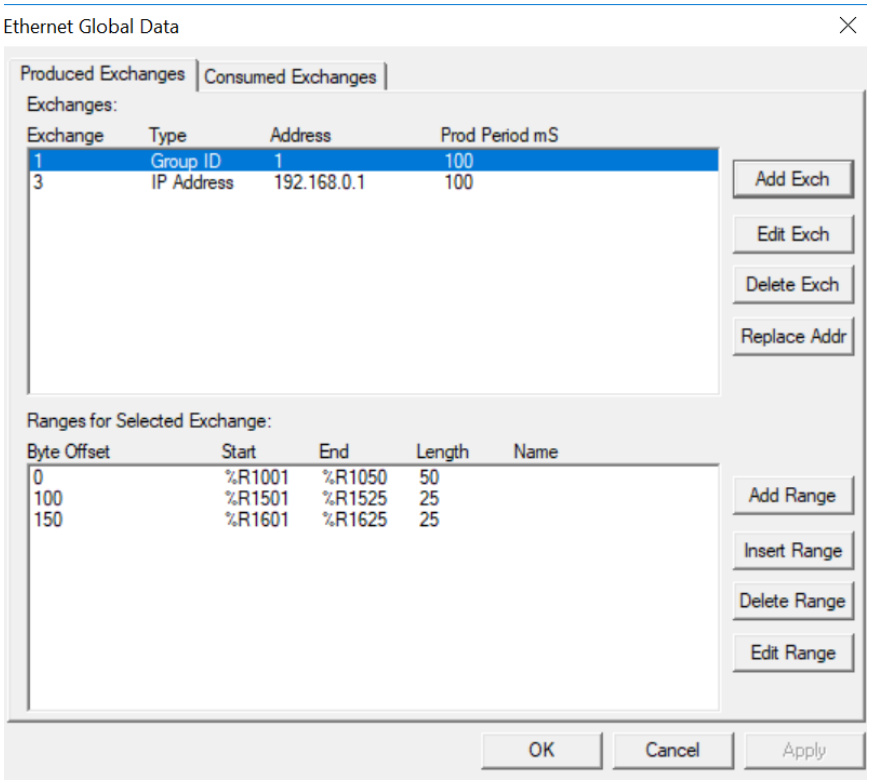


Figure 5.3 –Ethernet Global Data (EGD) dialog

When creating a Produced Exchange, the application programmer selects an Exchange Number for it, determines whether the Exchange will be sent to a specific Consumer or to a Group of Consumers, chooses which Consumer or Group of Consumers will receive the Exchange, and sets how often the Exchange will be sent to the EGD network.

When defining I/O Blocks for a Produced Exchange, the application programmer selects what type and how much *information* will be associated with the Exchange. For Produced Exchanges, there are two types of I/O Blocks to choose from: **Data** and **Status**. Refer to Table 5.2.

Table 5.3 – Produced Exchange I/O Block Type Definitions	
Type	**Definition**
Data Block	Block of consecutive OCS registers to be sent periodically to the EGD network
Status Block	16-bit OCS register to be written with the Produced Exchange's Status Word

When defining **Data Blocks** for a Produced Exchange, the maximum total OCS register data the Exchange can send to the EGD network is 1400 bytes. This means that up to a total of 700 16-bit registers (%R, %AI, AQ, etc.), or 11,200 1-bit registers (%M, %T, %I, %Q, etc.), or a combination thereof can be defined for a Produced Exchange.

NOTE: The OCS allows a total of up to 256 Data Blocks to be defined for all Produced Exchanges combined. This means that if 127 Produced Exchanges are configured, each can have an average of about 2 Data Blocks defined.

When a **Status Block** is defined for a Produced Exchange, exactly 2 bytes of register data are written with the Produced Exchange's Status Word. See Section 5.8 for general information regarding EGD Status Words, and Section 5.8.1 for specific information regarding EGD Produced Exchange Status Words.

NOTE: The OCS maintains just one 16-bit status word for each Produced Exchange. For this reason, there is never any need to define more than one Status Block for a given Produced Exchange.

5.4.1 CREATING EGD PRODUCED EXCHANGES: UNICAST & MULTICAST

To create **unicast** or **multicast** EGD Produced Exchanges, perform the following six steps:

1a. To configure a UNICAST communication: After opening the Ethernet Global Data Configuration dialog (Figure 5.2 in Section 5.3), click on the **Add Exch** button to open the Add / Edit Produced Exchange dialog (Figure 5.5). Select the destination Address and enter information.

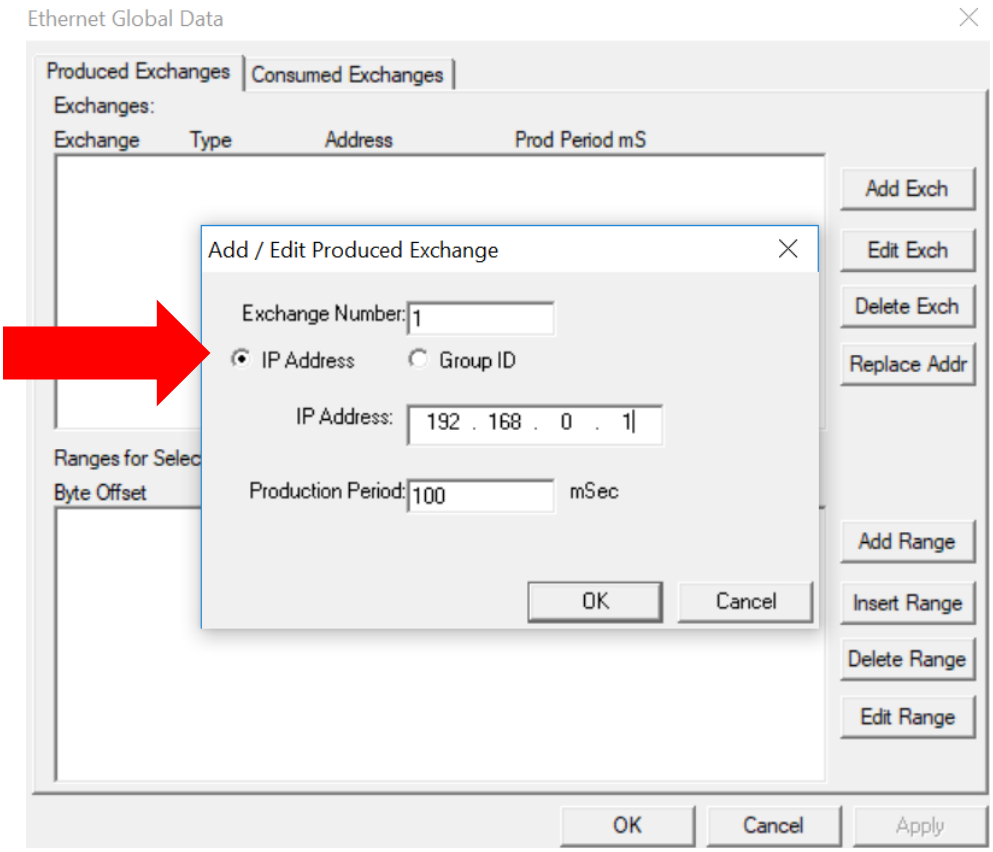


Figure 5.4 – Unicast Ethernet Global Data (EGD) dialog

1b. **To configure a MULTICAST communication**: After opening the Ethernet Global Data Configuration dialog (Figure 5.2 in Section 5.3), click on the **Add Exch** button to open the Add / Edit Produced Exchange dialog (Figure 5.5). Select and assign Group ID.

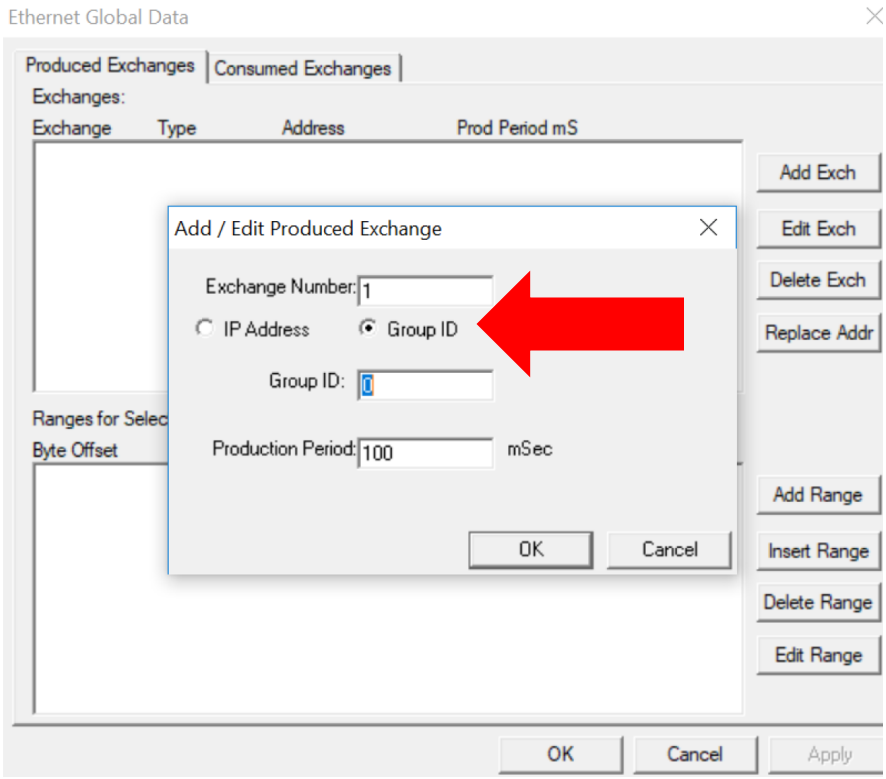


Figure 5.5 – Multicast Ethernet Global Data (EGD) dialog

2. Configure the Produced Exchange parameters as follows:

Table 5.4 - Produced Exchange Parameters	
Exchange Number	Enter a number between 1 and 16,383, which will be used to identify the Exchange to be sent.
IP Address Radio Button	Select this option if the Exchange will be sent to a specific Consumer. This is a **Unicast** exchange. (This will cause the next edit box to be for entering IP Address, instead of Group ID.)
Group ID Radio Button	Select this option if the Exchange will be sent to a Group of Consumers. This is a **Multicast** exchange. (This will cause the next edit box to be for entering Group ID, instead of IP Address.)
IP Address Edit Box	If the IP Address radio button was selected, enter the IP Address of the specific Consumer that will receive the Produced Exchange.
Group ID Edit Box	If the Group ID radio button was selected, enter the Group ID number (1 to 32) of the Group of Consumers that will receive the Produced Exchange.
Production Period	Enter a number (in milliseconds) for how often the Produced Exchange will be sent to the EGD network.

3. Click **OK** to accept the new Produced Exchange configuration, which will be now be displayed in the upper window of the Ethernet Global Data dialog (Figure 5.2).

4. To add another Produced Exchange to the list, click on the **Add Exch** button again.

5. To edit or delete a Produced Exchange, highlight it in the upper window of the Ethernet Global Data Configuration dialog (Figure 5.2 as described in Section 5.3), and then click on the **Edit Exch** button or the **Delete Exch** button.

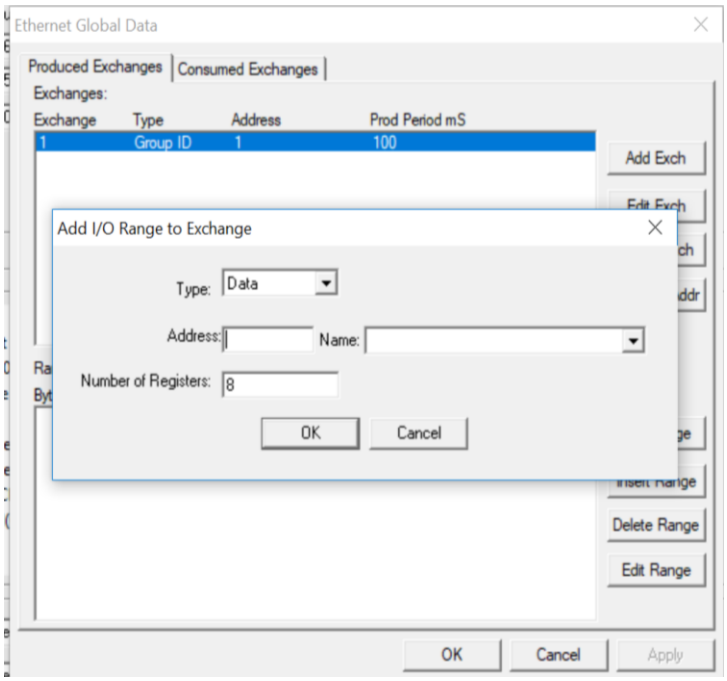
6. To quickly change the **IP Address** in multiple Produced Exchanges, click on the **Replace Addr** button. This will open a dialog to allow the user to search and replace the **IP Address** parameter, in all Produced Exchanges simultaneously.

5.4.2 DEFINING EGD PRODUCED EXCHANGE I/O BLOCKS

After creating a Produced Exchange (Section 5.4.1), one or more I/O Blocks should be defined for it. An I/O Block specifies what type and how much information will be associated with the Produced Exchange.

To define I/O Blocks for a Produced Exchange, perform the following six steps:

1. In the upper window of the Ethernet Global Data dialog (Figure 5.2 in Section 5.3), highlight one of the Produced Exchanges, and then click the **Add Range** button to open the Add I/O Range to Exchange dialog.



2. Configure the I/O Block parameters as follows:

Table 5.5 – I/O Block Parameters	
Type	Select Data Type to define a block of OCS registers, which the Produced Exchange will periodically read and send to the EGD network. Select Status Type to define a 16-bit OCS register, which will be written with the Produced Exchange's Status Word.
Address	Enter an OCS Register reference (such as %R1000), for the first OCS register, in a block of OCS registers, that will be periodically read and sent (Data Type), or for a 16-bit OCS register that will be written with status information (Status Type).
Number of Registers	If Data Type was selected, enter the number of registers to be periodically read and sent. If Status Type was selected, this edit box changes to **Number of Bytes**, and always has a fixed value of 2.

3. Click **OK** to accept the Produced Exchange's new I/O Block, which will be now be displayed in the lower window of the Ethernet Global Data Configuration dialog (Figure 6.1).

   NOTE: The OCS registers, specified in Data Block definitions, are sent to the EGD network in top-to-bottom order, as they appear in the lower window of the Ethernet Global Data Configuration dialog (Figure 5.2) in Section 5.3.

4. To add another I/O Block, click the **Add Range** button again and repeat steps 2 and 3. The new I/O Block will appear at the end of the list.

5. To insert an I/O Block into the middle of the list, highlight one of the I/O Block items in the list and then click on the **Insert Range** button. In this case, the new I/O Block will be inserted just before the highlighted I/O Block.

6. To edit or delete an I/O Block in the list, highlight it and then click on the **Edit Range** or **Delete Range** button.

5.5 EGD Consumed Exchange Configuration

To configure EGD Consumed Exchanges, open the Ethernet Global Data Configuration dialog (Figure 5.2) as described in Section 5.3, and select the **Consumed Exchanges** tab, where:

1. In the upper window, one or more Consumed Exchanges can be created (Section 5.5.1).
2. In the lower window, I/O Blocks can be defined for each Consumed Exchange (Section 5.5.2).

When creating a Consumed Exchange, the application programmer selects an Exchange Number for it, determines whether to receive the Exchange as a single Consumer or as a member of a Group of Consumers, chooses which Producer to receive the Exchange from, and sets how often to expect the Exchange to be received.

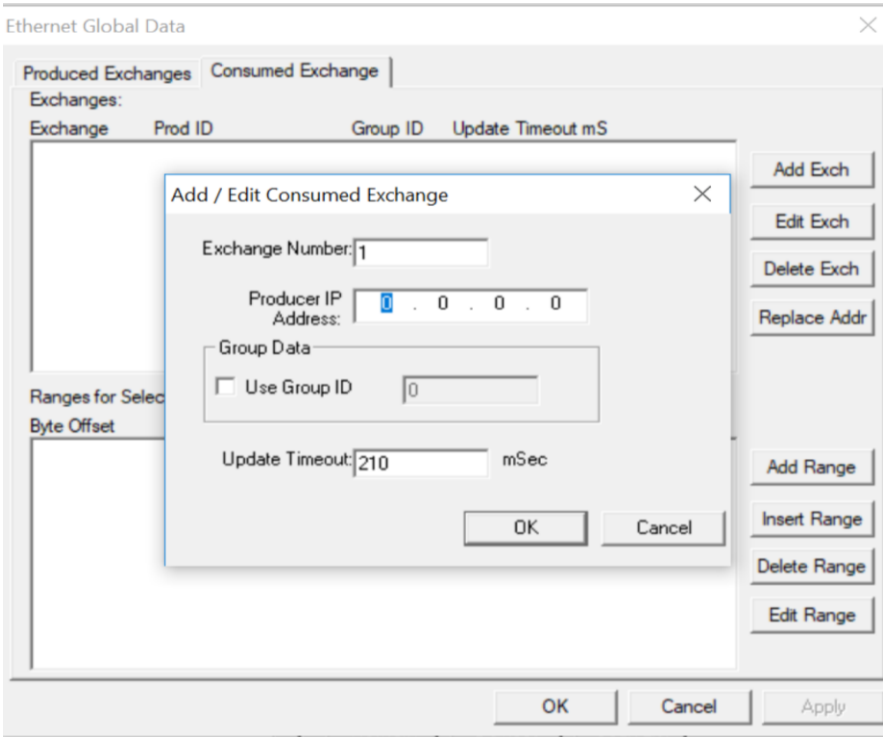
When defining I/O Blocks for a Consumed Exchange, the application programmer selects what type and how much *information* will be associated with the Exchange. For Consumed Exchanges, there are five types of I/O Blocks to choose from: Data, Status, Timestamp, OCS Timestamp, and Filler. Table 5.5 shows these I/O Block Types along with their definitions:

Table 5.6 – Consumed Exchange I/O Block Type Definitions	
Type	**Definition**
Data Block	Block of consecutive OCS registers to be written with received data. When defining **Data Blocks** for a Consumed Exchange, the maximum total OCS register data the Exchange can receive from the EGD network is 1400 bytes. This means that up to a total of 700 16-bit registers (%R, %AI, AQ, etc.), or 11,200 1-bit registers (%M, %T, %I, %Q, etc.), or a combination thereof can be defined for a Consumed Exchange. **NOTE:** The OCS allows a total of up to 256 Data Blocks to be defined for all Consumed Exchanges combined. This means that if 127 Consumed Exchanges are configured, each can have an average of about 2 Data Blocks defined.
Status Block	16-bit OCS register to be written with the Consumed Exchange's status word. When a **Status Block** is defined for a Consumed Exchange, exactly 2 bytes of register data are written with the Consumed Exchange's Status Word. See Section 5.8 for general information regarding EGD Status Words, and Section 5.8.2 for specific information regarding EGD Consumed Exchange Status Words.
Timestamp Block	Exactly 8 bytes of register data will be written with two 32-bit binary timestamp values, containing the numbers of seconds and nanoseconds since January 1, 1970. **NOTE:** The binary Timestamp Block is rarely used, and there is never any need to define more than one Status Block or OCS Timestamp Block for a given Consumed Exchange.
OCS Timestamp Block	Exactly 14 bytes of register data will be written with a 7-word OCS-format timestamp, consisting of second, minute, hour, day, month, year, and millisecond words. This OCS Timestamp can be displayed on the OCS screen, can be loaded into the OCS time-of-day clock, and can be more easily processed by ladder logic.
Filler Block	Specifies a block of received data to ignore (skips unwanted data).

5.5.1 CREATING EGD CONSUMED EXCHANGES

To create EGD Consumed Exchanges, perform the following six steps:

1. In the Ethernet Global Data Configuration dialog, click on the **Add Exch** button to open the Add / Edit Consumed Exchange dialog. Refer to image below.



2. Configure the Consumed Exchange parameters as follows:

Table 5.7 – Consumed Exchange Parameters	
Exchange Number	Enter a number between 1 and 16,383, which will be used to identify the Exchange to be received.
Producer IP Address:	Enter the IP Address of the Producer to receive the Exchange from.
Use Group ID	Check this box to receive an Exchange sent by a Producer to a Group ID.
Group ID	If the Use Group ID box is checked, enter a number from 1 to 32 in the edit box, to identify which of the 32 possible Groups of Consumers to become a member of.
Update Timeout	Enter a number (in milliseconds) for how long to wait to receive the Exchange, before considering it to be late. Typically, this should be set to twice the corresponding Production Timeout, plus 10 milliseconds.

3. Click **OK** to accept the new Consumed Exchange configuration, which will be now be displayed in the upper window of the Ethernet Global Data Configuration dialog (Figure 5.2) as described in Section 5.3.

4. To add another Consumed Exchange to the list, click on the **Add Exch** button again.

5. To edit or delete a Consumed Exchange, highlight it in the upper window of the Ethernet Global Data Configuration dialog (Figure 5.2), and then click on the **Edit Exch** or **Delete Exch** button.

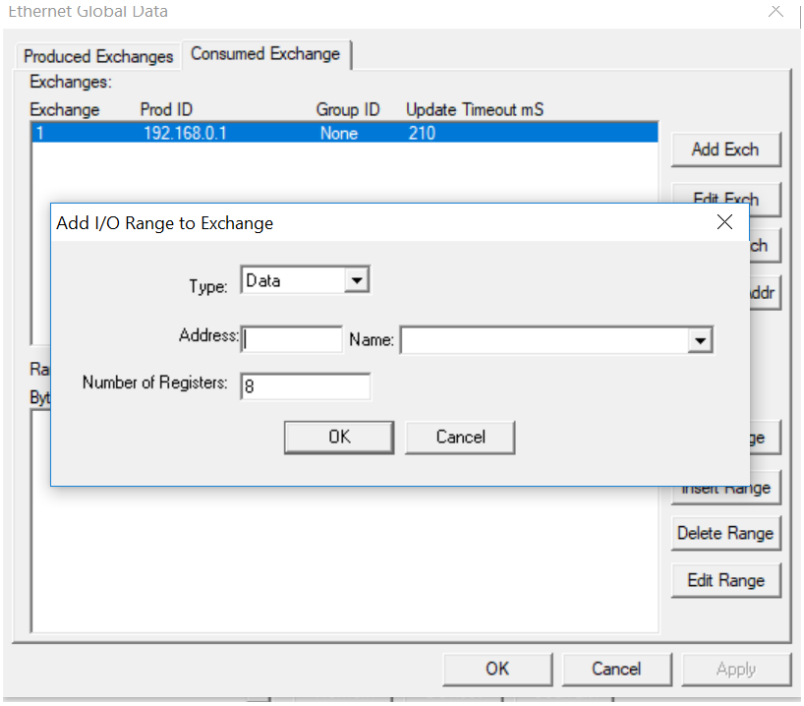
6. To quickly change the **Producer IP Address** in multiple Consumed Exchanges, click on the **Replace Addr** button. This will open a dialog to allow the user to search and replace the **Producer IP Address** parameter, in all Consumed Exchanges simultaneously.

5.5.2 DEFINING EGD CONSUMED EXCHANGE I/O BLOCKS

After creating a Consumed Exchange (Section 5.5.1), one or more I/O Blocks should be defined for it. An I/O Block specifies what type and how much information will be associated with the Consumed Exchange.

To define I/O Blocks for a Consumed Exchange, perform the following six steps:

1. In the upper window of the Ethernet Global Data Configuration dialog, highlight one of the Consumed Exchanges, and then click the **Add Range** button to open the **Add I/O Range to Exchange** dialog.



2. Configure the I/O Block parameters as follows:

Table 5.8– I/O Block Parameters	
Type	Select Data Type to define a block of OCS registers, which the Consumed Exchange will write with received data. Select Status Type to define a 16-bit OCS register, which will be written with the Consumed Exchange's Status Word. Select OCS Timestamp Type to define a 14-byte OCS register block to write with the received data's OCS-format timestamp. Select Filler Type, to skip unwanted received data.
Address	Enter an OCS Register reference (such as %R2000), for the first OCS register, in a block of OCS registers, that will be written with received data (Data Type), or for a 16-bit OCS register that will be written with status information (Status Type), or for the first OCS register in a block of OCS registers that will be written with the 14-byte timestamp (OCS Timestamp Type). For Filler Type, the Address edit box is not used.
Number of Registers	If Data Type was selected, enter the number of registers to be written with received data. Otherwise, this edit box changes to **Number of Bytes**, and has a fixed value of 2 for Status Type or 14 for OCS Timestamp Type. For Filler Type, enter the number of unwanted received data bytes to skip.

3. Click **OK** to accept the Consumed Exchange's new I/O Block, which will be now be displayed in the lower window of the Ethernet Global Data dialog (Figure 5.2) as described in Section 5.3 .

   NOTE: The OCS registers, specified in Data Block definitions, are filled with received data in top-to-bottom order, as they appear in the lower window of the Ethernet Global Data dialog. In this respect, Filler Blocks are placeholders for Data Blocks and can be thought of as Data Blocks whose received data bytes are discarded.

4. To add another I/O Block, click on the **Add Range** button again and repeat steps 2 and 3. The new I/O Block will appear at the end of the list.

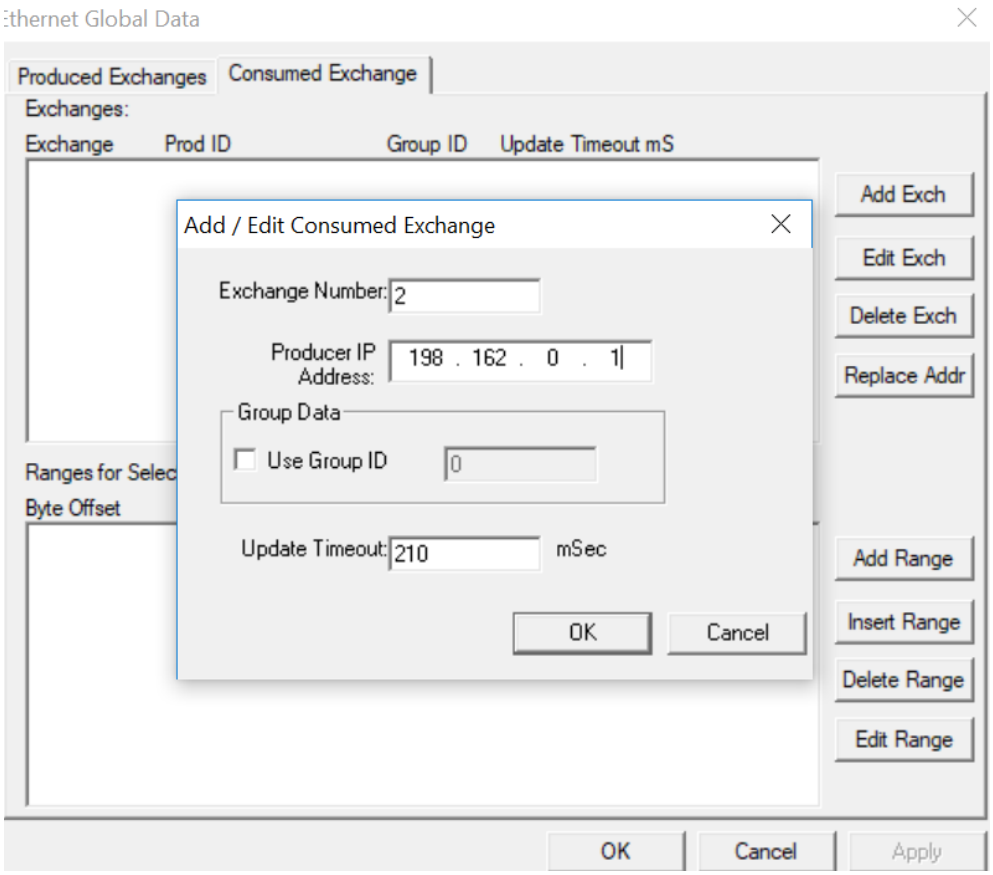
5. To insert an I/O Block into the middle of the list, highlight one of the I/O Block items in the list and then click on the **Insert Range** button. In this case, the new I/O Block will be inserted just before the highlighted I/O Block.

6. To edit or delete an I/O Block in the list, highlight it and then click on the **Edit Range** or **Delete Range** button.

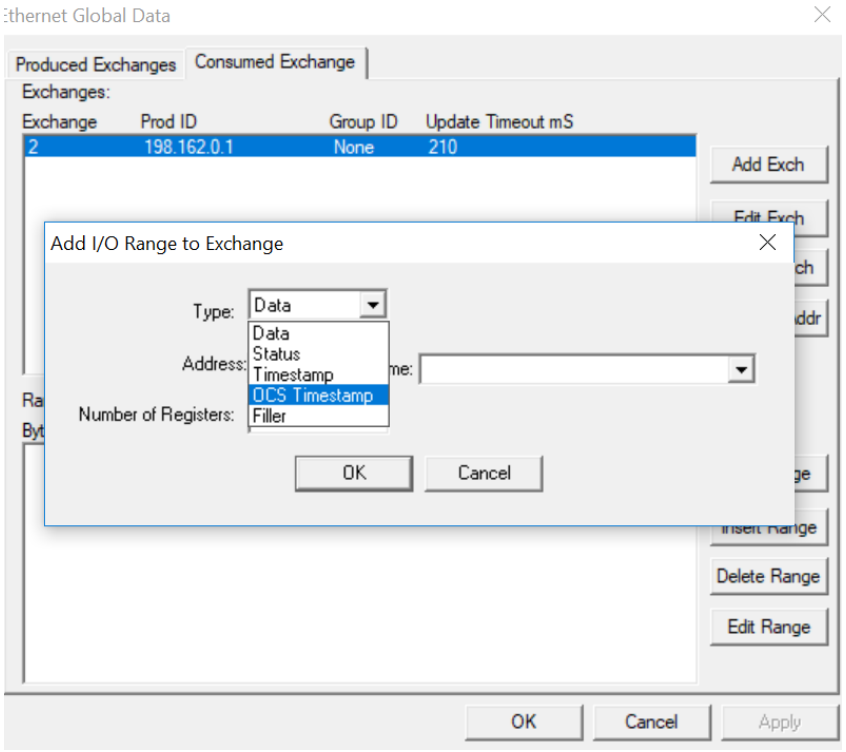
5.5.3 ADDING AN OCS TIMESTAMP BLOCK CONFIGURATION

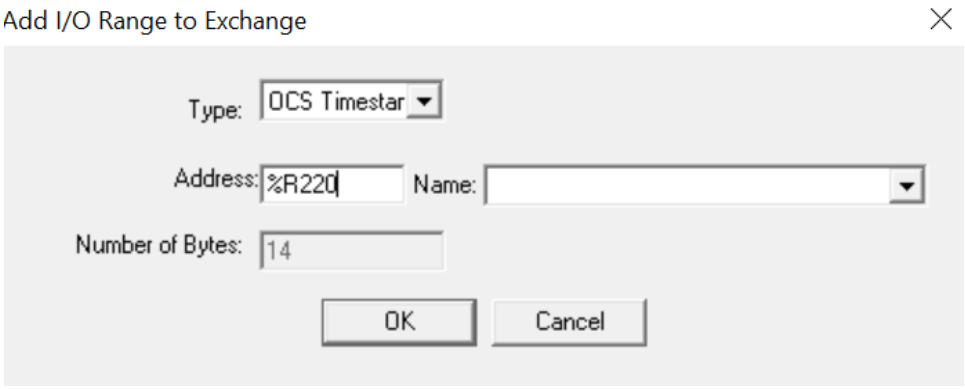
In EGD protocol, a timestamp is sent with every Produced Exchange, indicating when the Producer sampled the data being sent. For a Consumer of the Exchange to get this information into an OCS register, define an OCS Timestamp Block for the Consumed Exchange. Perform the following 2 steps:

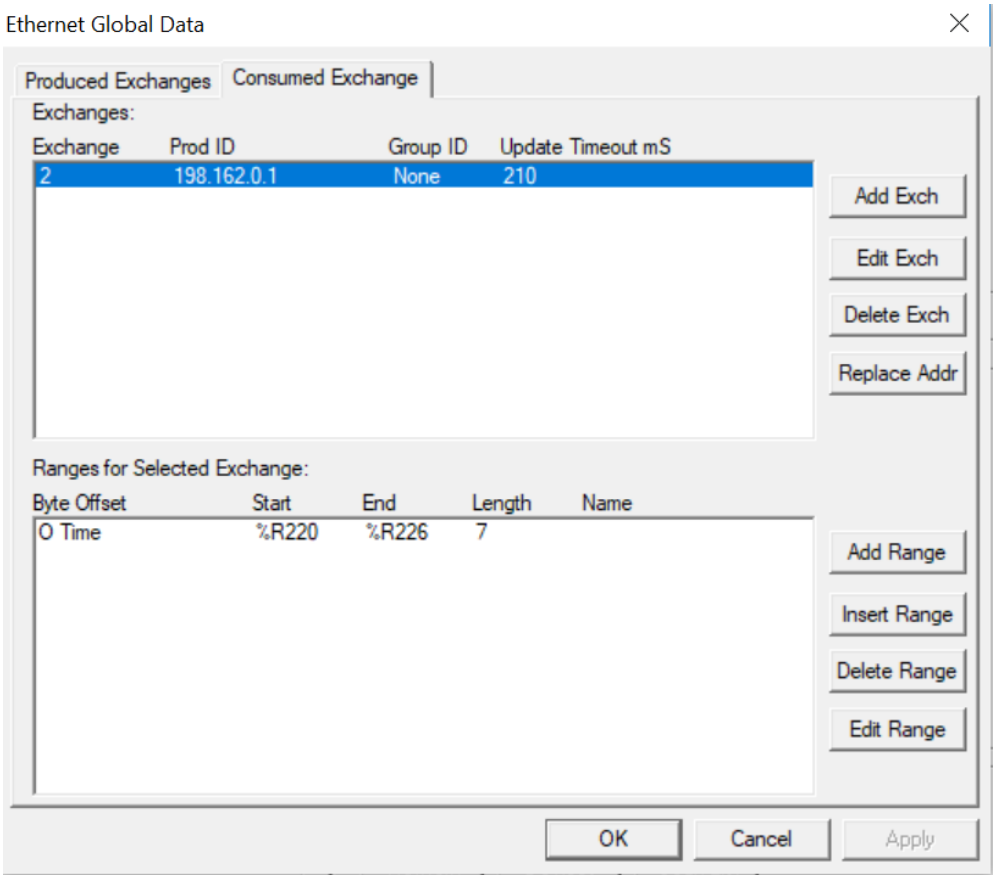
1. In the EGD Configuration dialog, select the **Consumed Exchanges** tab.



2. Make sure an exchange is highlighted in the upper and click on the **Add Range** button, to define an OCS Timestamp Block for Consumed Exchange. Then fill in the parameters, as shown in below, and click **OK**.






Ethernet Global Data
Produced Exchanges
Consumed Exchange
Exchanges:
Exchange
Prod ID
Group ID
Update Timeout mS
2
198.162.0.1
None
210
Add Exch
Edit Exch
Delete Exch
Replace Addr
Ranges for Selected Exchange:
Byte Offset
Start
End
Length
Name
0 Time
%R220
%R226
7
Add Range
Insert Range
Delete Range
Edit Range
OK
Cancel
Apply

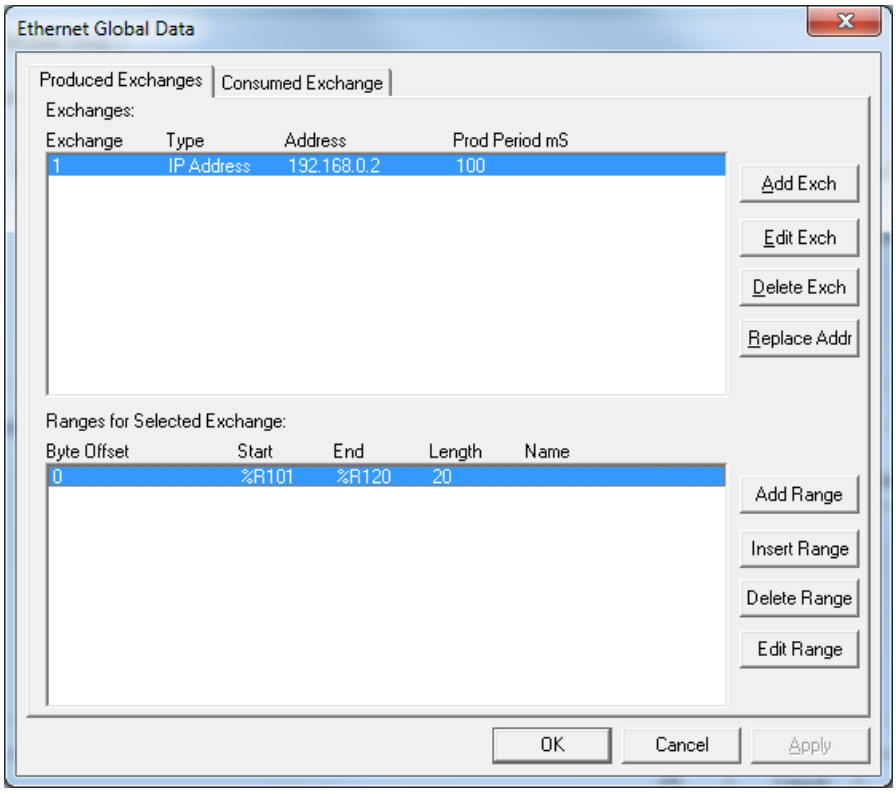
5.5.4 FILLER BLOCK CONFIGURATION

Sometimes a Consumer does not need all of the data sent by a Producer in an Exchange. In this case, the Consumed Exchange's Data Block should be defined to receive fewer registers than will be sent by the Producer of the Exchange.

However, if the partial data needed by the Consumer is not at the **beginning** of the data received in an Exchange, there must be some way to skip (ignore) the extra data. For this reason, the Ethernet Module supports Filler Blocks for Consumed Exchanges.

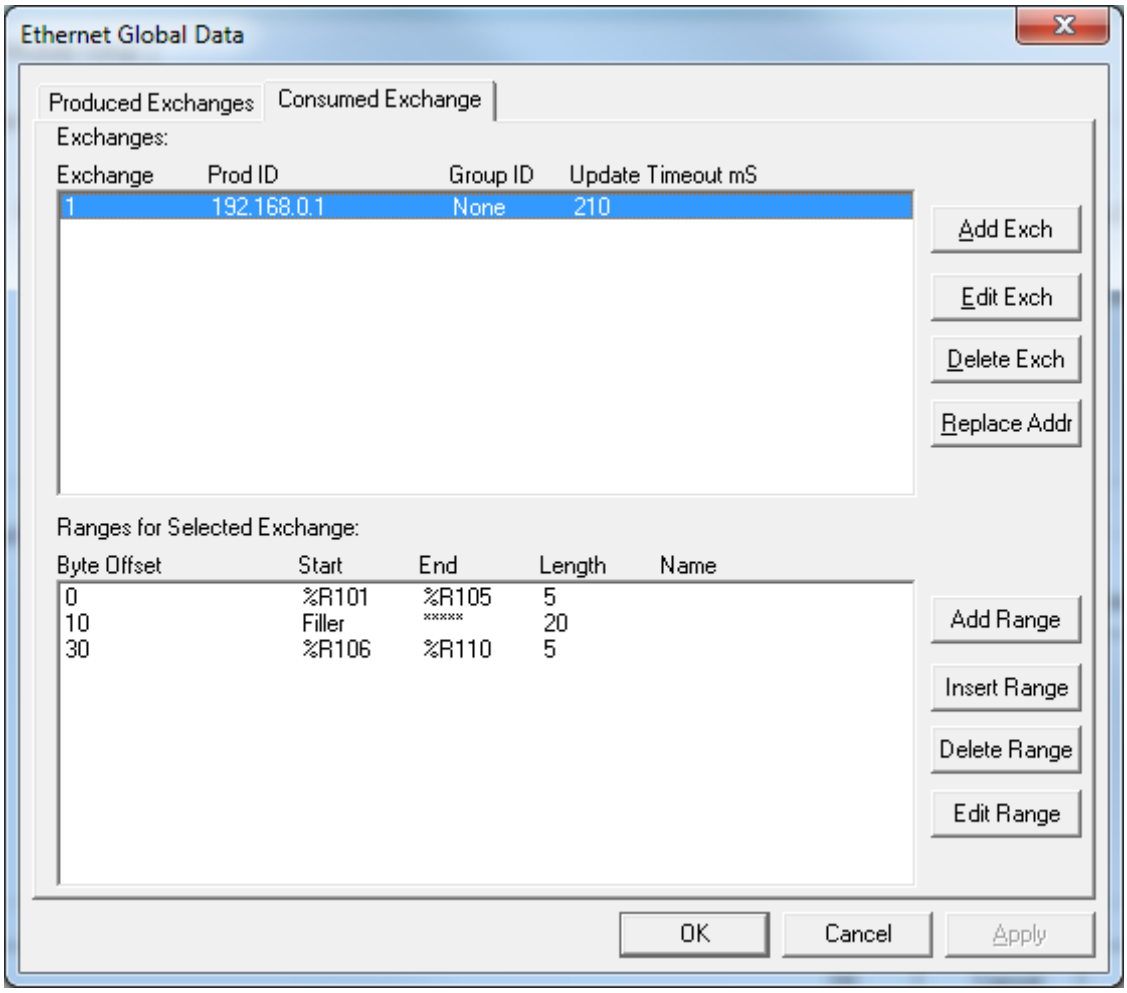
A **Filler Block** is used to skip unwanted data sent by the Producer. For example, if the Producer sends 40 data bytes in an Exchange, and the Consumer only needs the first and last 10 bytes, a Filler Block would be defined, in the appropriate slot in the list, to skip the middle 20 bytes of data.

Filler Produced Exchange Example: Controller A (IP 192.168.0.1)



Filler Consumed Exchange Example: Controller B (IP 192.168.0.2)

NOTE: The filler is used to skip the middle 20 bytes of Exchange 1.



5.6 EGD Unicast and Multicast IP Addressing

When using Ethernet Global Data (EGD) protocol for peer-to-peer communication, there are two methods for sending data: (1) send to a single device or (2) send to a group of devices. Refer to Section 5.1.1.

Unicast

When sending to a single device (method 1), EGD protocol uses **Unicast IP Addressing**. This means that the IP header's 32-bit Destination IP Address will contain the intended recipient's unique IP Address.

Multicast

When sending to a group of devices (method 2), EGD protocol uses **Multicast IP Addressing**.

This means that the IP header's 32-bit Destination IP Address will contain one of the 32 Multicast IP Addresses shown in Table 5.8.

Table 5.9 – EGD Multicast IP Addressing

Group ID	Multicast IP Address
1	224.0.7.1
2	224.0.7.2
:	:
32	224.0.7.32

Ethernet Switches normally do not support Multicast IP Addressing, while Ethernet Hubs do support Multicast IP Addressing. Some Ethernet Routers, known as Multicast Routers, do support Multicast IP Addressing, by using Internet Group Management Protocol (IGMP).

NOTE: For those customers wanting to use Multicast Routers to connect EGD devices, the OCS automatically handles IGMP communication with Multicast Routers.

5.7 EGD Status Words

EGD Status Words allow an Ethernet Global Data user to obtain the operating status of each EGD Exchange. The set of EGD Status Word values, implemented in the OCS, is a subset of the ones used in GE Fanuc EGD devices. This is because the OCS does not support dynamically defined (at run time) EGD Exchanges.

NOTE: Both Produced and Consumed Exchange Status Words are written to local OCS registers. This is the only case where a Produced Exchange can be configured to write to a local register.

The Status Word for a Produced Exchange is updated each time the Exchange's Production Period expires. The Status Word for a Consumed Exchange is updated when new data arrives for consumption or when the Exchange's Update Timeout expires.

In normal operation, each EGD Exchange's Status Word will always be 1 (OK), implying that new data was successfully Produced or Consumed. If the application needs to be notified when a data transfer has occurred on a given Exchange, the ladder program should clear the Exchange's Status Word register to 0 (IDLE) each time a non-zero event is detected.

5.7.1 EGD PRODUCED EXCHANGE STATUS WORDS

The Status Word for an EGD Produced Exchange can take on the following values:

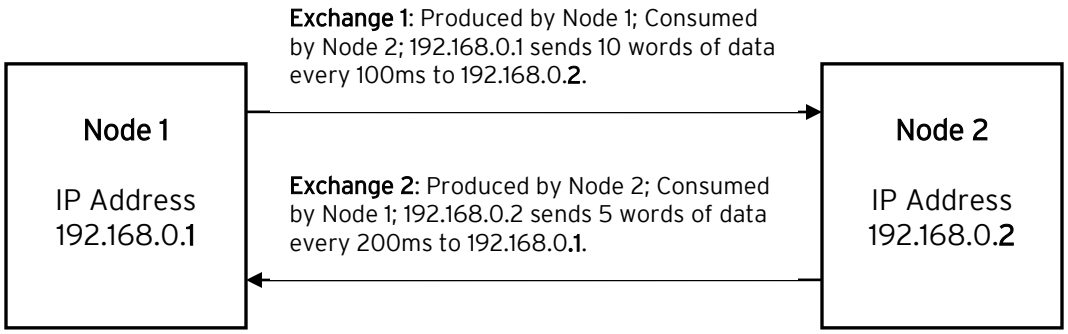
Table 5.10 – Status Word for EGD Produced Exchange		
0	IDLE	No new status event has occurred. The OCS initializes all Status Words to 0, only at power-up and each time the OCS enters RUN mode. Subsequently, the OCS application ladder program can write the value 0 to the Status Word, as an aid in knowing when Ethernet Global Data production occurs (see Status Word 1).
1	OK	Data Produced. New EGD network data has been transmitted. A transition to 1 indicates production of data occurred since the last OCS ladder scan.
2	NO LINK	The Ethernet link was down when EGD data production was attempted. This will occur, for example, if the Ethernet cable is unplugged from the OCS.

5.7.2 EGD CONSUMED EXCHANGE STATUS WORDS

The Status Word for an EGD Consumed Exchange can take on the following values:

Table 5.11 – Status Word for EGD Consumed Exchange		
0	IDLE	No new status event has occurred. The OCS initializes all Status Words to 0, only at power-up and each time the OCS enters RUN mode. Subsequently, the OCS application ladder program can write the value 0 to the Status Word, as an aid in knowing when Ethernet Global Data consumption occurs (see Status Words 1 and 7).
1	OK	Data Consumed. New EGD network data has been received as expected (before the configured Update Timeout expired). A transition to 1 (or to 7) indicates consumption of data occurred since the last OCS ladder scan.
2	NO LINK	The Ethernet link was down when EGD data consumption was expected. This will occur, for example, if the Ethernet cable is unplugged from the OCS.
3	No SYNC	**SNTP Error –** The Ethernet Interface in the device producing the exchange is configured for network time synchronization (Network Time Sync parameter is set to SNTP) but is not synchronized to an SNTP server. Therefore, the timestamp associated with this data is not synchronized to the network. **NOTE:** This error condition is considered the least important of all the error codes in this section. If another error condition exists, its status code will appear in the Status Word.
6	OVERDUE	**Data Refresh Error –** The Update Timeout has expired without receiving the expected data. Some possible causes for this error are: (1) the Producer has stopped producing the data, (2) the Consumed Exchange's configured Producer IP Address and/or Group ID do not exactly match those of the corresponding Produced Exchange, (3) or the Consumed Exchange's configured Update Timeout is too short (it should normally be twice the producer's Production Period, plus 10ms).
7	TARDY	**Data Consumed Late –** New EGD network data has been received, but it is later than expected (after the configured Update Timeout expired). A transition to 7 (or to 1) indicates consumption of data occurred since the last OCS ladder scan. The most probable cause for this error is that the Consumed Exchange's configured Update Timeout is too short (it should normally be twice the producer's Production Period, plus 10ms).

5.8 EGD Example 1: UNICAST EXCHANGE



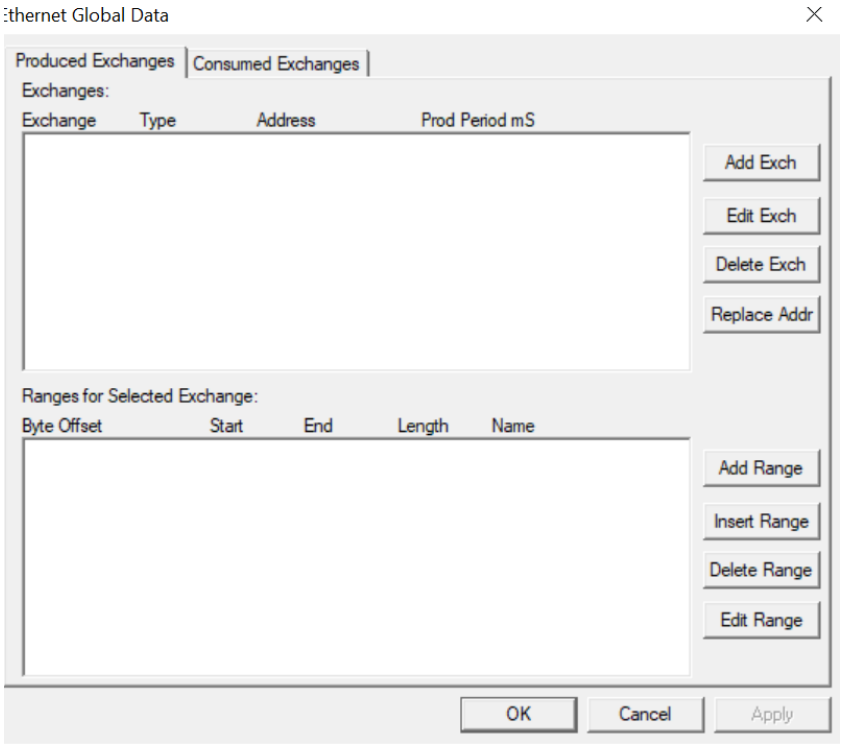
5.8.1 EGD EXAMPLE 1 – CONFIGURATION OF NODE 1

To configure Node 1 for EGD Example 1, as shown above, perform the following six steps:

1. To configure EGD Exchanges, open the Ethernet Global Data Configuration dialog (Figure 5.2) as described in Section 5.3.

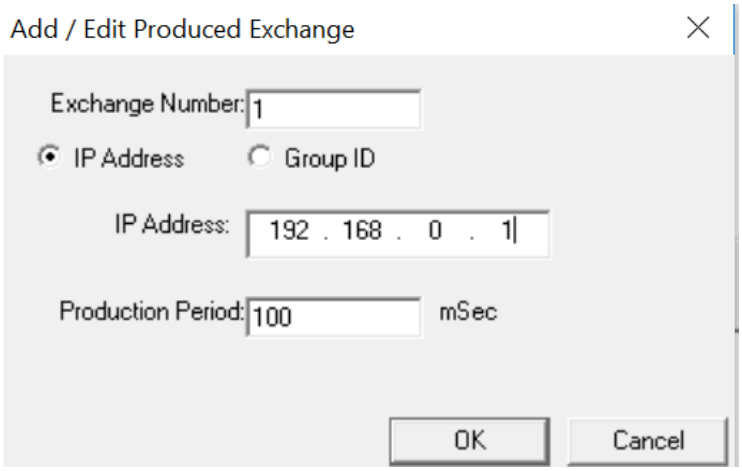
In this example Node 1 will have a Static IP address (refer to Section 3.2), and the %R1, %R2, and %R3 registers will be used to report the Ethernet Module's status, firmware version, and IP address.

NOTE: Since an IP Address is 32-bits long, Node 1's IP Address (192.168.0.1) will actually be written into %R3 and %R4.



Ethernet Module Configuration - Node 1

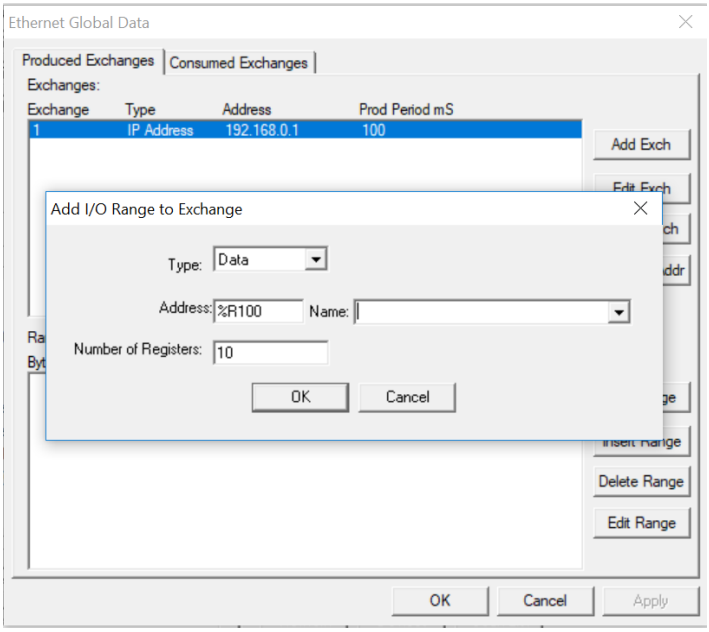
2. Click on **LAN1** next to the EGD (Ethernet Global Data) checkbox to open the Ethernet Global Data dialog. Click on **Add Exch** in the Produced Exchanges dialog.

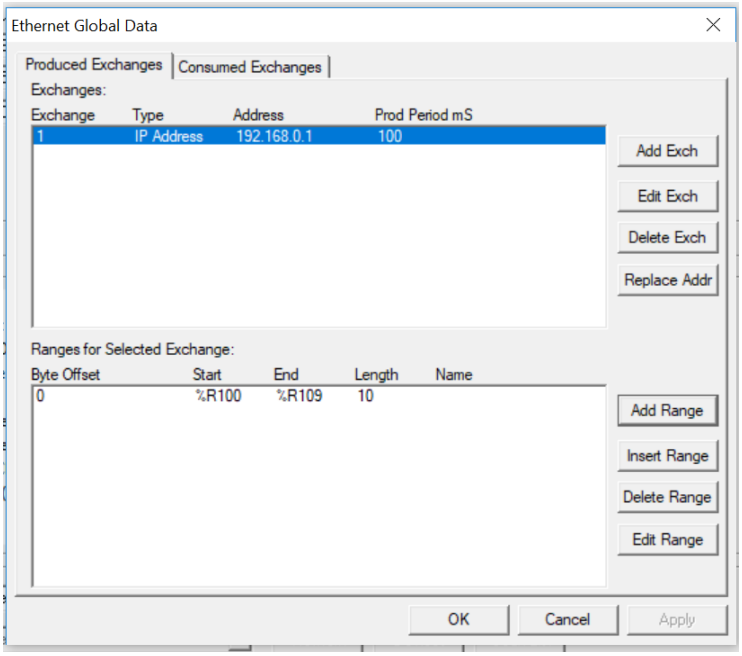


Creating Produced Exchange – Node 1

3. Click on Add Range button, in the EGD Configuration dialog, to define a Data Block for Produced Exchange 1. Then fill in the parameters as shown below. Then click **OK**.

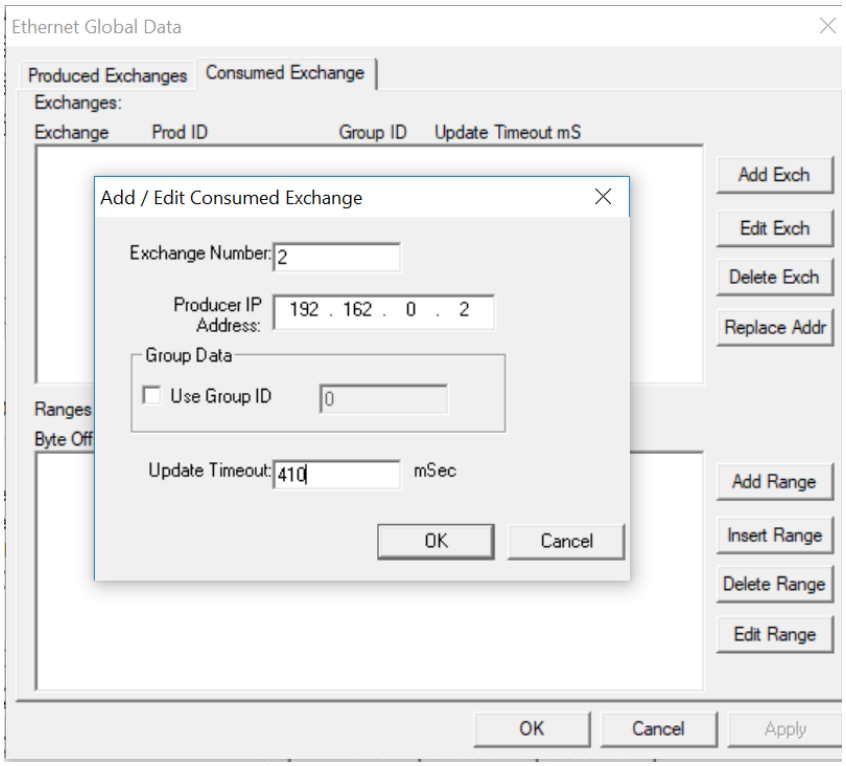
In this example, Node 1 will use Exchange 1 to transmit 10 words of data taken from %R100 through %R109.





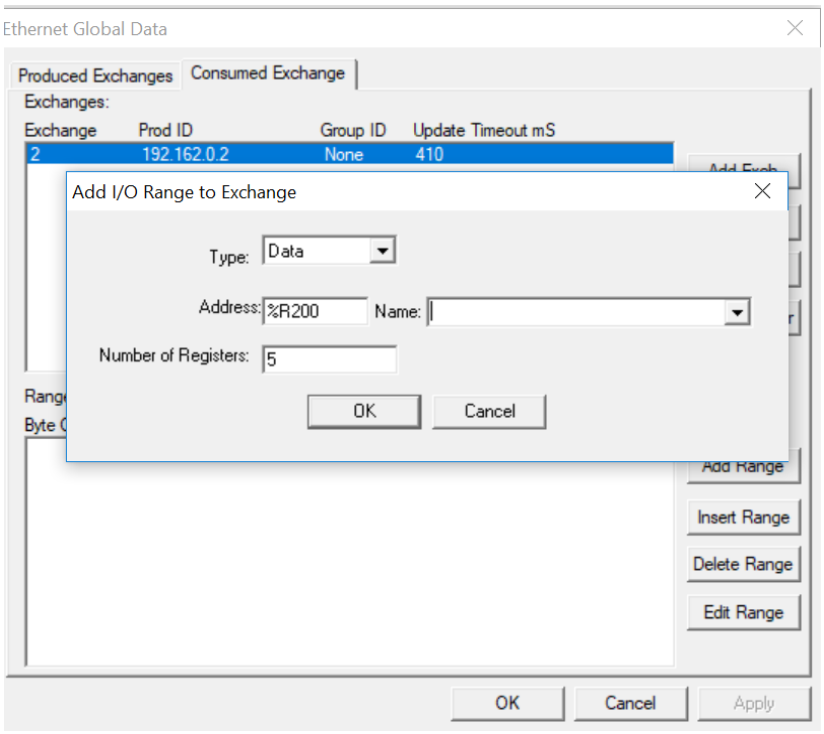
4. To configure Exchange 2 as Consumed Exchange for Node 1, select the Consumed Exchanges tab in the Ethernet Global Data dialog. Click on the **Add Exch** button to create a Consumed Exchange. Then fill in the parameters as shown.

NOTE: This is a **unicast**. Node 1 will be the only Consumer to receive Exchange 2 from Node 2 and will NOT be a member of a Group of Consumers. Also, the Update Timeout is set to 410ms, which means Node 1 will expect to receive Exchange 2 from Node 2 at least every 410ms. [This is twice the time, plus 10ms.]

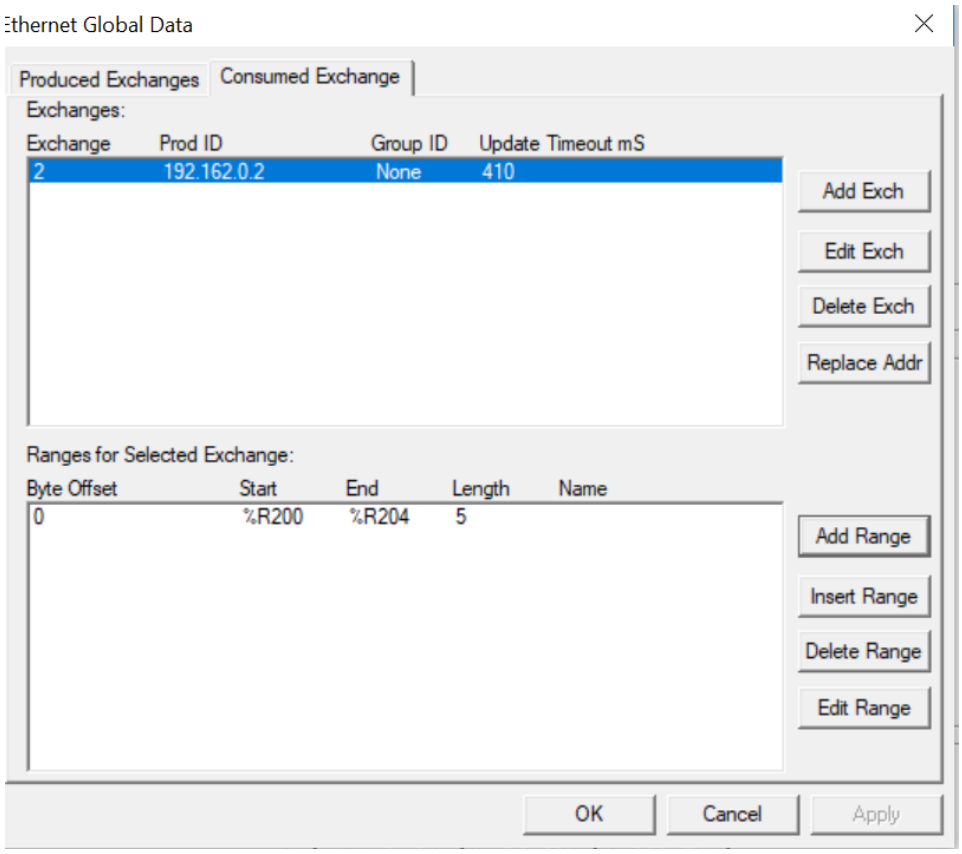


5. Click on the **Add Range** button in the EGD Configuration dialog to define a Data Block for Consumed Exchange 2. Then fill in the parameters as shown below and click **OK**.

   In this example, Node 1 will use Exchange 2 to receive 5 data words in %R200 through %R204.



At this point, the Consumed Exchanges tab in the EGD Configuration dialog should be as shown



6. Node 1 configuration is now complete. Click **OK**, save the program using an appropriate filename, such as **EGD Node 1.csp**, and then continue with Section 5.9.2, Node 2 Configuration.

5.8.2 EGD EXAMPLE 1 – CONFIGURATION OF NODE 2

To configure Node 2 for EGD Example 1, perform the following steps:

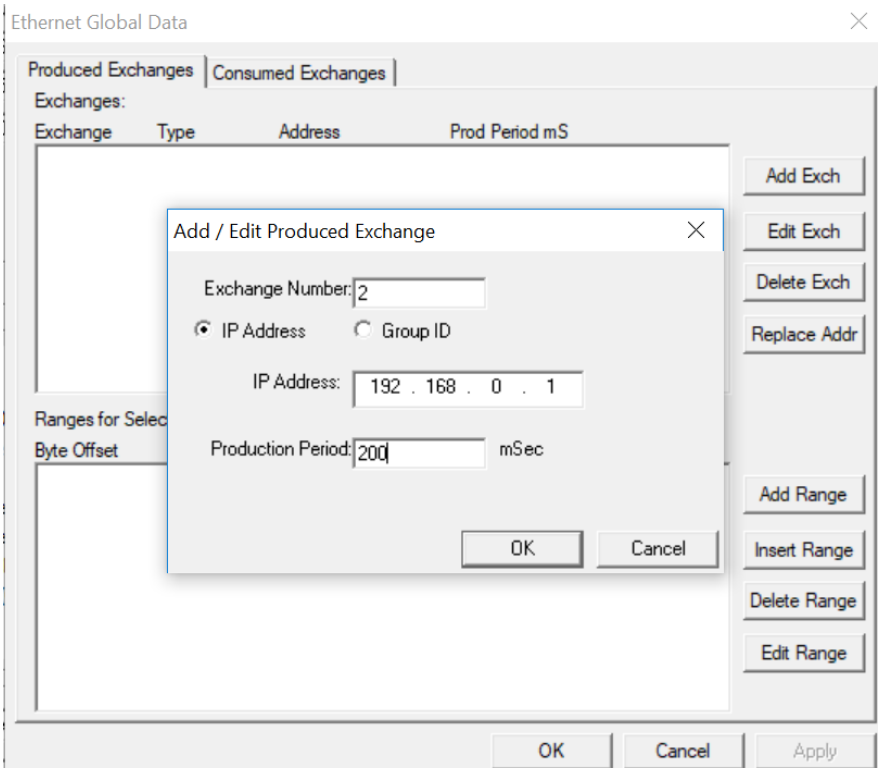
1. On the main Cscape screen, select **New** on the **File** menu to start a new user program. To configure EGD Exchanges, open the Ethernet Global Data Configuration dialog as described in Section 5.3.

In this example, Node 2 will have a Static IP Address, and the %R1, %R2, and %R3 registers will be used to report the Ethernet Module's status, firmware version, and IP Address.

NOTE: Since an IP address is 32-bits long, Node 2's IP Address (192.168.0.2) will actually be written into %R3 and %R4.

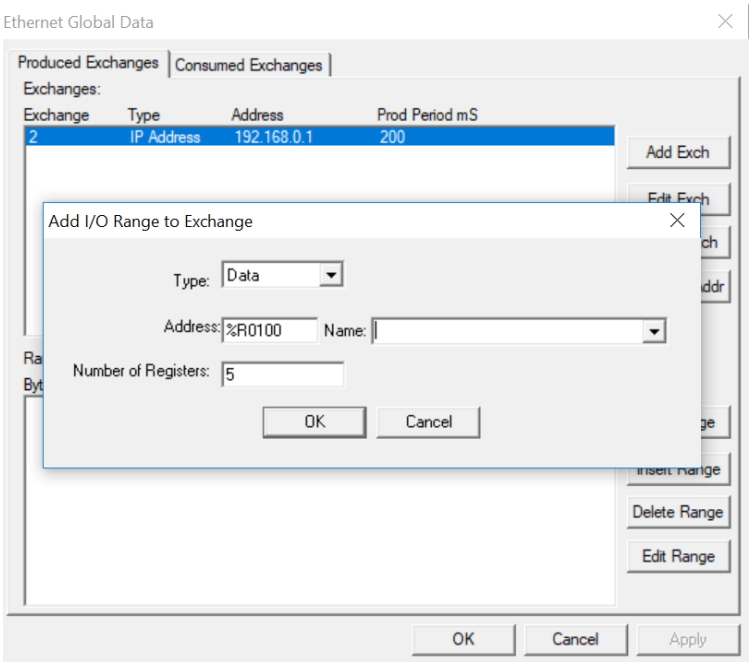
Node 2 will transmit Exchange 2 to a specific Consumer (Node 1) because this is a unicast communication, instead of to a group of Consumer (multicast).

The Production Period is set to 200, which will cause Node 2 to transmit Exchange 2 every 200ms.

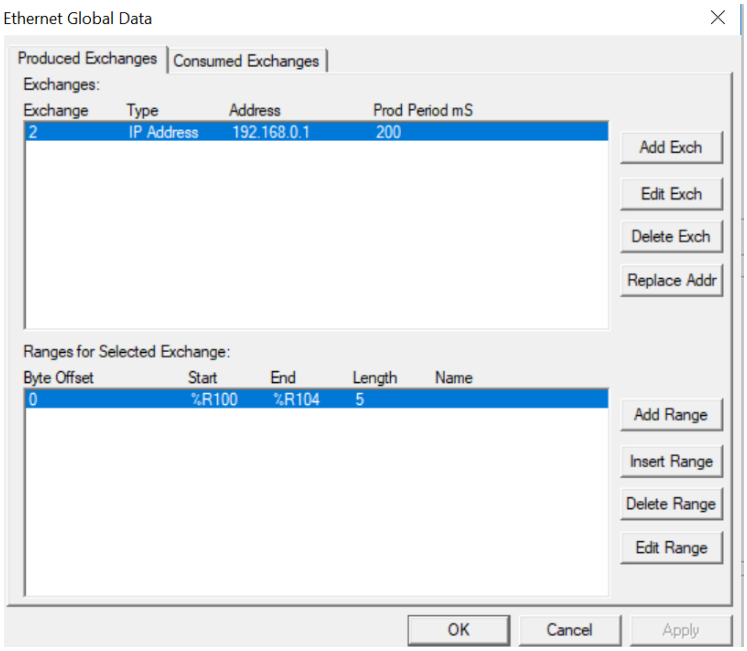


3. Click on the **Add Range** button in the EGD Configuration dialog to define a Data Block for Produced Exchange 2. Then fill in the parameters as shown below and click **OK**.

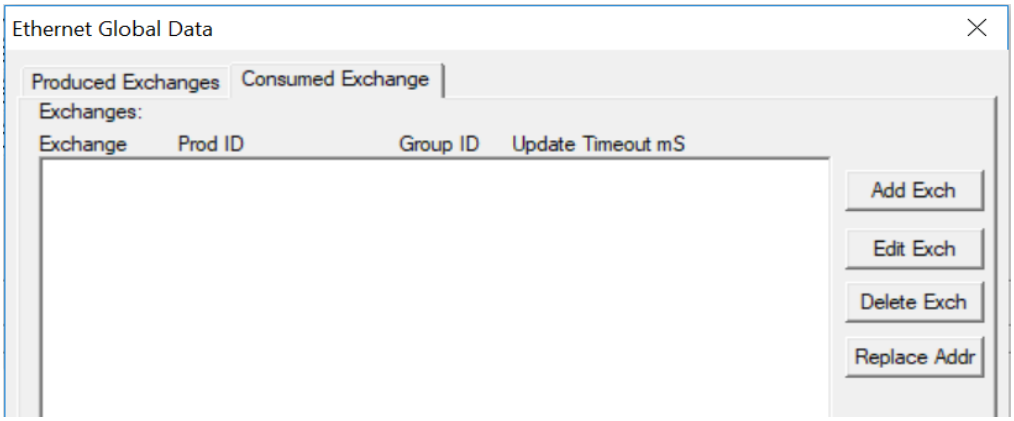
In this example, Node 2 will use Exchange 2 to transmit 5 words of data taken from %R100 through %R104.



At this point, the Produced Exchanges tab in the EGD Configuration dialog should be as shown in the figure below.



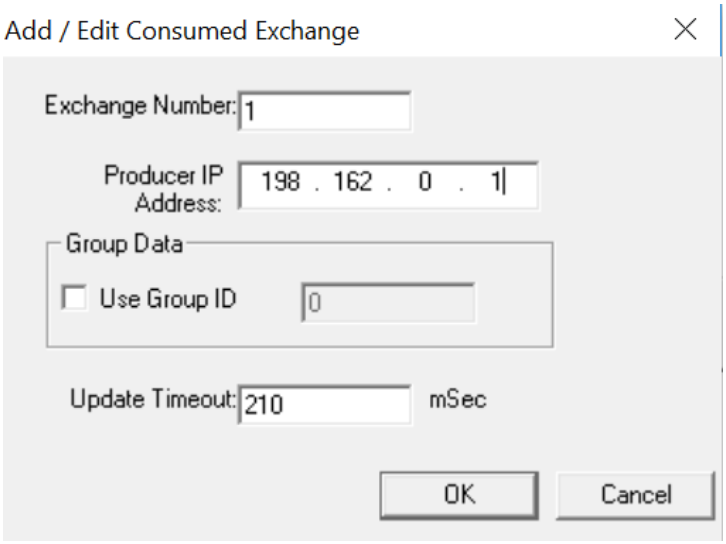
4. Now that Exchange 2 has been configured as a Produced Exchange for Node 2, it is time to configure Exchange 1 as a **Consumed Exchange** for Node 2. To do this, first select the Consumed Exchanges tab and click on the **Add Exch** button to create a Consumed Exchange.



Then fill in the parameters, as shown below and click OK.

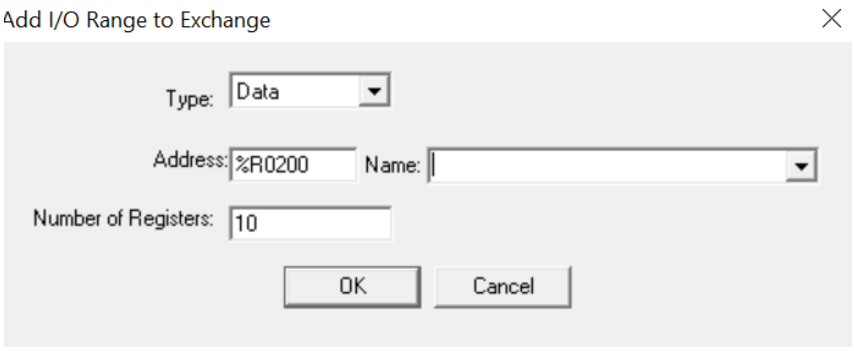
In this example, Node 2 will be the ONLY consumer to receive Exchange 1 from Node 1 and will NOT be a member of a Group of Consumers. Also, the Update Timeout is set to 210, which means that Node 2 will expect to receive Exchange 1 from Node 1 at least every 210ms.

NOTE: This is twice the time, plus 10ms that Node 1 was configured to transmit Exchange 1, as recommended in Table 5.1 in Section 5.2.

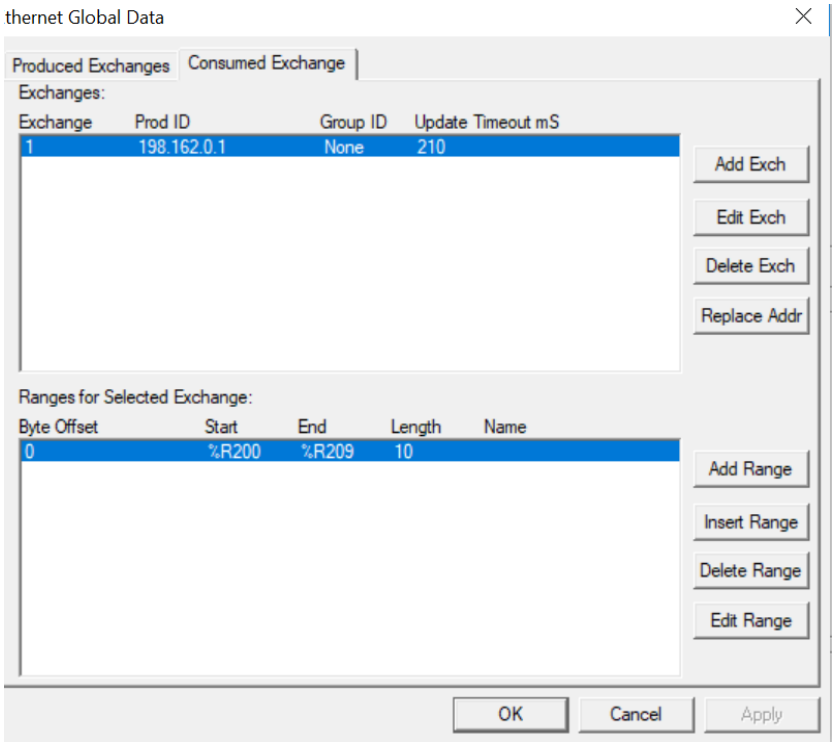


5. Click on the **Add Range** button, in the EGD Configuration dialog to define a Data Block for Consumed Exchange 1. Then fill in the parameters as shown below and click **OK**.

In this example, Node 2 will use Exchange 1 to receive 10 data words from %R200 through %R209.



Click **OK** so that dialog matches the one below.



6. Node 2 configuration is now complete. Click **OK** and save the user program using an appropriate filename, such as **EGD Node 2.csp**, and then start EGD communication between Node 1 and Node 2.

5.8.3 EGD EXAMPLE 1 – STARTING EGD COMMUNICATION BETWEEN NODE 1 & NODE 2

Now that both Node 1 and Node 2 have been configured for Example 1, start them communicating following these steps:

1. Open the saved Node 1 user program (**EGD Node 1.csp**), and download it to Node 1.
2. Open the saved Node 2 user program (**EGD Node 2.csp**) and download it to Node 2.
3. Connect both Node 1 and Node 2 to an Ethernet network and put them both in RUN mode.

At this point, Node 1 and Node 2 should be exchanging EGD data as follows:

1. Every 100ms, Node 1 will read its %R100 through %R109 registers and send them via Exchange 1 to be received by Node 2's %R200 through %R209 registers.

2. Every 100ms, Node 2 will read its %R100 through %R104 registers and send them via Exchange 2 to be received by Node 1 into Node 1's %R200 through %R204 registers.

5.9 EGD Example 2 – MULTICAST EXCHANGE

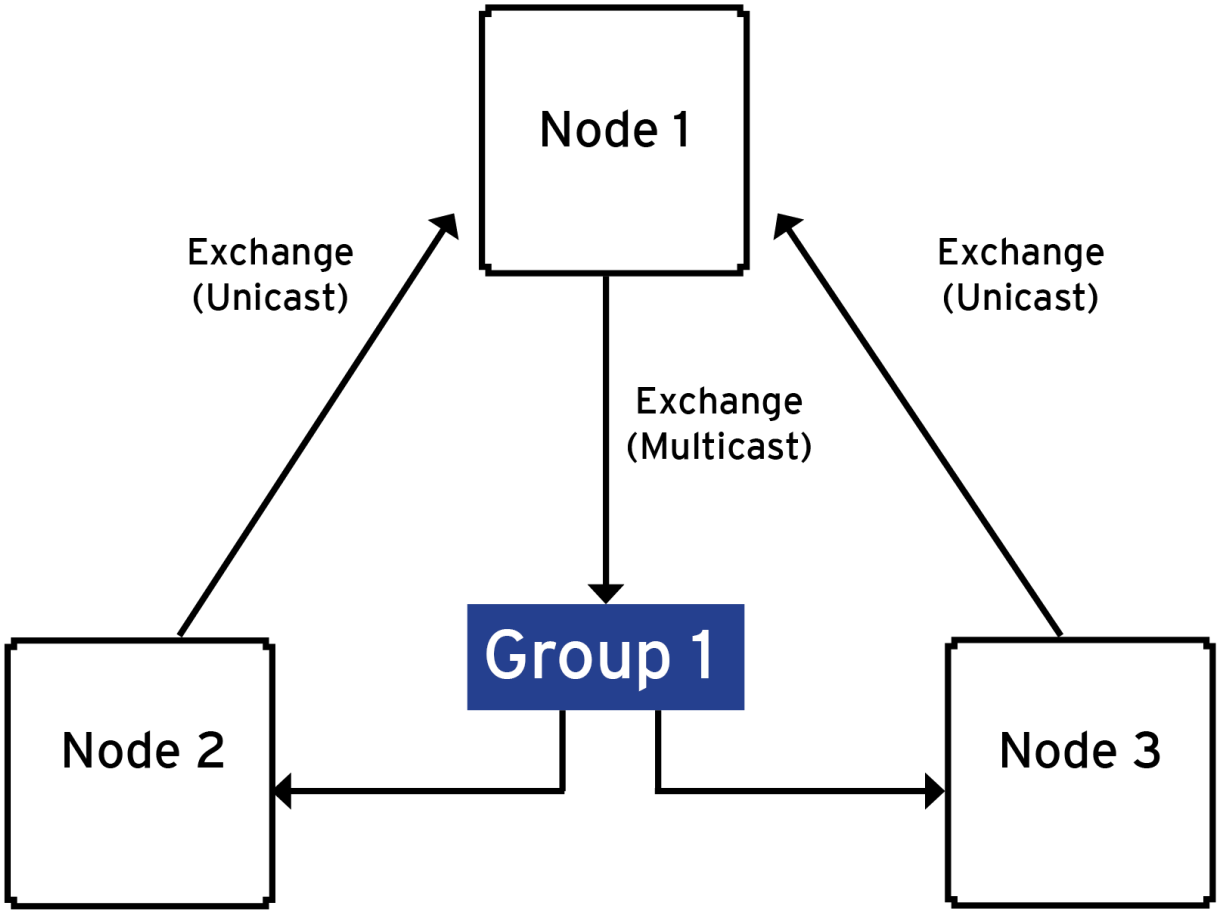


Figure 5.6 – Multicast - Three Nodes

5.9.1 EGD EXAMPLE – CONFIGURING NODE 1

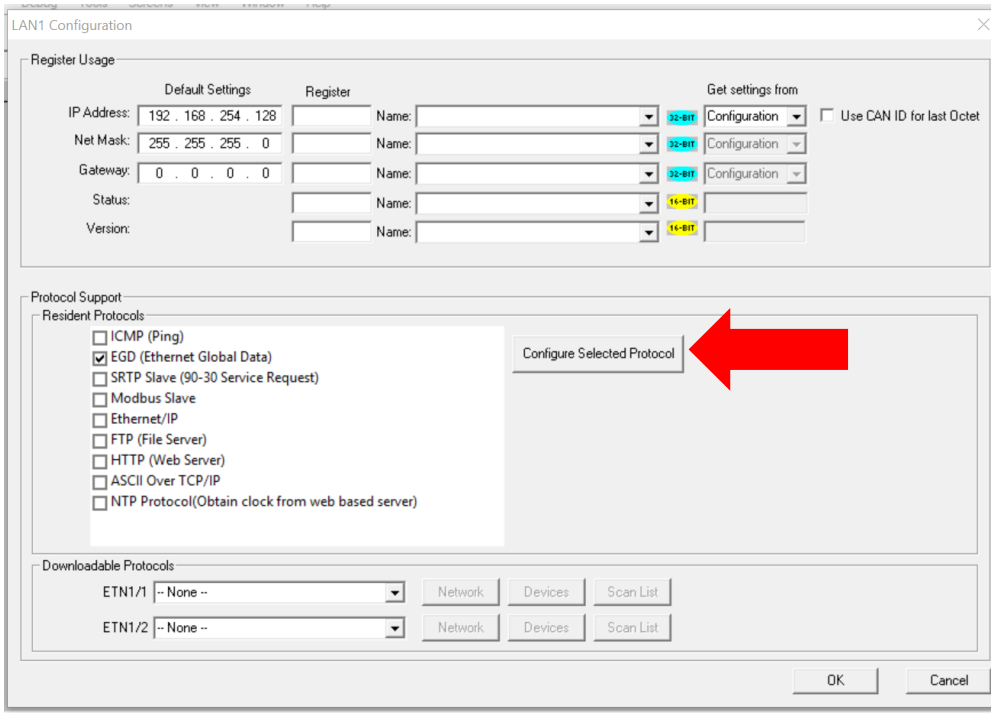
The following are the production and consumption range details:

Table 5.12 – EGD Example Specifications					
Node 1 – Production Node					
Exchange #1: Production Details		**Exchange #1: Produced Range Details**			
Exchange Number	#1	**Byte Offset**	**Start**	**Length**	
Produced to	Group 1	0	%R1001	50	
Production Period	100mS	100	%R1501	25	
		150	%R1601	25	
Node 2 – Consumption Node					
Exchange Number	#1	**Type**	**Byte Offset**	**Start**	**Length**
Producer IP	192.168.0.1	Data	0	%R1	50
Group Data ID	1	Data	100	%R101	25
Update Timeout	210mS	Status	n/a	%R1001	n/a
Node 3 – Consumption Node					
Exchange Number	#1	**Type**	**Byte Offset**	**Start**	**Length**
Producer IP	192.168.0.1	Data	0	%R1	50
Group Data ID	1	Data	100	%R101	25
Update Timeout	210mS	Status	n/a	%R1001	n/a

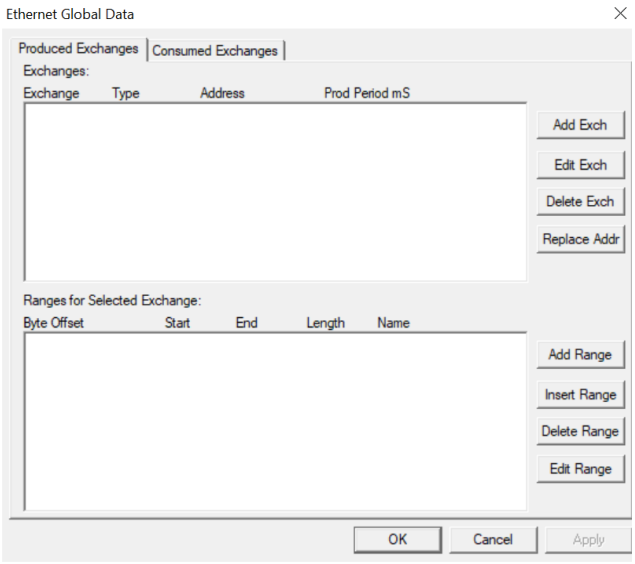
5.9.2 – MULTICAST CONFIGURATION NODE 1

To configure Node 1 for the EGD Example, perform the following steps:

1. On the main Cscape screen, select **New** on the **File** menu to start a new user program. Then open the OCS configuration dialog by selecting the following in Cscape: **Controller → Hardware Configuration (**or select "I-O" button from the Menu bar), then select Device and Model number. Select **EGD** checkbox → **Configure Selected Protocol**. Refer to image below.

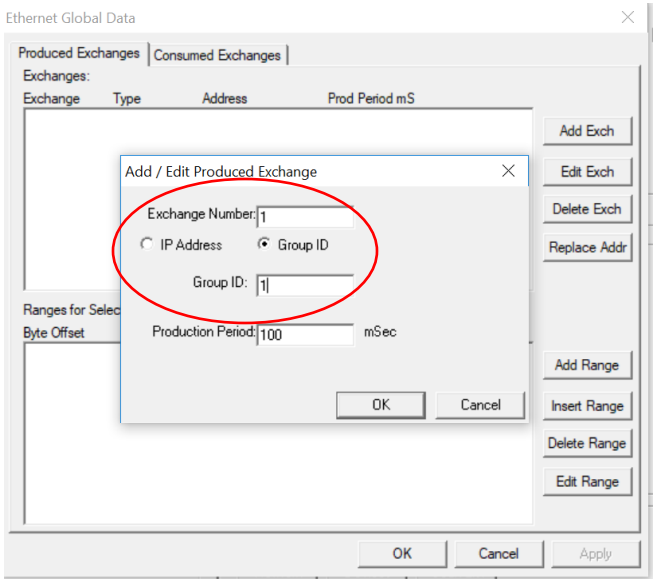


2. Once the **Configure Selected Protocol** button is selected, the **Ethernet Global Data** dialog is displayed. Two options are available: Produced Exchanges and Consumed Exchanges.

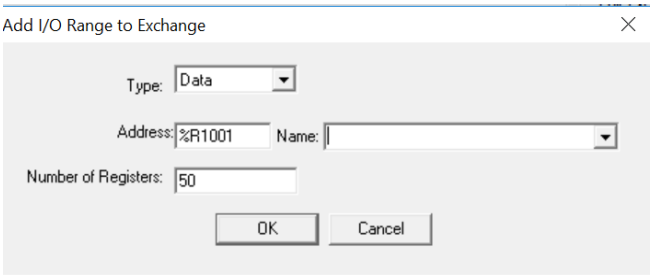


3. Select **Add Exch** to create a produced exchange, and then select **Group ID** and add appropriate group number. Click **OK**.

   NOTE: To create Multicast communication, use Group ID and not IP Address.



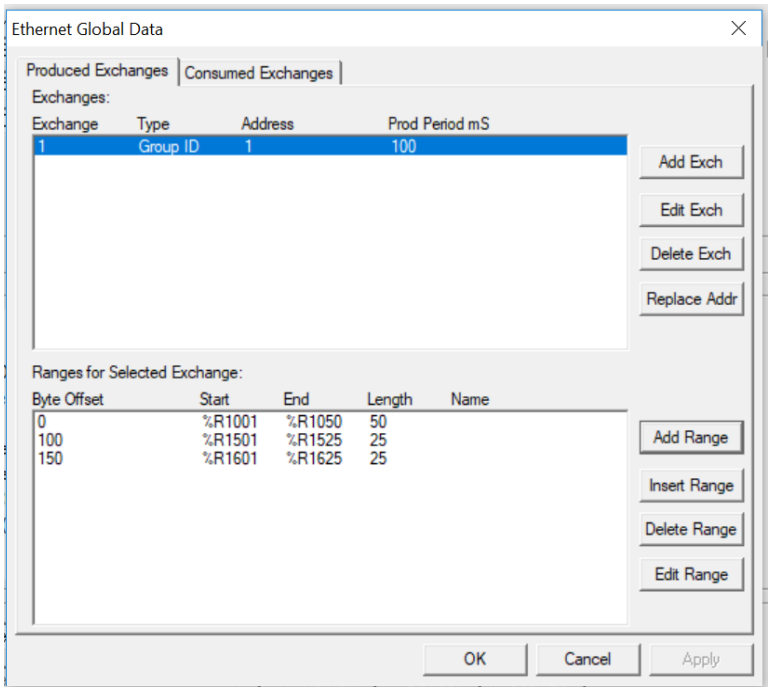
4. The Produced Exchange has been created. Select **Exchange 1**, then click on the **Add Range** button to define a Data Block for Produced Exchange 1. Then fill in the parameters and click **OK**.



Continue adding the rest of the produced range details.

Exchange #1: Produced Range Details		
Byte Offset	**Start**	**Length**
0	%R1001	50
100	%R1501	25
150	%R1601	25

At this point, the Produced Exchanges tab in the EGD Configuration dialog should be as shown in the figure below.



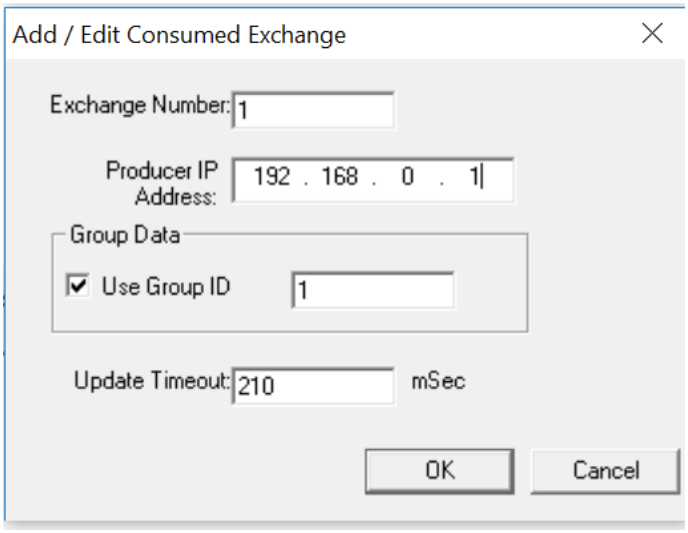
6. Node 1 configuration is now complete. Click OK, save the program using an appropriate filename such as **EX_2 Node 1.csp**, and then continue with Node 2 configuration.

NOTE: This is a **multicast**. Node 1 will produce exchanges to **GROUP 1** which in turn will send exchanges to all nodes that are assigned to Group 1.

5.9.3 MULTICAST EGD EXAMPLE - CONFIGURING NODE 2

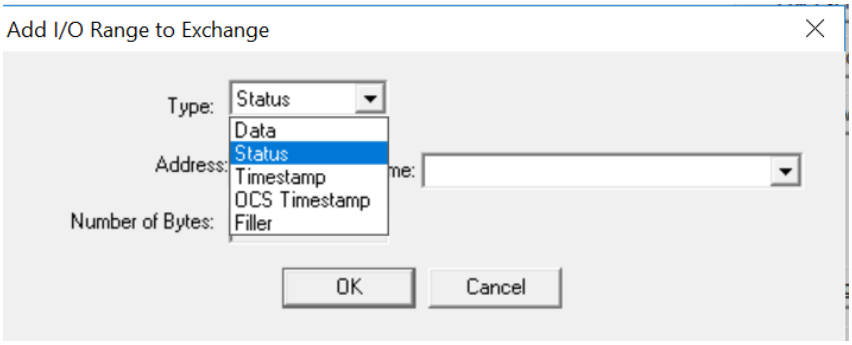
1. On the main Cscape screen, select **New** on the **File** menu to start a new user program. Then open the OCS configuration dialog by selecting the following in Cscape: **Controller → Hardware Configuration (**or select "I-O" button from the Menu bar), then select Device and Model number. Select **EGD** checkbox → **Configure Selected Protocol**. (Refer to Section 5.9.2)

2. Once Exchange 1 has been configured as a Produced Exchange for Node 1, configure Exchange 1 as a Consumed Exchange for Node 2. To do this, first select the **Consumed Exchanges** tab and click on the **Add Exch** button to create a Consumed Exchange.



Then fill in the parameters, as shown below, and click **OK**.

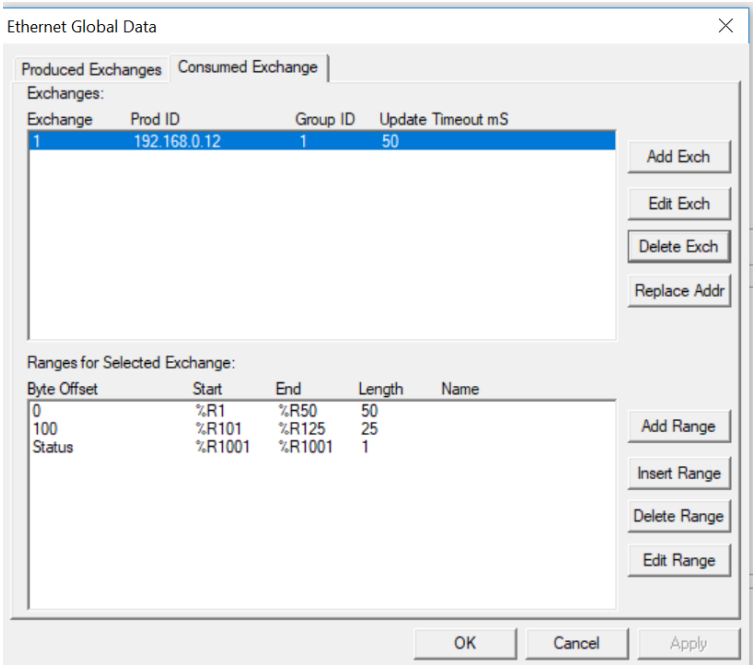
3. Once a Consumed Exchange has been made and highlighted, click on the **Add Range** button, in the EGD Configuration dialog, to define a Data Block for Consumed Exchange 1. Then fill in the parameters, as shown in below, and click **OK**.



Continue adding the rest of the produced range details for Node 2.

Exchange #1: Consumed Range Details			
Type	**Byte Offset**	**Start**	**Length**
Data	0	%R1	50
Data	100	%R101	25
Status	n/a	%R1001	n/a

At this point, the Consumed Exchanges tab in the EGD Configuration dialog should be as shown in the figure below.



4. Node 2 configuration is now complete. Click OK, save the program using an appropriate filename such as **EX_2 Node 2.csp**, and then continue with Node 3 configuration.

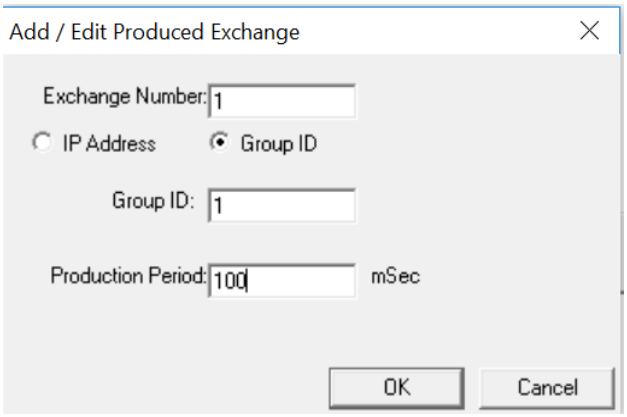
5.9.4 MULTICAST EGD EXAMPLE – CONFIGURING NODE 3

1. On the main Cscape screen, select **New** on the **File** menu to start a new user program. Then open the OCS configuration dialog by selecting the following in Cscape: **Controller → Hardware Configuration** (or select "I-O" button from the Menu bar), then select Device and Model number. Select **EGD** checkbox → **Configure Selected Protocol**. (Refer to Section 5.9.2)

Exchange #1: Consumed Range Details			
Type	**Byte Offset**	**Start**	**Length**
Data	0	%R1	50
Data	150	%R101	25
Status	n/a	%R1001	n/a

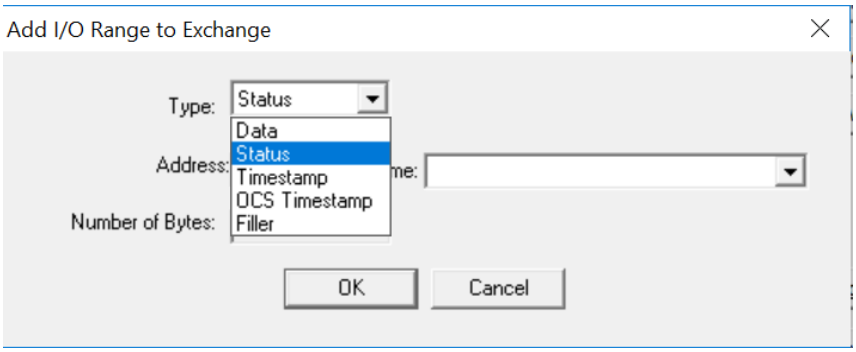
   At this point, the Consumed Exchanges tab in the EGD Configuration dialog should be as shown in the figure below.

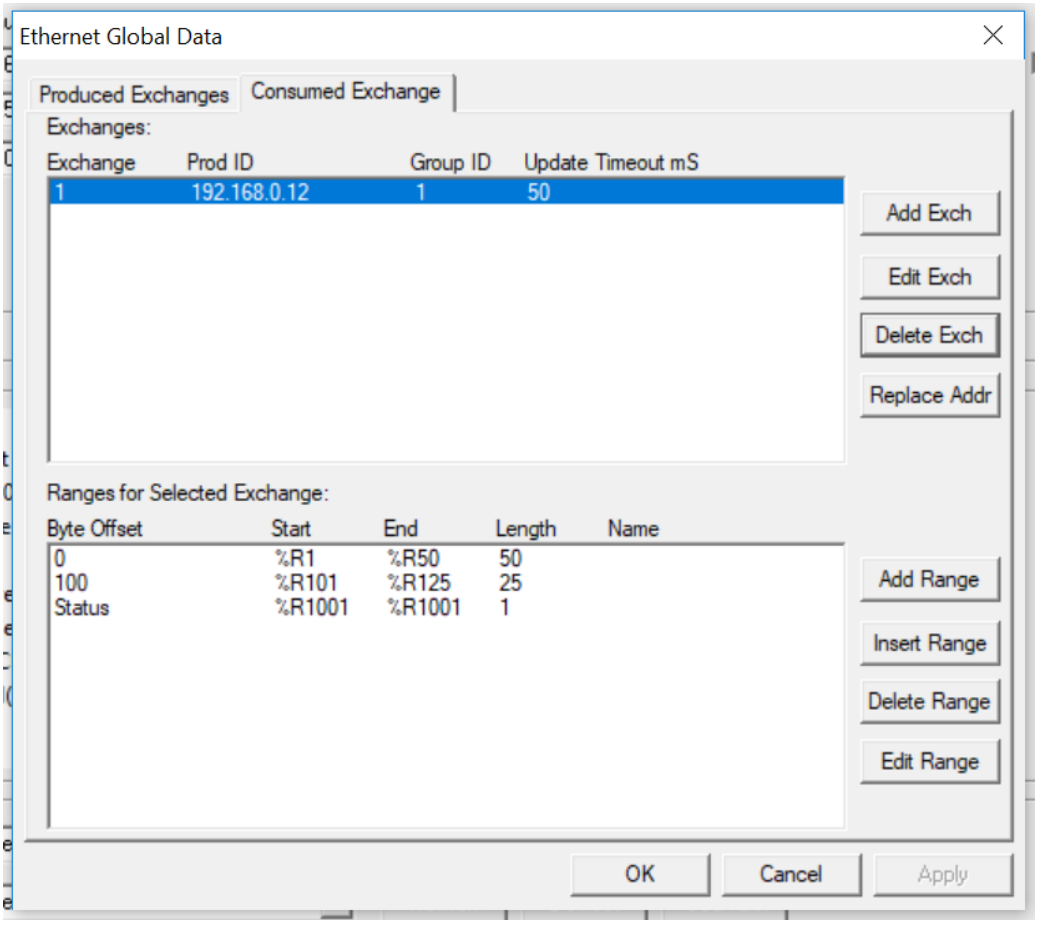
2. Repeat steps from Node 2. Open Then fill in the parameters, as shown below, and click **OK**.



Then fill in the parameters, as shown below, and click **OK**.

3. Once a Consumed Exchange has been made and highlighted, click on the **Add Range** button, in the EGD Configuration dialog, to define a Data Block for Consumed Exchange 1. Then fill in the parameters, as shown in below, and click **OK**.





4. Node 1 configuration is now complete. Click OK, save the program using an appropriate filename such as **EX_2 Node 2.csp**, and then continue with Node 3 configuration.

5.9.5 EGD Example 1 – Starting multicast egd communication

1. Open the saved Node 1 user program (Node 1.csp) and download it to Node 1.
2. Open the saved Node 2 user program (Node 2.csp) and download it to Node 2.
3. Open the saved Node 3 user program (Node 3.csp) and download it to Node 3.

Connect both Node 1, Node 2, and Node 3 to an Ethernet network and put them both in RUN mode.

CHAPTER 6: SERVICE REQUEST TRANSFER PROTOCOL (SRTP)

6.1 SRTP Overview

Service Request Transfer Protocol (SRTP) is a GE Fanuc Automation protocol, which allows a remote SRTP Client to request services from an SRTP Server. In this context, the OCS acts as an SRTP Server, which responds to requests from one or more SRTP Clients.

Since SRTP was originally designed to support services provided by GE Fanuc Series 90 PLC products, the OCS's SRTP protocol, for OCS and RCS products, does not support all possible SRTP services. The OCS's implementation of SRTP is mainly limited to those services required for the exchange of PLC register data.

The following SRTP service requests are supported by the OCS:

Table 6.1 – SRTP Service Requests	
Request Code	Service Name
0	PLC_SSTAT
1	PLC_LSTAT
4	READ_SMEM
7	WRITE_SMEM
33	CHG_PRIV_LEVEL
67	RET_CONFIG_INFO
79	SESSION_CONTROL
97	PLC_FEATURES_SUPP

The READ_SMEM and WRITE_SMEM requests are supported for the following register types:

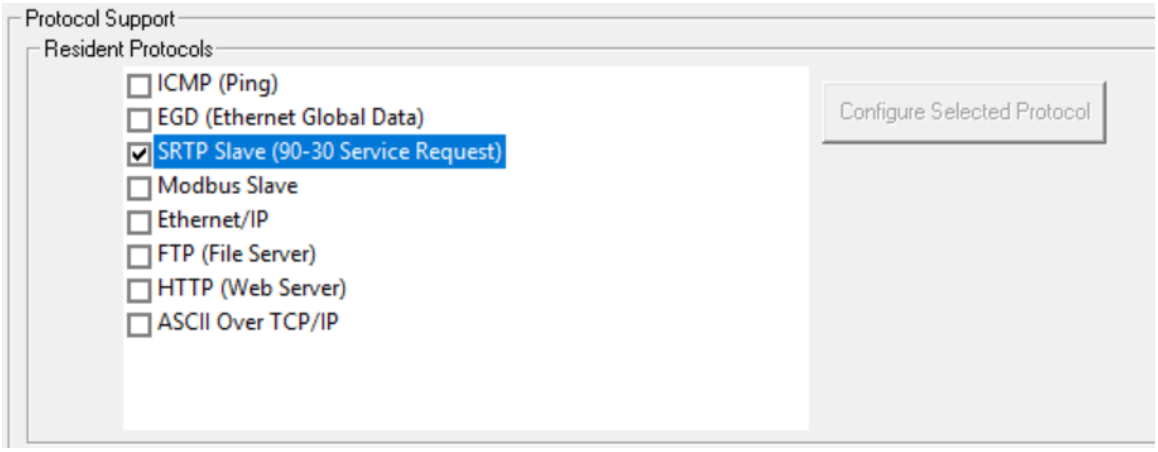
Table 6.2 – SRTP Register Types		
Selector	Register	Type
8	%R	16 bit
10	%AI	16 bit
12	%AQ	16 bit
16	%I	8 bit
18	%Q	8 bit
20	%T	8 bit
22	%M	8 bit
30	%S	8 bit
70	%I	1 bit
72	%Q	1 bit
74	%T	1 bit
76	%M	1 bit
84	%S	1 bit

6.2 SRTP Configuration

If SRTP protocol will be used in the application, the general OCS Configuration, previously described in Chapter 3, must be performed, and SRTP must be enabled. To enable SRTP protocol, use Cscape Programming Software, to perform the following two steps:

1. Open the OCS configuration dialog (**Controller → Hardware Configuration → LAN1 / Config**) as described in Chapter 3.

2. Enable SRTP by checking the **SRTP Slave (90-30 Service Request)** checkbox in the Module Configuration dialog, see below.



No additional configuration is required for SRTP protocol, so the Configure Selected Protocol button is disabled. If SRTP is enabled and the OCS has been assigned an IP Address and Net Mask, then it will respond to service requests from an SRTP Client.

6.3 SRTP Operation

With respect to SRTP protocol, the OCS's behavior depends on whether the OCS is in RUN mode or not.

- When the OCS is in RUN mode, the OCS responds normally to SRTP Client requests.
- When the OCS is not in RUN mode, it still responds to SRTP Client requests, with the following differences:

    1. When a WRITE_SMEM request is received from an SRTP Client, to write a block of OCS registers, a reply is sent to the client, but the data is not written to OCS registers.

    2. When a READ_SMEM request is received from an SRTP Client, to read a block of OCS registers, a reply is sent to the client, but the data returned to the client is the last known values of the requested OCS registers, and is not actually read from the OCS registers.

    3. For all requests received from an SRTP Client, the reply will indicate to the client that the OCS is not in RUN mode, by returning the RUN DISABLED code in the Piggyback Status word.

CHAPTER 7: MODBUS TCP SERVER PROTOCOL

7.1 Modbus Configuration

If Modbus TCP Server protocol will be used in the application, the general OCS configuration, previously described in Chapter 3, must be performed, and Modbus must be enabled. To enable Modbus protocol, use Cscape Programming Software, to perform the following two steps:

1. Open the Ethernet LAN1 Configuration dialog (**Controller → Hardware Configuration → LAN1 / Config**), as described in Chapter 3.

2. Enable Modbus by checking the **Modbus Server** checkbox in the LAN1 Configuration dialog.

No additional configuration is required for Modbus protocol As long as Modbus is enabled, and the OCS has been assigned an IP Address and Net Mask, it will respond to service requests from a Modbus TCP Client.

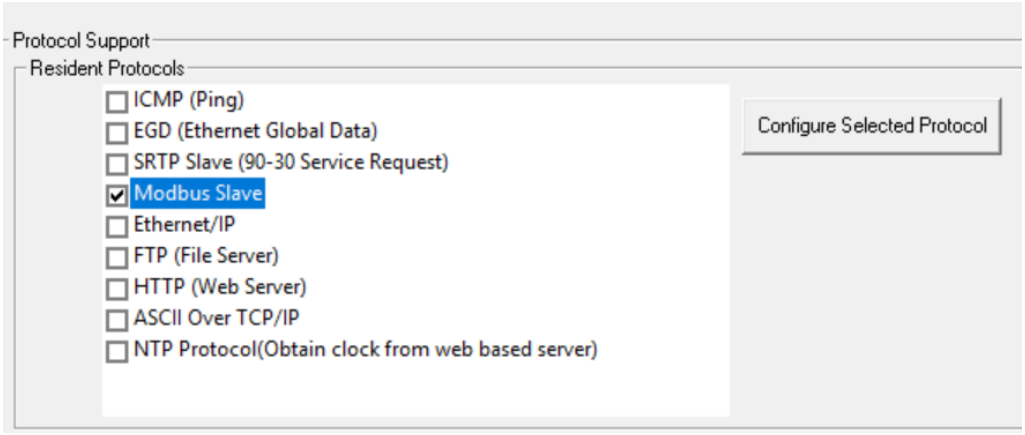
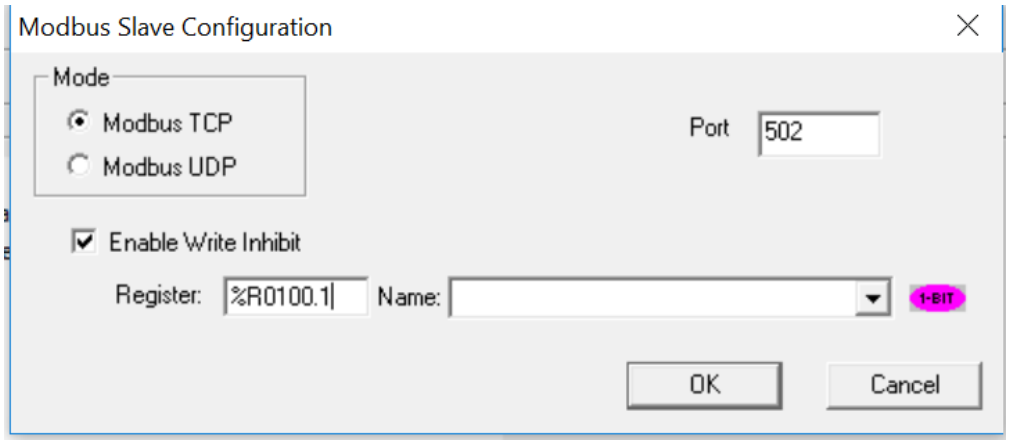


Figure 7.1 – LAN1: Modbus Server (Slave)

3. If there is a need to use Modbus UDP or to change the default Modbus TCP port, or if there needs to be restrictions on Modbus Write Access to this OCS, click on the Configure Selected Protocol button.



4. To restrict Modbus Write access to this OCS, check the Enable Write Inhibit checkbox. Specify a 1-bit register that, when set high, will not allow the Modbus Client to write data to this OCS. Change the mode to Modbus TCP or Modbus UDP as needed per the Client device. Change the Port used for Modbus communications if the default port 502 is not sufficient.

7.2 Modbus Operation

With respect to Modbus TCP Server protocol, the OCS's behavior depends on whether the OCS is in RUN mode or not.

When the OCS is in RUN mode, it responds normally to Modbus TCP Client requests.

When the OCS is not in RUN mode, it still responds to Modbus TCP Client requests, with the following differences:

1. When a request is received from a Modbus TCP Client to write a block of OCS registers, a reply is sent to the Client, but the data is not written to OCS registers.

2. When a request is received from a Modbus TCP Client, to read a block of OCS registers, a reply is sent to the Client, but the data returned to the Client is all zeroes.

1. A Modbus TCP Client can determine if the OCS is RUN mode in either of the following two ways:

    a. By reading the OCS Always On register (%S7), which is **1** if the OCS is in RUN mode and **0** if the OCS is not in RUN mode (because of item 2 above).

    b. By issuing a Read Exception Status request, which returns an Exception Status of **0** if the OCS is in RUN Mode or **1** if the OCS is not in RUN mode.

2. When "Write Inhibit" is enabled in the configuration:

    a. If the configured register value is "High", client cannot write any values to the server.
    b. If the configured register value is "Low", client can be enabled to write values to the server.

CHAPTER 8: ETHERNET / IP PROTOCOL

8.1 Ethernet / IP Overview

Ethernet / IP protocol is ODVA's Common Industrial Protocol (CIP) over Ethernet. Figure 8.1 illustrates the protocol layers. Ethernet/IP starts at the Ethernet Physical Layer, and moves up through the IP, TCP/UDP, and Encapsulation Layers. It is beyond the scope of this supplement to discuss the specifics of the Ethernet/IP protocol. See the Ethernet/IP specification at http://www.odva.org for more details of the Ethernet/IP protocol.

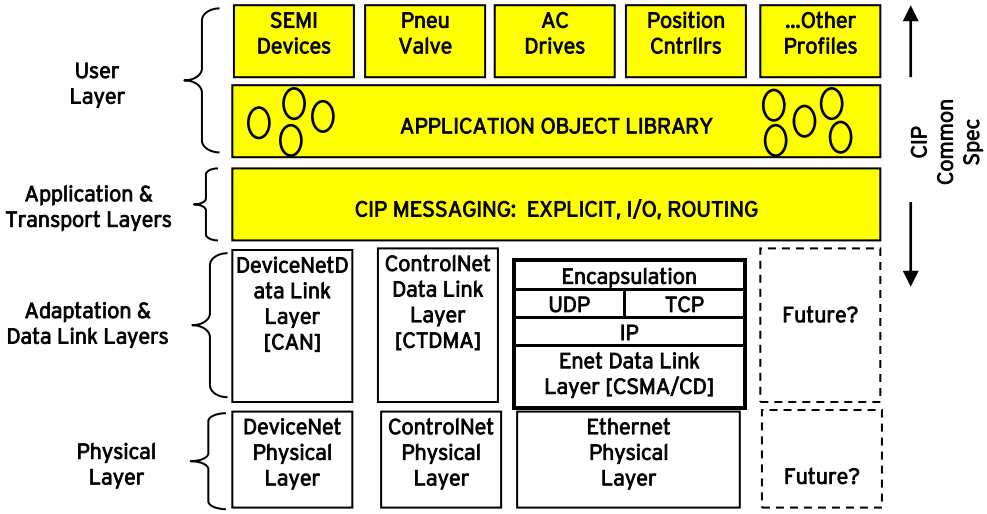


Figure 8.1 – Ethernet / IP Protocol Layers

The OCS provides Ethernet/IP server capability, and implements the following CIP objects:

- Identity → Instance #1
- Connection Manager → Instance #1
- Message Router → Instance #1
- Assembly → Instance #100 (0x64) for Consumed data
- Assembly → Instance #101 (0x65) for Produced data
- TCP/IP → Instance #1
- Ethernet Link → Instance #1

The OCS has an Unconnected Message Manager (UCMM), which forwards the following services on to the addressed object:

- Forward Open (Connection Manager object service)
- Forward Close (Connection Manager object service)
- Get Attribute Single
- Set Attribute Single

It will forward the above services on to the addressed object, however the addressed object may or may not support the service.

The OCS supports Class 3 connections for explicit messaging, and Class 1 connections for I/O (Implicit) messaging.

The OCS can be configured to produce and consume 0 bytes of data up to 256 bytes of data.

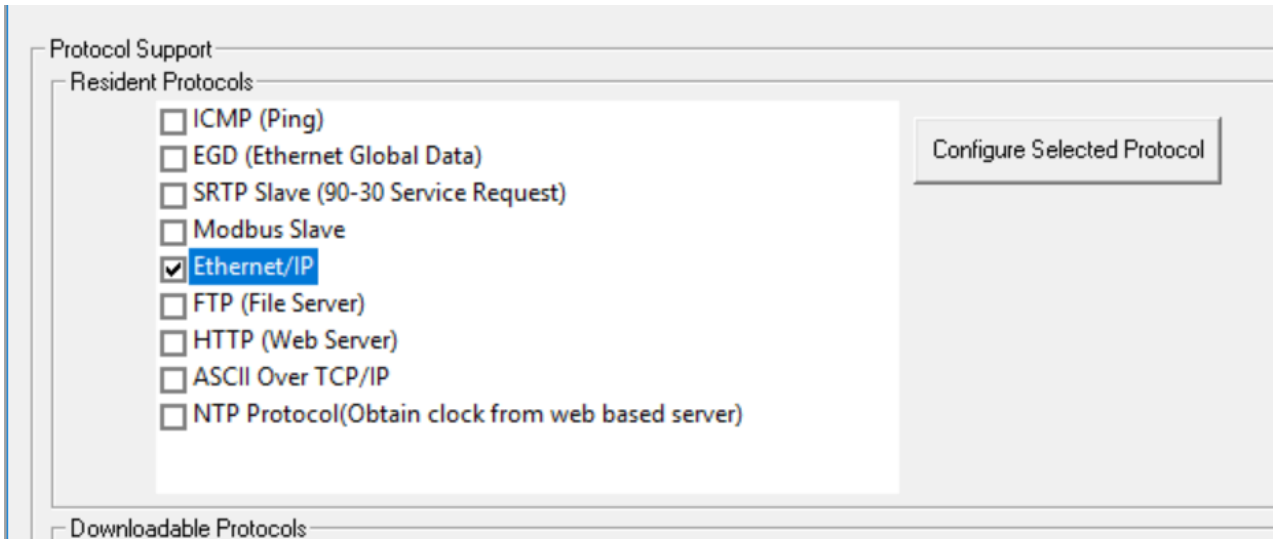
The TCP connections used for Ethernet/IP have an inactivity timeout of 60 seconds. In the event of an inactivity timeout, the TCP connection will automatically close.

8.2 Ethernet/IP Configuration

If Ethernet/IP protocol will be used in the application, Ethernet/IP configuration must be performed, in addition to the general Ethernet LAN1 Configuration previously described in Chapter 3. To configure Ethernet/IP protocol, use Cscape Programming Software to perform the following five steps:

1. Open the Ethernet LAN1 Configuration dialog (**Controller → Hardware Configuration → LAN1/Config**).

2. Enable Ethernet/IP by checking the **Ethernet IP** checkbox.



3. Click on the **Configure Selected Protocol** button next to the **Ethernet/ IP** checkbox to open the **Ethernet IP Configuration** dialog.

4. Set the location and size of Produced and Consumed data.
5. Also, set the location of the 16-bit Ethernet/IP Status word.

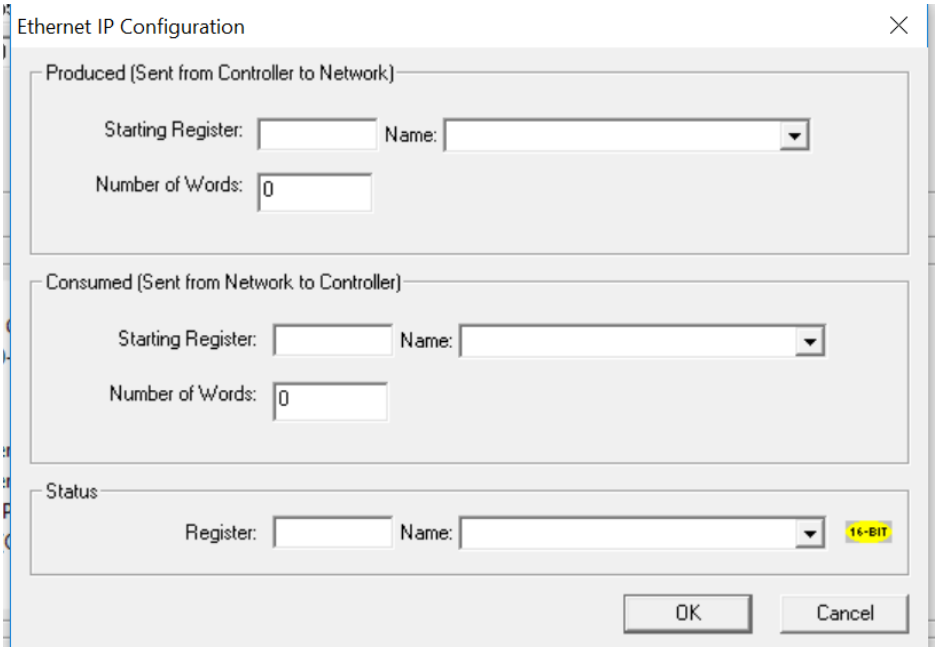


Figure 8.2 – Ethernet IP Configuration

8.3 Ethernet/IP Operation

After configuring the OCS as described in Section 8.2, it is ready to respond to Ethernet/IP requests. The OCS handles unconnected requests anytime.

To exchange I/O data with the OCS, a class 1 connection pair must be established. The connection pair consists of a class 1 originator-to-target connection and a class 1 target-to-originator connection, both of which should be set for point-to-point Transport Type. The data sizes must match the OCS's configured Consumed and Produced data sizes, respectively. The application path should be set as follows:

- The originator-to-target connection point should be set to 100 (0x64), which stands for the assembly class (4), instance 100 (0x64).
- The target-to-originator connection point should be set to 101 (0x65), which stands for the assembly class (4), instance 101 (0x65).

There is NO configuration assembly instance. All the class 1 connection pair setup data is transmitted through a forward open service to the connection manager.

Once the class 1 connection pair is established, I/O data is exchanged to/from the OCS through these connections. The Consumed data (data coming from the scanner) and Produced data (data going to the scanner) are available at the locations within the OCS register map as configured.

The TCP connections used for all encapsulated messages and explicit messages have an inactivity timeout of 60 seconds. If the TCP connections are required to stay open, a NOP encapsulated command can be used at a periodic rate to keep the TCP connection open.

When the OCS is not is RUN mode, Produced data is all zeros, and Consumed data is not written to the configured registers within the OCS. When the OCS is in RUN mode, the configured Produced OCS registers are read and sent as Produced data, while Consumed data is received and written to the configured Consumed OCS registers.

The Status word provides Ethernet/IP connection status. The upper byte of the word contains the Class 3 (Explicit) connection count and the lower byte contains the Class 1 (IO) connection count.

NOTE: When the Status word indicates no connections, the Consumed OCS registers contain old data.

PART THREE: DOWNLOADABLE INDUSTRIAL PROTOCOLS

CHAPTER 9: MODBUS TCP/IP CLIENT

Section 9.1: Overview

MODBUS TCP/IP is a variant of the MODBUS family of simple, vendor-neutral communication protocols. It is intended for supervision and control of automation equipment. Specifically, it covers the use of MODBUS messaging in an 'Intranet' or 'Internet' environment using the TCP/IP protocols. The most common use of this protocol is for Ethernet attachment of PLC's, I/O modules, and 'gateways' to other simple field buses or I/O networks.

Modbus TCP is a Client / Server protocol, which allows a remote Modbus TCP Client to request services from a Modbus TCP Server. In this context, the OCS acts as a Modbus TCP Server, which responds to requests from one or more Modbus Clients.

All Modbus requests that contain the OCS's IP Address are serviced. The client needs to be configured with the OCS's IP Address, and most clients also require the server unit number. Since each Ethernet module must have its own unique IP Address, the server unit number is not relevant and is discarded by the Ethernet module.

To access OCS registers, a Modbus TCP Client must be configured with the appropriate register type and offset. This is usually accomplished with one of two methods:

The first method uses either **Traditional Modbus References** or **Expanded Modbus References**, in which the high digit represents the register type and the lower digits represent the register offset (starting with register 1 for each type). Since only four Modbus register types (0, 1, 3, and 4) can be represented in this manner, the OCS's Modbus implementation packs several OCS register types into each Modbus register type. Starting addresses of each OCS register type are shown in the **Traditional Modbus Reference** and **Expanded Modbus Reference** columns.

The second method requires the Modbus TCP Client to be configured with a specific **Modbus Command and Offset**. The supported Modbus commands and the associated offsets are also illustrated in the tables below.

NOTE: Each controller family has its own Modbus Mapping table.

XLE/XLEe Series – Modbus Mapping					
OCS Ref.	Max. Range	Trad. Modbus Ref. (5 digits)	Expanded Modbus Ref. (6 digits)	Modbus Command(s)	Modbus Offset
%I1	2048	10001	010001	Read Input Status (2)	0
%IG1	64	13001	013001		3000
%S1	13	14001	014001		4000
%K1	10	15001	015001		5000
%Q1	2048	00001	000001	Read Coil Status (1) Force Coil (5) Force Multiple Coils (15)	0
%M1	2048	03001	003001		3000
%T1	2048	06001	006001		6000
%QG1	64	09001	009001		9000
%A1	512	30001	030001	Read Input Register (4)	0
%AIG1	32	33001	033001		3000
%SR1	192	34001	034001		4000
%AQ1	512	40001	040001	Read Holding Registers (3) Load Register (6) Load Multiple Registers (16)	0
%R1	2488	40513	040513		0
%R1	2048	43001	043001		3000
%AQG1	32	46001	046001		6000
%R1	9999	--	410001		10000

XLT/XLTe Series – Modbus Mapping					
OCS Ref.	Max. Range	Trad. Modbus Ref. (5 digits)	Expanded Modbus Ref. (6 digit)	Modbus Command(s)	Modbus Offset
%I1	2048	10001	010001	Read Input Status (2)	0
%IG1	64	13001	013001		3000
%S1	13	14001	014001		4000
%K1	4	15001	015001		5000
%Q1	2048	00001	000001	Read Coil Status (1) Force Coil (5) Force Multiple Coils (15)	0
%M1	2048	03001	003001		3000
%T1	2048	06001	006001		6000
%QG1	64	09001	009001		9000
%A1	512	30001	030001	Read Input Register (4)	0
%AIG1	32	33001	033001		3000
%SR1	192	34001	034001		4000
%AQ1	512	40001	040001	Read Holding Registers (3) Load Register (6) Load Multiple Registers (16)	0
%R1	2488	40513	040513		0
%R1	2048	43001	043001		3000
%AQG1	32	46001	046001		6000
%R1	9999	--	410001		10000

XL4 Series – Modbus Mapping					
OCS Ref.	**Max. Range**	**Trad. Modbus Ref. (5 digits)**	**Expanded Modbus Ref. (6 digits)**	**Modbus Command(s)**	**Modbus Offset**
%I1	2048	10001	010001	Read Input Status (2)	0
%IG1	256	13001	013001		3000
%S1	256	14001	014001		4000
%K1	4	15001	015001		5000
%Q1	2048	00001	000001	Read Coil Status (1) Force Coil (5) Force Multiple Coils (15)	0
%M1	2048	03001	003001		3000
%T1	2048	06001	006001		6000
%QG1	256	09001	009001		9000
%A1	512	30001	030001	Read Input Register (4)	0
%AIG1	32	33001	033001		3000
%SR1	256	34001	034001		4000
%AQ1	512	40001	040001	Read Holding Registers (3) Load Register (6) Load Multiple Registers (16)	0
%R1	2488	40513	040513		0
%R1	2048	43001	043001		3000
%AQG1	32	46001	046001		6000
%R1	9999	--	410001		10000

XL+, XL7 & EXL6 – Modbus Mapping					
OCS Ref.	**Max. Range**	**Trad. Modbus Ref. (5 digits)**	**Expanded Modbus Ref. (6 digits)**	**Modbus Command(s)**	**Modbus Offset**
%I1	2048	10001	010001	Read Input Status (2)	0
%IG1	256	13001	013001		3000
%S1	256	14001	014001		4000
%K1	5	15001	015001		5000
%Q1	2048	00001	000001	Read Coil Status (1) Force Coil (5) Force Multiple Coils (15)	0
%M1	2048	03001	003001		3000
%T1	2048	06001	006001		6000
%QG1	256	09001	009001		9000
%A1	512	30001	030001	Read Input Register (4)	0
%AIG1	32	33001	033001		3000
%SR1	256	34001	034001		4000
%AQ1	512	40001	040001	Read Holding Registers (3) Load Register (6) Load Multiple Registers (16)	0
%R1	2488	40513	040513		0
%R1	2048	43001	043001		3000
%AQG1	32	46001	046001		6000
%R1	9999	--	410001		10000

EXL10 – Modbus Mapping					
OCS Ref.	Max. Range	Trad. Modbus Ref. (5 digits)	Expanded Modbus Ref. (6 digits)	Modbus Command(s)	Modbus Offset
%I1	2048	10001	010001	Read Input Status (2)	0
%IG1	256	13001	013001		3000
%S1	256	14001	014001		4000
%K1	7	15001	015001		5000
%Q1	2048	00001	000001	Read Coil Status (1) Force Coil (5) Force Multiple Coils (15)	0
%M1	2048	03001	003001		3000
%T1	2048	06001	006001		6000
%QG1	256	09001	009001		9000
%A1	512	30001	030001	Read Input Register (4)	0
%AIG1	32	33001	033001		3000
%SR1	256	34001	034001		4000
%AQ1	512	40001	040001	Read Holding Registers (3) Load Register (6) Load Multiple Registers (16)	0
%R1	2488	40513	040513		0
%R1	2048	43001	043001		3000
%AQG1	32	46001	046001		6000
%R1	9999	--	410001		10000

X5 Series – Modbus Mapping					
OCS Ref.	Max. Range	Trad. Modbus Ref. (5 digits)	Expanded Modbus Ref. (6 digits)	Modbus Command(s)	Modbus Offset
%I1	2048	10001	010001	Read Input Status (2)	0
%IG1	256	13001	013001		3000
%S1	256	14001	014001		4000
%K1	4	15001	015001		5000
%Q1	2048	00001	000001	Read Coil Status (1) Force Coil (5) Force Multiple Coils (15)	0
%M1	2048	03001	003001		3000
%T1	2048	06001	006001		6000
%QG1	256	09001	009001		9000
%A1	512	30001	030001	Read Input Register (4)	0
%AIG1	32	33001	033001		3000
%SR1	256	34001	034001		4000
%AQ1	512	40001	040001	Read Holding Registers (3) Load Register (6) Load Multiple Registers (16)	0
%R1	2488	40513	040513		0
%R1	2048	43001	043001		3000
%AQG1	32	46001	046001		6000
%R1	8192	--	410001		10000

X2, X4 & X7 – Modbus Mapping					
OCS Ref.	Max. Range	Trad. Modbus Ref. (5 digits)	Expanded Modbus Ref. (6 digits)	Modbus Command(s)	Modbus Offset
%I1	1024	10001	010001	Read Input Status (2)	0
%IG1	256	13001	013001		3000
%S1	256	14001	014001		4000
%K1	10	15001	015001		5000
%Q1	1024	00001	000001	Read Coil Status (1) Force Coil (5) Force Multiple Coils (15)	0
%M1	1024	03001	003001		3000
%T1	1024	06001	006001		6000
%QG1	256	09001	009001		9000
%A1	256	30001	030001	Read Input Register (4)	0
%AIG1	32	33001	033001		3000
%SR1	200	34001	034001		4000
%AQ1	256	40001	040001	Read Holding Registers (3) Load Register (6) Load Multiple Registers (16)	0
%R1	2488	40513	040513		0
%R1	2048	43001	043001		3000
%AQG1	32	46001	046001		6000
%R1	5000	--	410001		10000

RCC Series – Modbus Mapping					
OCS Ref.	Max. Range	Trad. Modbus Ref. (5 digits)	Expanded Modbus Ref. (6 digits)	Modbus Command(s)	Modbus Offset
%I1	2048	10001	010001	Read Input Status (2)	0
%IG1	64	13001	013001		3000
%S1	13	14001	014001		4000
%K1	0	15001	015001		5000
%Q1	2048	00001	000001	Read Coil Status (1) Force Coil (5) Force Multiple Coils (15)	0
%M1	2048	03001	003001		3000
%T1	2048	06001	006001		6000
%QG1	64	09001	009001		9000
%A1	512	30001	030001	Read Input Register (4)	0
%AIG1	32	33001	033001		3000
%SR1	200	34001	034001		4000
%AQ1	512	40001	040001	Read Holding Registers (3) Load Register (6) Load Multiple Registers (16)	0
%R1	2488	40513	040513		0
%R1	2048	43001	043001		3000
%AQG1	32	46001	046001		6000
%R1	4096	--	410001		10000

Section 9.2: Configuring Modbus Client Device

The following are the steps for configuring Modbus Client with Cscape.

Open **Controller → Hardware Configuration** (select a series and device type)→ **LAN1/ETN300 Config** to open the LAN1/Ethernet dialog and select Controller with Ethernet.

Open Program → Protocol Conf and perform the following configurations:

a. Select Modbus Client from the Ethernet drop-down list. Then click on the Network button. This will open the Network Config (Modbus Client) dialog.

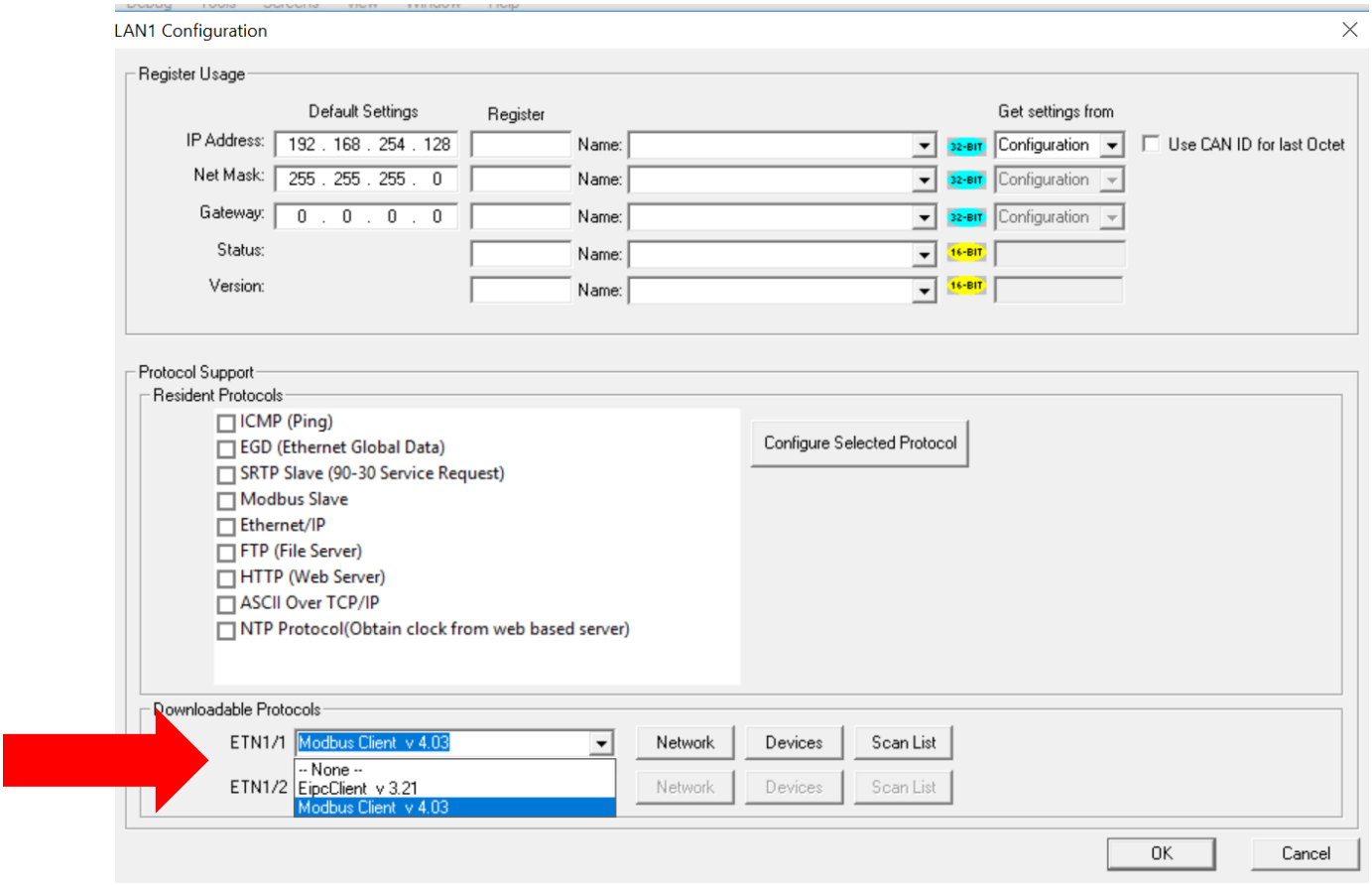


Figure 9.1 – Accessing Modbus Dialog

Section 9.3. Network Config

Network parameters and status registers can be configured in the Network Config dialog.

- Configure minimum and maximum value for port ID's as 1024 and 4096 respectively.
- Configure the update Scan type as automatic or manual according to process requirements.
- Configure the (optional) network status register.

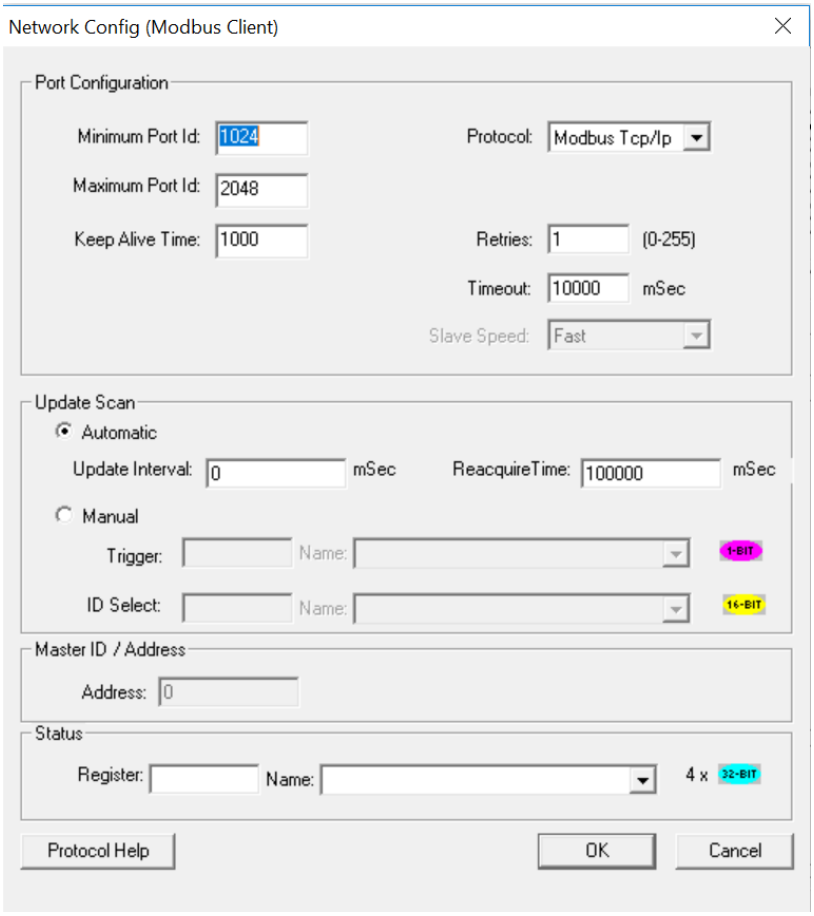
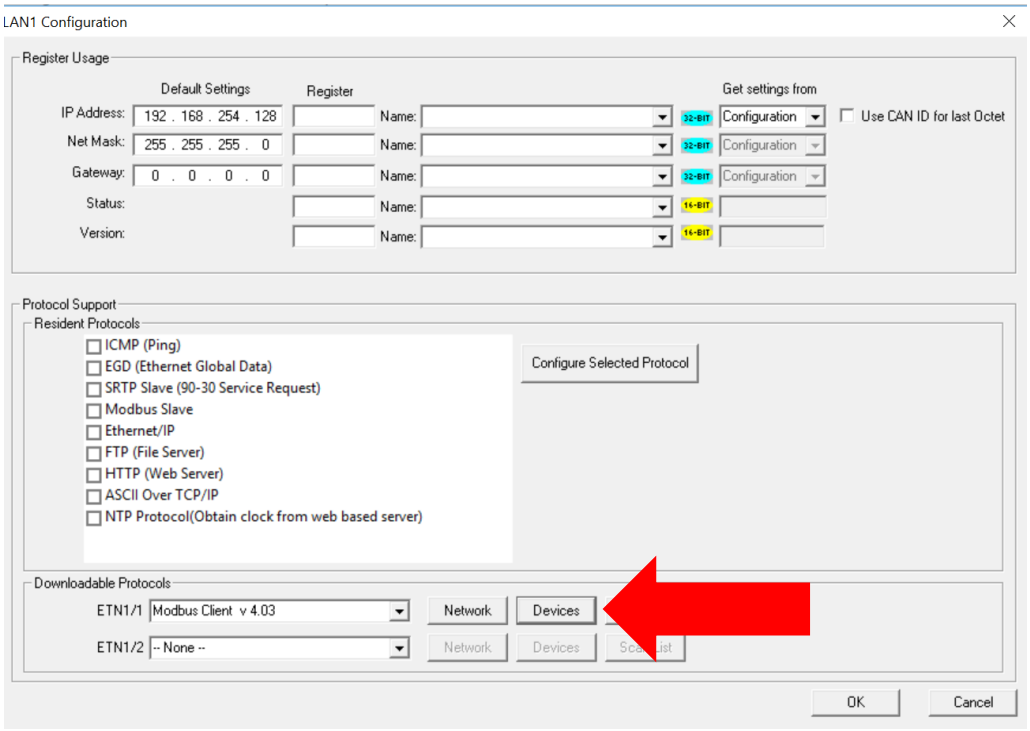


Figure 9.2 – Network Config (Modbus Client)

Section 9.4 Device Config

Click on **Devices** button to configure Server device (max 64) as follows:



Select a device:

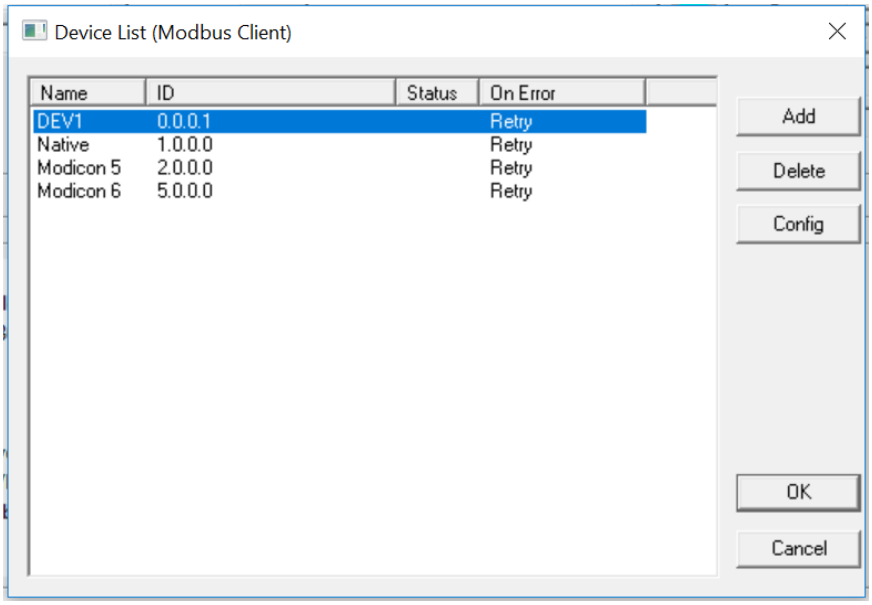


Figure 9.3 – Device List (Modbus Client)

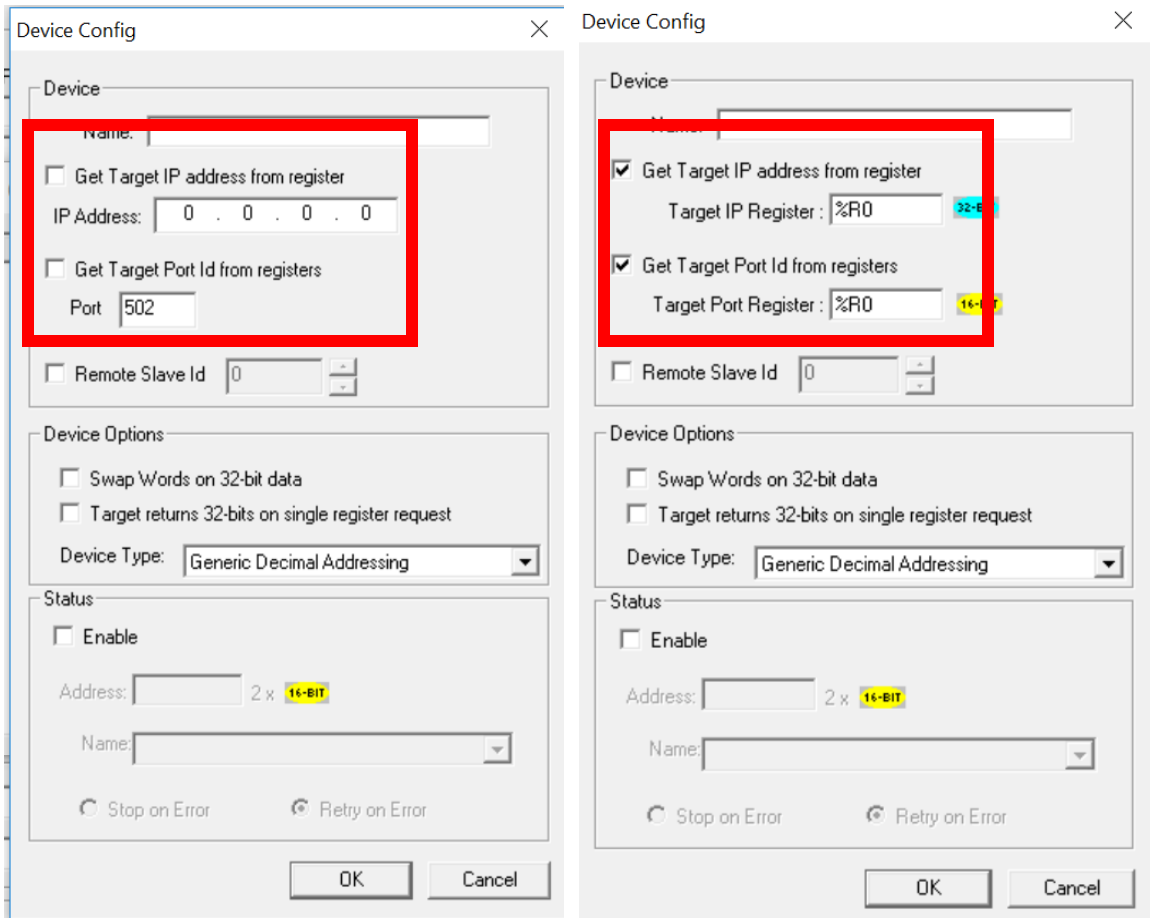


Figure 9.4 – Device Config

- Configure server device name, IP address (either direct or through registers) and port ID (either direct or through registers). Configuring IP address or port ID through registers allows user to modify IP address or port ID during runtime.

  NOTE: User should change the controller mode to Idle and then to Run after modifying port ID during runtime.

- Select the address mode (refer to Section 9.2) required when talking to this device.
- If option for Swap Words on 32-bit Data is checked, the high and low 16-bit values of 32-bit Data are swapped when transferred between the target and OCS.
- Device status registers can be optionally enabled and used to determine the current state of communications.
- Select Stop on Error as per process requirement (located in the Status box at the bottom).

Device Type: Modbus TcpIpClient.Dll supports the following addressing modes:

- Generic Decimal Addressing
- Generic Hexadecimal Addressing
- Native Addressing
- Modicon PLC 5-Digit Addressing
- Modicon PLC 6-Digit Addressing

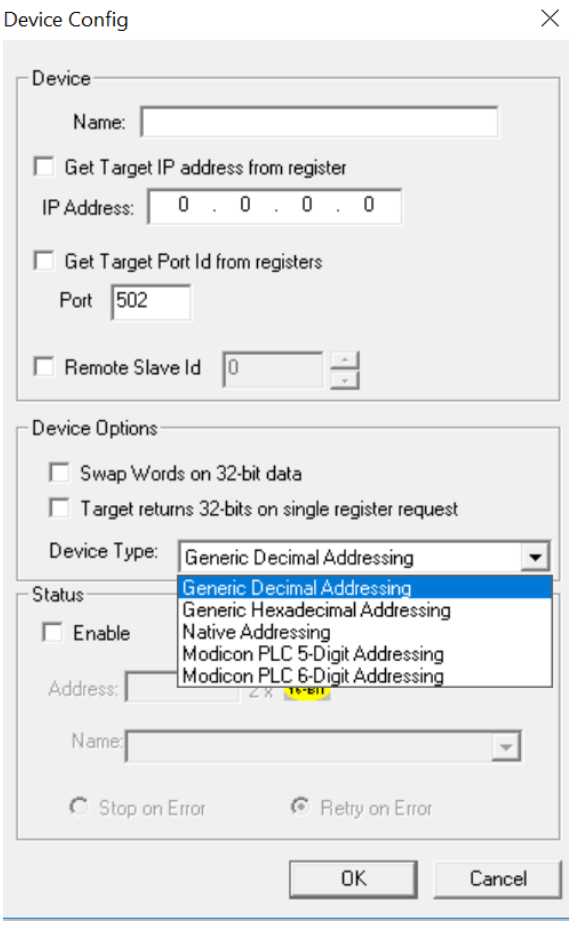
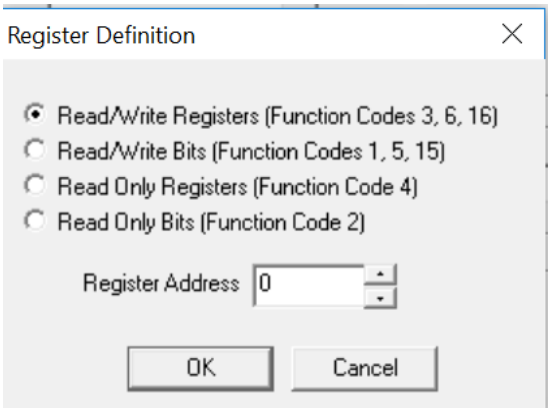


Figure 9.5 – Device Type

Generic Decimal Addressing

The first option is **Generic Decimal Addressing**. Use this when the connected device uses an addressing mode which does not correspond to a Modicon PLC, and offsets in the documentation for the target device are given in decimal.



Addresses are entered in the following form:

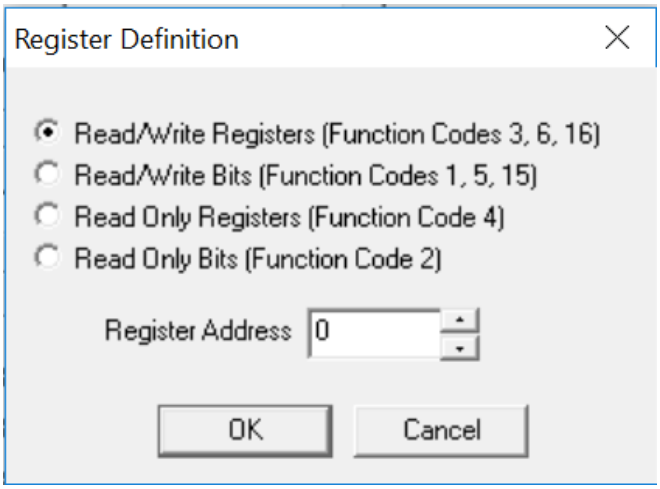
<Data Type> <Decimal number>

Data type takes one of the following values:
RWR : (Read/Write Registers) Uses Modbus function codes 3, 6, 16
RWB : (Read/Write Bits) Uses Modbus function codes 1, 5, 15
ROR : (Read Only Registers) Uses Modbus function code 4
ROB : (Read Only Bits) Uses Modbus function code 2

The decimal number gives the offset of the target data in the selected data type.

Generic Hexadecimal Addressing

Use this when the connected device uses an addressing mode which does not correspond to a modicon PLC, and offsets in the documentation for the target device are given in hexadecimal



Addresses are entering the in the following form:

<Data Type> <Decimal number>

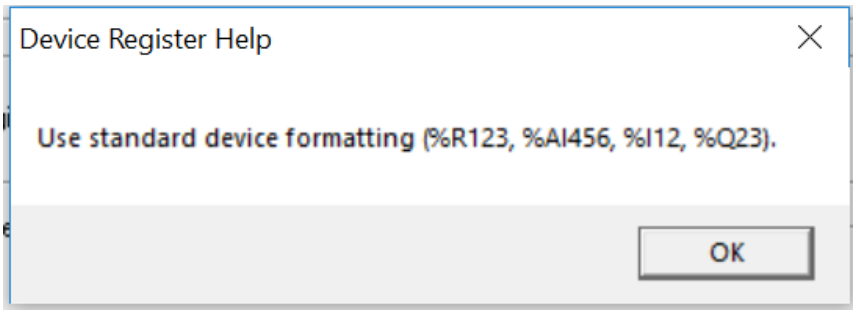
Data type takes one of the following values

RWR : (Read/Write Registers) Uses Modbus function codes 3, 6, 16
RWB : (Read/Write Bits) Uses Modbus function codes 1, 5, 15
ROR : (Read Only Registers) Uses Modbus function code 4
ROB : (Read Only Bits) Uses Modbus function code 2

The hexadecimal number gives the offset of the target data in the selected data type.

Native Addressing

Use this when the connected device is an OCS unit running Modbus TCP / IP host.



Addresses are entered as standard OCS addresses.

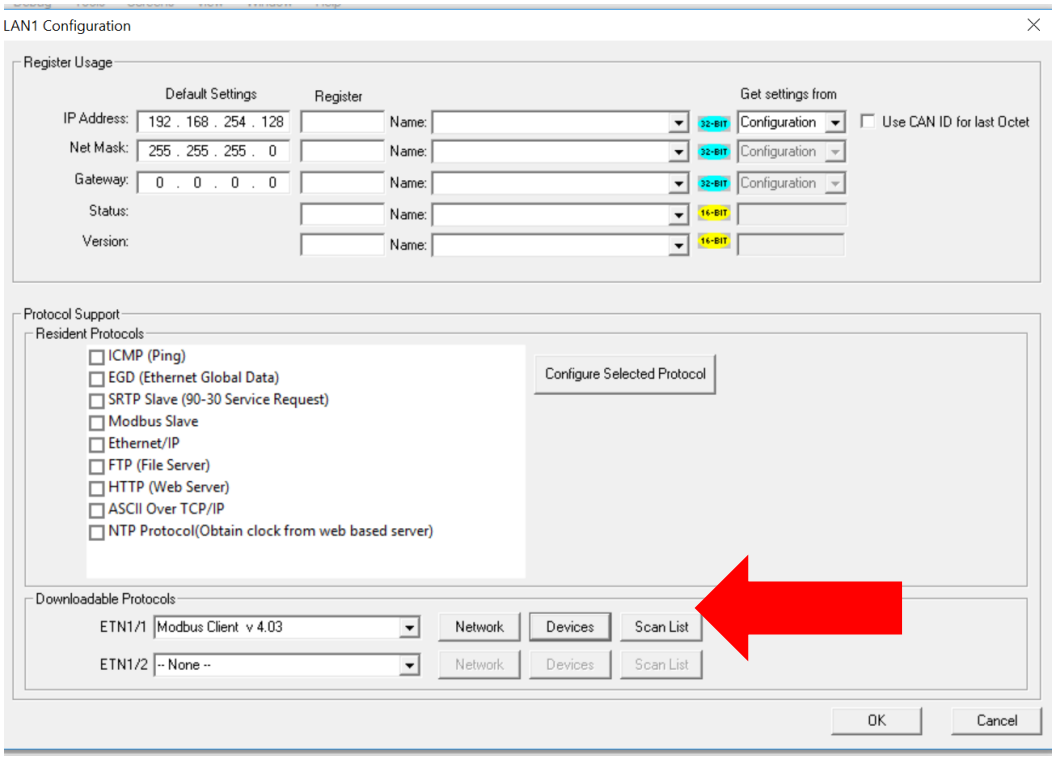
Modicon PLC 5-Digit Addressing

Use this when the connected device uses an addressing mode which corresponds to a reduced memory map modicon PLC.

Modicon PLC 6-Digit Addressing

Use this when the connected device uses an addressing mod which corresponds to an extended memory map modicon PLC.

Section 9.5 Scan List



Click on Scan List button to open the Scan List (Modbus Client) dialog.

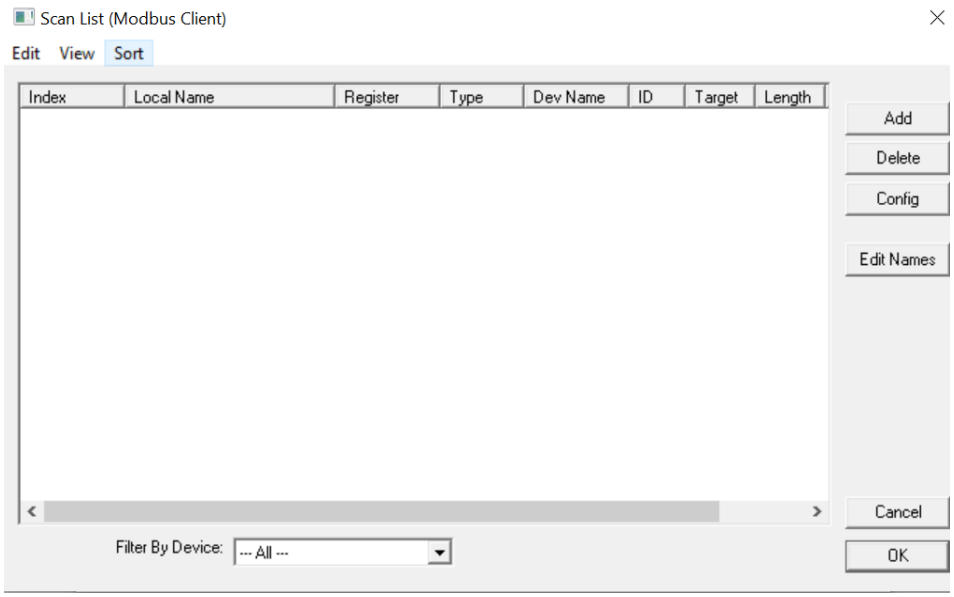


Figure 9.6 – Scan List (Modbus Client)

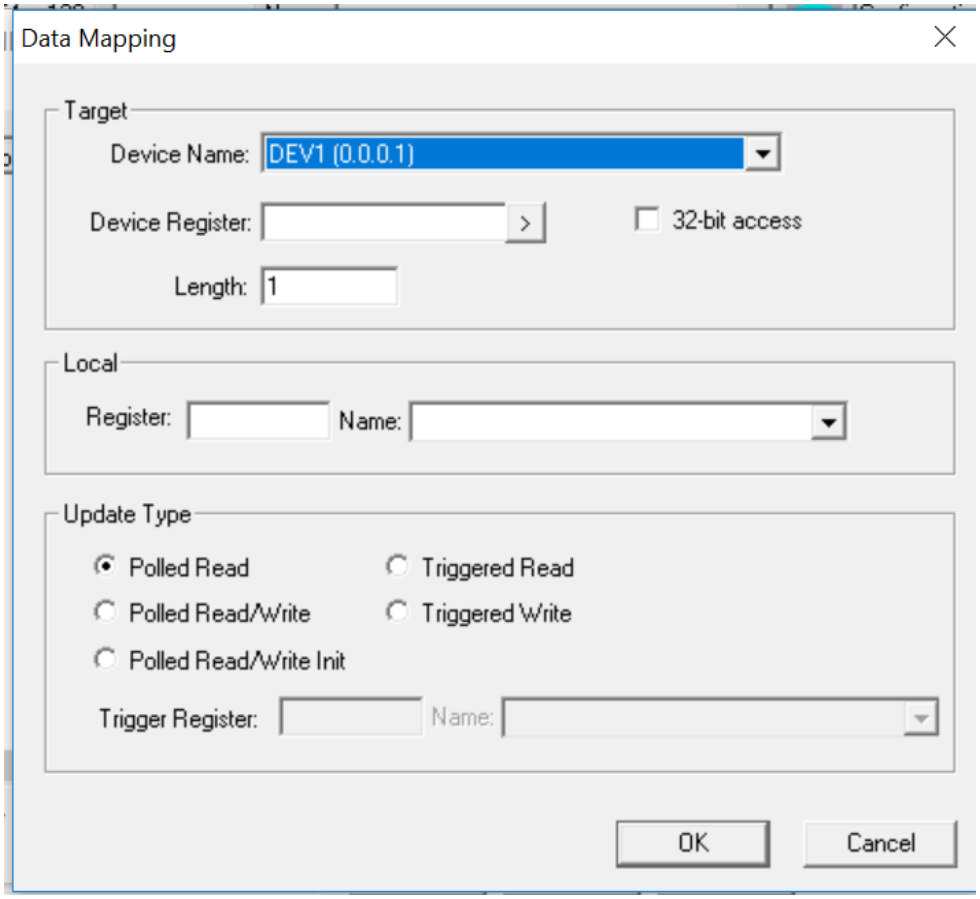


Figure 9.7 – Data Mapping

- To transfer data between the OCS and remote target, a Scan List must be created that defines each transaction. Each mapping entry (transaction) contains the source and destination registers, the number of consecutive registers transferred, the direction of the transfer and trigger for transfer (optional). Maximum number of entries is 512.

  NOTE: Order of the Scan List is the order in which the transaction occurs.

- Scan list for a client device is developed in the following steps:

  Click **ADD** button to open the Data Mapping dialog for adding new entry to the Scan List.

  Configure the following:

Table 9.1 – Data Mapping Fields	
Device Name	Select the target device from the drop-down list. Only those device entries previously created from the Device Config menu are available.
Device Register	Specify the target device's register to be mapped according to the selected addressing mode. (Section 9.2)
32-Bit Access	Allows two local (OCS) 16-bit registers to be treated as a single 32-bit value. For example, if the value in either 16-bit register is modified, both registers are written to the device.
Length	Specify the number of consecutive device registers to Read/Write. Maximum length is 32.
Local Register	Specify the local (OCS) register that is the source or destination for transfer of the data.
Local Name	(Optional) Enter/Select name for the **Local Register.**
Update Type	This field specifies the direction and triggers the transfer of data between the OCS and target device. Select **Update Type** as per the system requirements.

PART FOUR: NON-INDUSTRIAL ETHERNET PROTOCOLS

CHAPTER 10: INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

10.1 ICMP Overview

ICMP is used for diagnostic purposes only, to determine if another device exists on the Ethernet network.

Using ICMP, the OCS sends Ping Echo Requests to another device and expects the other device to answer with Ping Echo Responses. The OCS measures the round-trip time of each Ping Echo Request / Response exchange and puts the result (in milliseconds) into an OCS register.

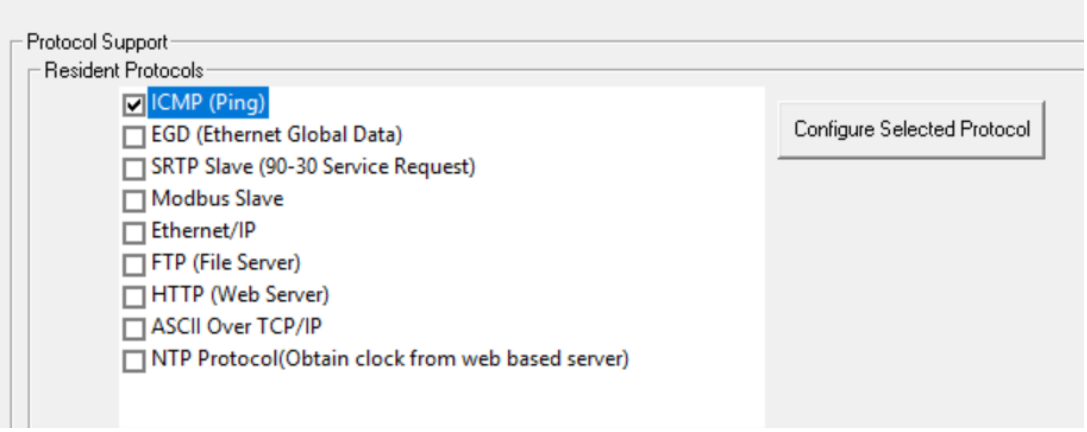
In addition, when the OCS receives a Ping Echo Request from another device, it answers with a Ping Echo Response.

NOTE: Although ICMP supports other network diagnostics, the OCS only supports *Ping*.

10.2 ICMP Configuration

If ICMP protocol will be used in the application, ICMP Configuration must be performed, in addition to the general Ethernet LAN1 Configuration previously described in Chapter 3. To configure ICMP protocol, use Cscape Programming Software to perform the following five steps:

1. Open the Ethernet LAN1 Configuration dialog by selecting **Controller → Hardware Configuration** (select a series and device type)→ **LAN1/Config**.



2. Enable ICMP by checking the **ICMP (Ping)** checkbox in the LAN1 Configuration dialog.

3. Click on the **Configure Selected Protocol** button next to the **ICMP (Ping)** checkbox to open the ICMP Configuration dialog. Refer to figure below.

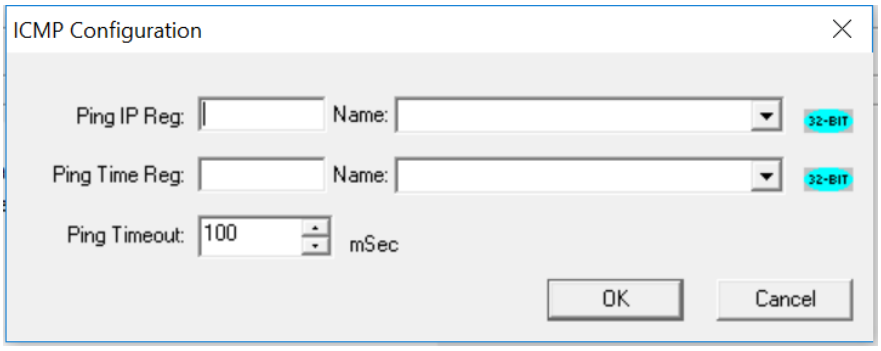


Figure 10.1 – ICMP Configuration

4. Set up the ICMP Configuration parameters as follows:

Table 10.1 – ICMP Configuration Parameters	
Ping IP Reg	Enter an OCS Register reference (such as %R200) to indicate which 32-bit OCS register will be read to obtain the IP Address of the network device to send Ping Echo Requests to.
Ping Time Reg	Enter an OCS Register reference (such as %R202) to indicate which 32-bit DINT OCS register will be written with the Ping Echo Request / Response round-trip time (in milliseconds). If no response is received within the Ping Timeout interval, a Ping Time of -1 is reported, indicated a timeout.
Ping Timeout	Enter a number between 100 and 100,000 for how often (in milliseconds) the OCS should send Ping Echo Requests. If the response takes longer than this amount of time, it is considered a timeout.

5. Click **OK** to accept the new ICMP Configuration.

10.3 ICMP Operation

To start Ping Echo Requests, use Ladder Code, User Screens, or Cscape Data Watch to write an IP Address to the OCS register indicated by **Ping IP Reg**. To check the resulting Ping Echo Response times, read the OCS register indicated by **Ping Time Reg**. If the round-trip time exceeds the configured **Ping Timeout** (or if there is no response at all), the reported response time will be -1.

To stop Ping Echo Requests, write 0.0.0.0 to the OCS register indicated by **Ping IP Reg**.

NOTE: ICMP protocol is active even when the OCS is not in RUN mode. Therefore, when the OCS is not in RUN mode, User Screens and Cscape Data Watch can still be used to control and monitor ICMP Pinging.

CHAPTER 11: FTP SERVER PROTOCOL

11.1 FTP Overview

File Transfer Protocol (FTP) is a standard Client/Server Internet protocol, based on RFC959, which supports efficient and reliable file transfers over a TCP/IP network. In this context, the OCS acts as an FTP Server, which responds to file transfer requests from one or more FTP Clients.

NOTE: FTP protocol is supported only by OCS Models, which have built-in Ethernet *and* an OCS file system with removable media (such as Compact Flash / microSD). For OCS Models that do not have an OCS file system, the **FTP (File Transfer)** checkbox will be grayed out in Cscape's Ethernet LAN1 Configuration dialog.

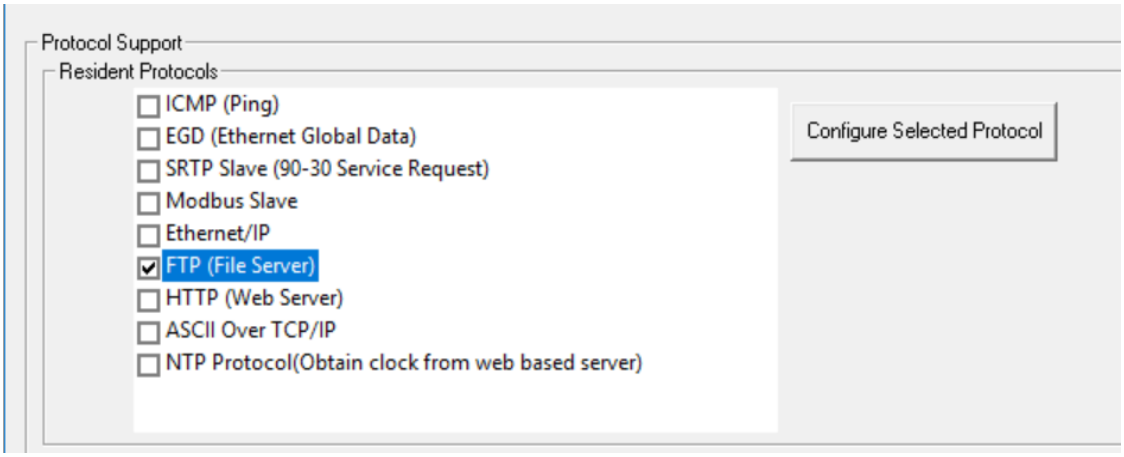
The OCS's FTP Server supports both anonymous and authenticated file transfers between an FTP Client and the OCS file system. Authenticated file transfers require that the FTP Client provide one of two possible User Name and Password pairs, before the OCS file system can be accessed.

Properly authenticated, an FTP Client can access OCS file system functions, which include file read, file create, file delete, file rename, file write, directory read, directory create, and directory delete.

11.2 FTP Configuration

If FTP File Transfer will be used in the application, FTP Configuration must be performed, in addition to the general Ethernet LAN1 Configuration previously described in Chapter 3. To configure FTP protocol, use Cscape Programming Software to perform the following five steps:

1. Open the Ethernet LAN1 Configuration dialog by selecting **Controller → Hardware Configuration** (select a series and device type)→ **LAN1/Config.**



2. Enable FTP by checking the **FTP (File Server)** checkbox in the LAN1 Configuration dialog Until this is done, the OCS will not respond to any FTP Client requests.

3. Click on the **Configure Selected Protocol** button next to the **FTP (File Server)** checkbox to open the FTP Configuration dialog. Refer to figure below.

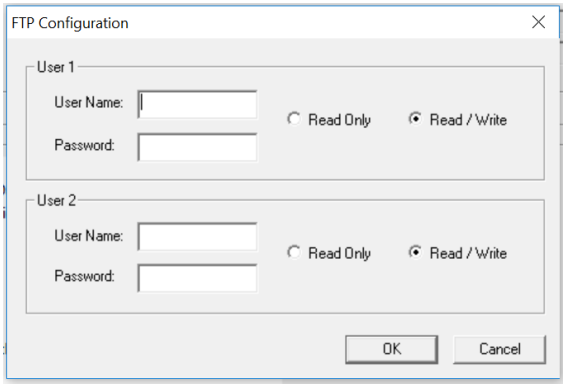


Figure 11.1 – FTP Configuration Dialog

4. Optionally, set up the FTP Configuration parameters for **User 1** and/or **User 2** as follows:

Table 11.1 – FTP Configuration Parameters	
User Name	Enter an alphanumeric string (up to 40 characters) for the FTP Client's User Name
Password	Enter an alphanumeric string (up to 40 characters) for the FTP Client's Password.
Read Only	Select this option to give the FTP Client read-only access to the OCS file system.
Read / Write	Select this option to give the FTP Client read-write access to the OCS file system.

NOTE: User Names and Passwords are case-sensitive and, by default, User Name and Passwords for both User 1 and User 2 are empty. In this case, *anonymous* FTP Clients can access OCS files, but file access will be read-only. In a typical application, both User 1 and User 2 should be configured - one with read-only access and the other with read-write access.

5. Click **OK** to accept the new FTP protocol configuration.

11.3 FTP Operation

After performing OCS FTP Configuration, a 3rd party FTP Client (such as a PC running SmartFTP) can be used to access the OCS file system.

The target OCS's configured IP Address, FTP User Name (if any), and FTP Password (if any), must be entered into the FTP Client, to establish an FTP connection before any file accesses can be initiated. Please refer to documentation provided with the 3rd party FTP Client, regarding how to install, configure and operate the FTP Client.

When configuring an FTP Client, the user should be aware that the FTP protocol standard defines many optional features. Like many UNIX implementations, the OCS FTP Server imposes constraints on some of these options, as shown in Table 13.1 below.

Table 11.2 –FTP Server Option Constraints		
FTP Option	**Supported**	**Not Supported**
File Transfer Type	Binary Type	ASCII Type & EBCDIC Type
File Format	Non-Print Format	Telnet Format & Carriage Control Format
File Structure	File Structure	Record Structure & Page Structure
Transmission Mode	Stream Mode	Block Mode & Compressed Mode
Data Connection Mode	Active (PORT) Mode	Passive (PASV) Mode

In addition, since the FTP Server will automatically disconnect an FTP Client after about 3 minutes of inactivity, the user might need to configure the FTP Client to periodically send NOOP (No Operation) commands to the FTP Server, to keep the connection alive.

11.4 FTP File Accessing

When using FTP to exchange files with an OCS, the user should know that the OCS file system implements an "8.3" filename format, which means all file and directory names should consist of up to 8 characters, followed by an optional dot, and an optional extension with up to 3 characters.

Also, the OCS file system allows multiple concurrent file accessing. For example, an FTP Client can read a file at the same time the OCS ladder program is logging data to another file. It is also possible for both the FTP Client and OCS ladder functions to read the same file at the same time.

If there is a file access conflict, such as an FTP Client attempting to delete a file that is currently being read or written by ladder, the file delete request is denied and an error response is sent to the FTP Client.

CHAPTER 12: HTTP SERVER PROTOCOL

12.1 HTTP Overview

NOTE: The HTTP functionality covered in this chapter is not the same as the WebMI functionality also available for many OCS products. The HTTP protocol section should not be checked if WebMI is to be used. Refer to Chapter 15 for WebMI functionality.

Hypertext Transfer Protocol (HTTP) is a standard Client/Server Internet protocol, based on RFC1945, which transfers web content over a TCP/IP network. In this context, the OCS acts as an HTTP Server (web content provider), which responds to requests from HTTP Clients (web browsers).

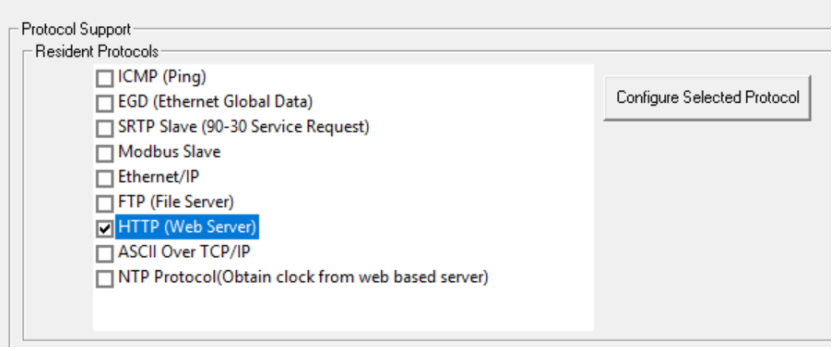
The OCS's HTTP Server supports optional HTTP Client authentication, requiring the web browser operator to enter a User Name and Password, before web content can be accessed.

Properly authenticated, an HTTP Client can access web content previously stored in the OCS file system. This web content can include standard static HTML files and other web resource files. In addition, the HTTP Server allows the HTTP Client to display and edit OCS register data, by embedding specially named dynamic HTML files (**readreg.htm** and **writereg.htm**) into a static HTML file.

12.2 HTTP Configuration

If the HTTP Web Server will be used in the application, HTTP Configuration must be performed, in addition to the general Ethernet LAN1 Configuration previously described in Chapter 3. To configure HTTP protocol, use Cscape Programming Software to perform the following five steps:

1. Open the Ethernet LAN1 Configuration dialog by selecting **Controller → Hardware Configuration** (select a series and device type)→ L**AN1/Config.**



2. Enable HTTP by checking the **HTTP (Web Server)** checkbox in the LAN1 Configuration dialog Until this is done, the OCS will not respond to any HTTP Client requests.

3. Click on the **Configure Selected Protocol** button next to the **HTTP (Web Server)** checkbox to open the HTTP Configuration dialog. Refer to figure below.

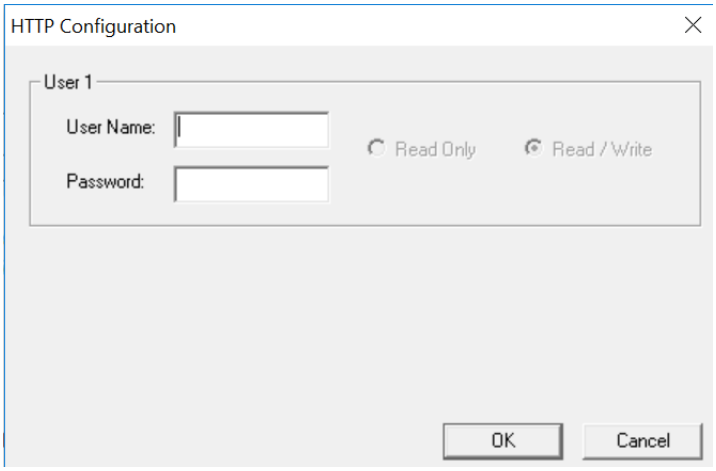


Figure 12.1 – HTTP Configuration Dialog

4. Optionally, set up the HTTP Configuration parameters for **User 1** as follows:

Table 12.1 – HTTP Configuration Parameters	
User Name	Enter an alphanumeric string (up to 40 characters) for the HTTP Client's User Name
Password	Enter an alphanumeric string (up to 40 characters) for the HTTP Client's Password.
Read Only	This option is always deselected and grayed out.
Read / Write	This option is always selected and grayed out.

NOTE: User Names and Passwords are case sensitive, and, by default, the User Name and Password are empty. In this case, all HTTP Clients can access the OCS web content. In a typical application, the User Name and Password should be configured for security purposes.

5. Click **OK** to accept the new HTTP protocol configuration.

12.3 HTTP Operation

After performing HTTP Configuration, as described in section 11.2, a 3rd party HTTP Client (such as a PC running Internet Explorer or Firefox) can be used to browse OCS web content.

To start browsing web resources stored in the OCS file system, enter the appropriate URL (Uniform Resource Locator) into the HTTP Client (web browser). As a minimum, the URL should include the target OCS's IP Address and a filename. For example, if the OCS's IP address is 192.168.254.128, the following URL will access the **index.htm** file in the OCS file system root directory:

http://192.168.254.128/index.htm

NOTE: The HTTP Server does not automatically default to **index.htm,** if no file is specified in the URL.

If the OCS was configured with a User Name and Password, the browser will prompt for the User Name and Password to be entered, before opening the web page. Please refer to documentation provided with the 3rd party HTTP Client, regarding how to install, configure and operate the HTTP Client.

The OCS HTTP Server implements HTTP/1.0 and does not support persistent connections. Therefore, each time a web page (or other resource) is transferred, the following sequence takes place:

1. The HTTP Client (web browser) opens a connection with the HTTP Server and sends a request.
2. The HTTP Server sends a response to the HTTP Client and then closes the connection.

Since the connection only stays open long enough to complete one request, multiple HTTP Clients can concurrently access OCS web content, even though they are sharing a single connection.

12.4 Web Content - Storing

When storing web content files in the OCS file system, the user should know that the OCS file system implements an "8.3" filename format, which means all file and directory names must consist of up to 8 characters, followed by an optional dot, and an optional extension with up to 3 characters.

Web content files can be stored into the OCS file system media (such as Compact Flash / microSD), by temporarily plugging the media into a PC with an installed card reader, or via Ethernet using FTP (File Transfer Protocol). See Chapter 10 for information on using the OCS FTP Server.

To keep the web content files organized, multiple files, directories and subdirectories can be created on the OCS file system media as desired. Normally, it is a good practice to store an **index.htm** file in the root directory as a "home page", but this is not required.

12.5 Web Content - Dynamic Data

An OCS HTML file is either *static* or *dynamic*. The application engineer creates a static HTML file and stores it on the OCS file system media. Because of special markups in the static HTML file, the HTTP Server creates dynamic HTML files to access OCS Registers. This technique allows a static HTML file to read and write OCS registers dynamically.

12.5.1 STATIC HTML FILES

Custom HTML files are created to suit the application. These files can be created with a simple text editor or by using a 3rd party HTML Editor.

Within the static HTML file, <iframe> and <form> tags respectively read and write OCS Register data, by referencing two specially named dynamic HTML files (**readreg.htm** and **writereg.htm**).

12.5.2 DYNAMIC HTML FILES – READING OCS REGISTER DATA (READREG.HTM)

OCS Register data is read by placing an <iframe> tag, that references **readreg.htm**, within a static HTML file. The following example <iframe> tag reads and displays OCS Register %R11 as an Unsigned Integer every two seconds:

```
<iframe src="readreg.htm;reg-R11;fmt-UINT;rfs-2"></iframe>
```

The required **src** attribute is a quoted string containing the **readreg.htm** reference and two or three parameters, all separated by semicolons. The **reg, fmt** and **rfs** parameters specify the OCS Register to read, the display format to use and optionally how often to refresh the data. These parameters are described in more detail in the following 3 sections:

12.5.2.1 REG PARAMETER – OCS REGISTER READ REFERENCE

Examples: reg-R11
reg-R11.3

The required **reg** parameter is followed by a dash and an OCS Register reference. Register reference numbers start at 1 and the maximum depends on the register type and OCS type, as show in the following table:

Table 12.2 – OCS Register Read References		
OCS Register Type	Maximum Reference (ETN200)	Maximum Reference (ETN300)
%R	R2048	R9999
%AI, %AQ	AI512, AQ512	AI512, AQ512
%I, %Q, %M, %T	I2048, Q2048, M2048, T2048	I2048, Q2048, M2048, T2048
%S	S16	S16

- For BOOL formatting (see next section), a bit number can be appended to 16-bit register references (%R, %AI and %AQ). For example, **reg-R11.3** references bit 3 of register %R11. Valid bit numbers are 1 through 16.

12.5.2.2 fmt Parameter – OCS Register Read Format

Examples: fmt-BOOL
fmt-BOOL-ON-OFF
fmt-BOOL-YES-NO
fmt-BIN
fmt-HEX
fmt-INT
fmt-UINT
fmt-DINT
fmt-UDINT
fmt-REAL
fmt-REAL-E
fmt-ASCII-10
fmt-IPADR

The required **fmt** parameter is followed by a dash and a display format type. Valid format types are **BOOL**, **BIN**, **HEX**, **INT**, **UINT**, **DINT**, **UDINT**, **REAL**, **ASCII** and **IPADR**. Three of these format types (BOOL, REAL and ASCII) have optional or required modifiers as follows:

The **BOOL** format reads and displays a single bit, and two optional modifiers indicate the text to display for the bit's logical On and Off states. The text "TRUE" and "FALSE" will be displayed if the modifiers are omitted. In other words, **fmt-BOOL** is equivalent to **fmt-BOOL-TRUE-FALSE**.

The **REAL** format reads a 32-bit floating-point value, and the optional **-E** modifier can be used to display data in exponential (scientific) notation. The **fmt-REAL** format displays a variable number of digits to the left of the decimal point and 6 digits to the right of the decimal point (e.g. -12345.000000). On the other hand, the **fmt-REAL-E** format only shows 1 digit to the left of the decimal point and shows 6 digits to the right of the decimal point followed by a 4-character power of 10 (e.g. -1.234500E+04).

The **ASCII** format accesses a string of ASCII characters, and a string size modifier from **-1** to **-31** is required, indicating how many ASCII characters to read and display. For example, the **fmt-ASCII-10** format reads and displays a string of 10 ASCII characters.

12.5.2.3 rfs Parameter - OCS Register Read Auto-Refresh

Example: rfs-2

The optional **rfs** parameter specifies how often the <iframe> should automatically refresh the data on the browser screen. For example, **rfs-2** will cause the data to be refreshed every 2 seconds.

The number of seconds specified can range from 0 to the limits of the browser. A refresh time of 0 will refresh the data as often as possible, but this is not recommended because of the volume of Ethernet traffic it can create. If the optional **rfs** parameter is omitted, the data will only be refreshed when the user manually refreshes their web browser screen.

The **rfs** parameter uses a technique called "client-pull" to tell the web browser to do the periodic refresh. Not all web browsers support "client pull" and those that do support it can usually be configured to disable it for security purposes.

NOTE: When using an **Internet Explorer** browser to view OCS web pages that employ the rfs parameter, the browser plays the currently configured **Start Navigation** sound (a short "tick" by default) for each refresh. To disable the "ticking", (1) open the **Windows Control Panel**, (2) click the **Sounds** or **Sound and Audio Devices** icon or link and (3) turn off the **Start Navigation** sound by selecting **(None)** for it in the drop-down list.

12.5.2.4 fnt Parameter – OCS Register Read Font

Example: fnt-5-arial-blue

The optional **fnt** parameter specifies the font size, typeface and color to be used when OCS Register data is displayed on the browser screen.

All 3 font attributes (size, typeface and color) are optional, but must be specified in the order shown. For example, **fnt-5-arial** specifies a size 5, Arial font, but will use the browser's default text color.

If the size or typeface is omitted, its *dash* character should still be used as a *placeholder*. For example, **fnt---blue** specifies that the displayed text will be blue, but the browser's default font size and typeface will be used.

NOTE: In this last example, there are 3 dashes between **fnt** and **blue**.

For the 1st attribute (font size), a number from **1** to **7** can be used, with **1** as the smallest and **7** as the largest. This is a browser limitation, as browsers typically support only 7 font sizes. Optionally, a point size can be specified, such as **12pt**, but the browser will translate this into one of its supported font sizes.

For the 2nd attribute (font typeface), any typeface name supported by the browser can be used, such as **Arial**, **Times New Roman** or **Courier New**.

For the 3rd attribute (font color), the 16 standard HTML color names should always work. These 16 standard colors are **aqua**, **black**, **blue**, **fuchsia**, **gray**, **green**, **lime**, **maroon**, **navy**, **olive**, **purple**, **red**, **silver**, **teal**, **white** and **yellow**.

12.5.3 DYNAMIC HTML FILES – WRITING OCS REGISTER DATA (WRITEREG.HTM)

OCS Register data is written by placing a <form> tag, that references **writereg.htm**, within a static HTML file. The following example <form> tag sets up an Unsigned Integer write to OCS Register %R11:

```
<form action="writereg.htm;reg-R11;fmt-UINT" method="post">...</form>
```

Between the <form> and </form> tags, there should be one or more <input> tags that enable a browser operator to enter or select data and write it to the register.

For example, in the following complete <form> markup, the first <input> tag creates a text input box, containing a default value of 0, which allows the user to enter a new value for %R11. The second <input> tag creates a **Send** button, which can be clicked to submit the new value (write it to the register):

```
<form action="writereg.htm;reg-R11;fmt-UINT" method="post">
    <input type="text" name="reg" value="0">
    <input type="submit" value="Send">
</form>
```

In the following example, the first <input> tag creates an **ON** button that can be clicked to turn %T22 On, while the second <input> tag creates an **OFF** button that can be clicked to turn %T22 Off.

```
<form action="writereg.htm;reg-T22;fmt-BOOL-ON-OFF" method="post">
    <input type="submit" name="reg" value="ON">
    <input type="submit" name="reg" value="OFF">
</form>
```

Important: In the above two examples, the OCS HTTP Server expects the **name="reg"** attributes to appear exactly as shown.

The <form> tag's required **action** attribute is a quoted string containing the **writereg.htm** reference and two parameters, all separated by semicolons. The **reg** and **fmt** parameters specify the OCS Register to write and the data entry/select format to use. These parameters are described in more detail in the following 2 sections:

12.5.3.1 reg Parameter – OCS Register Write Reference

Examples: reg-R11
reg-R11.3

The required **reg** parameter is followed by a dash and an OCS Register reference. Register references start at 1 and the maximum reference depends on the register type and OCS type, as show in the following table:

Table 12.3 – OCS Register Write References

OCS Register Type	Maximum Reference (ETN200)	Maximum Reference (ETN300)
%R	R2048	R9999
%AI, %AQ	AI512, AQ512	AI512, AQ512
%I, %Q, %M, %T	I2048, Q2048, M2048, T2048	I2048, Q2048, M2048, T2048

- For BOOL formatting (see next section), a bit number can be appended to 16-bit register references (%R, %AI and %AQ). For example, **reg-R11.3** references bit 3 of register %R11. Valid bit numbers are 1 through 16.

12.5.3.2 fmt Parameter – OCS Register Write Format

Examples:
fmt-BOOL
fmt-BOOL-ON-OFF
fmt-BOOL-YES-NO
fmt-BIN
fmt-HEX
fmt-INT
fmt-UINT
fmt-DINT
fmt-UDINT
fmt-REAL
fmt-ASCII-10

The required **fmt** parameter is followed by a dash and a data entry format type. Valid format types are **BOOL**, **BIN**, **HEX**, **INT**, **UINT**, **DINT**, **UDINT**, **REAL** and **ASCII**. Three of these format types (BOOL, REAL and ASCII) have optional or required modifiers as follows:

The **BOOL** format writes a single bit, and two optional modifiers indicate the text that must be submitted to set the bit to its logical On and Off states. If these modifiers are omitted, they default to "TRUE" and "FALSE", so that **fmt-BOOL** is equivalent to **fmt-BOOL-TRUE-FALSE**.

The **REAL** format writes a 32-bit floating-point value, and data can be entered in either standard decimal notation or in scientific notation as desired, with or without the **-E** modifier. The **-E** modifier is not necessary and is ignored if present.

The **ASCII** format accesses a string of ASCII characters, and a string size modifier from **-1** to **-31** is required, indicating how many ASCII characters will be written when the form is submitted. For example, the **fmt-ASCII-10** format writes a string of 10 ASCII characters. If the user enters an ASCII string that is smaller than the specified string size, the submitted string will be padded with spaces.

12.6 Web Content - Authoring

Web content can range from very simple to extremely complex depending on application needs. As such, it is well beyond the scope of this supplement to attempt to provide a tutorial on the subject.

12.6.1 WEB SERVER EXAMPLE APPLICATIONS

One of the best ways to learn how to use the HTTP Web Server is to study working examples. OCS HTTP Web Server Examples are available for download on the Horner Automation website (**Support → HE-XEC Ethernet Utility → OCS Web Demo**.

https://hornerautomation.com/support-files/

12.6.2 WEB AUTHORING REFERENCE MATERIAL

Those new to web content development would be well advised to obtain and study one or more published books on the subject.

Crowder, David A. and Bailey, Andrew.
Creating Web Sites Bible, Third Edition.
Indianapolis, IN: Wiley Publishing, Inc., 2004.
ISBN: 0470223634

Powel, Thomas A.
HTML & XHTML: The Complete Reference, Fourth Edition.
Emeryville, CA: McGraw-Hill/Osborne, 2003.
ISBN: 0-07-222942-X

12.6.3 WEB AUTHORING TOOLS

There are several 3rd party tools available to assist in the development of web content. These tools come in four basic types: Text Editors, HTML Editors, WYSIWYG Programs, and Word Processors.

Text Editors require a thorough knowledge of HTML (HyperText Markup Language). However, many experienced web developers prefer to work using a plain text editor.

HTML Editors are specially enhanced text editors designed specifically to make web page creation easier, but they still require a great deal of HTML knowledge.

WYSIWYG (What You See Is What You Get) Programs allow HTML code creation using drag and drop techniques, and thus promise to minimize the author's need to understand HTML. However, when problems arise, there is no substitute for getting into the generated HTML code to see what is going on.

Word Processors should be avoided for web content design because they are not designed for this purpose.

Table 12.3 contains suggestions for web authoring tools:

Table 12.4 – Web Content Authoring Tools

Tool Name	Tool Type	Platform(s)	Web Site
Homesite	HTML Editor	Windows	http://www.adobe.com/
Dreamweaver	WYSIWYG Program	Windows MAC	http://www.adobe.com/

CHAPTER 13: Email (SMTP Protocol)

13.1 Overview

Email has been implemented using SMTP (Simple Mail Transfer Protocol) protocol in OCS.

An electronic mail message consists of two components, the message *header*, and the message *body*, which is the email's content. The message header contains control information, including, an originator's email address and one or more recipient addresses. Usually additional information is added, such as a subject header field.

This feature is designed to send email messages embedding real time data (logic registers) of the controllers and can be configured using Cscape.

Minimum version requirements for Email feature Firmware 12.6.

Horner ethernet email supports:

- login type authentication
- allows up to 20 logic registers to be embedded per email.
- supports up to 128 different emails to be configured.
- Supports email with attachments.

NOTE: OCS cannot receive emails.

NOTE: For SSL/TLS email servers, default port is 587 and for other non-SSL servers it is 25.

OCS Series and Email Support			
Controller(s)	**TLS 1.3 encryption supported from Firmware Ver:**	**Email Support**	**Gmail Support**
Legacy Controllers: Classic OCS, NX series, QX Series, RX Series, RCX Series, XL10e, XL6e	No	Yes	No
XLEe / XLTe	No	Yes	No
XL4e, EXL6e, XL7e, ELX10e, XL+	Yes (FW Ver 15.00 or later)	Yes	Yes
X4, X5 & X7 (Micro OCS Series)	No	Yes	No
X2, XLE, XLT (No Ethernet)	NA	NA	No
ZX Series	Yes (FW Ver 15.00 or later)	Yes	Yes

13.2 Email Configuration

NOTE: Ethernet port configuration is needed before configuring Email feature. Refer to Chapter 3.

1. Open Cscape and then **Email Configuration** dialog, Click on Program and then select **Messaging → Email**.

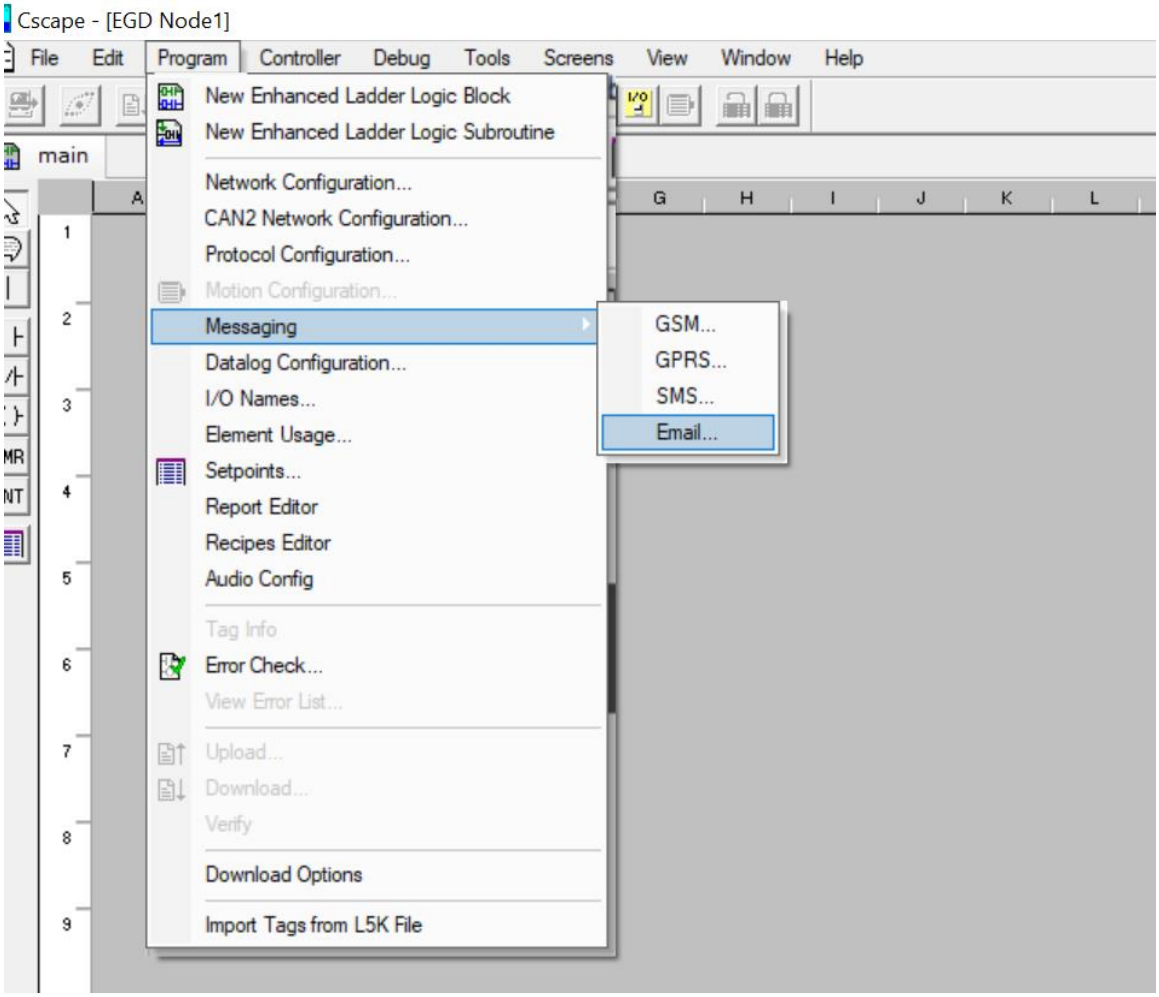


Figure 13.1 – Opening Email Configuring Dialog

2. This will bring up the following dialog. Select **Enable Email Configuration** to begin email configuration. See Table 15.1 for more configuration information.

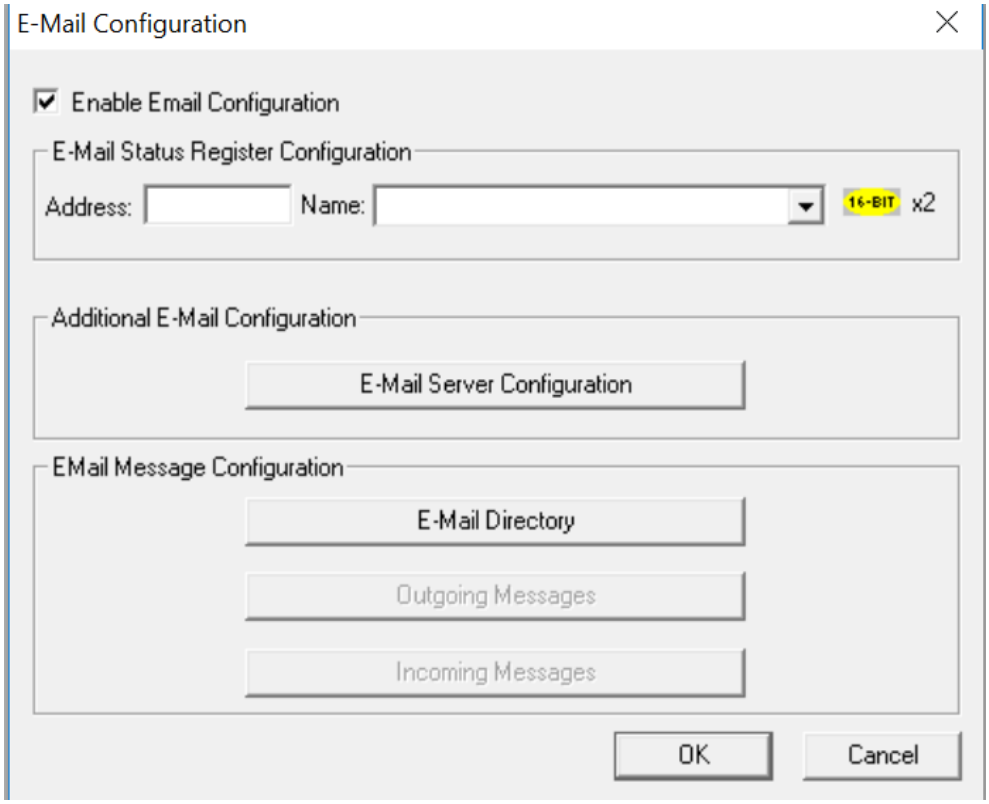


Figure 13.2 – Email Configuring Dialog

Table 13.1 – Email Status Register Configuration	
Address	Enter the starting register location to indicate the status of the email communication.
Name	Enter (or select) an I/O name.

13.3 Email Server Configuration

Table 13.2 – Email Server Settings	
E-Mail Address	The Email address of the controller can be configured here. If the user wants to provide email address through register, the **Get Settings from Register** checkbox can be clicked and register reference can be provided in **E-mail Address** edit-box. The register mentioned here should contain the email address followed by null termination or space.
SMTP Server	**SMTP Server IP:** The SMTP Server IP address of the email service provider needs to be provided here. In case SMTP server IP needs to be obtained from DNS server, select **Obtain SMTP Server IP Address from DNS Server** checkbox. This will enable **DNS Server IP** and **SMTP Server Name** edit-boxes. Configure these with the information provided by email service provider. **Port:** The default port for email configuration is set to 25 but can be changed by user if required. If '**Get Settings from Register**' is checked: **DNS Server IP** and **SMTP Server IP** needs to be provided in 32-bit IP format in the defined register location. **Port** number needs to be provided in 16-bit numeric format at the defined register location. **SMTP Server Name** (if required) needs to be provided at the register mentioned followed by null termination or space.
SMTP Authentication	The firmware for E-Mail also supports Login type of SMTP authentication. To use this feature, the **Authentication SMTP** checkbox should be checked. The username and password for authentication needs to be provided here. (TSL/SSL authentication servers are not supported) If '**Get Settings from Register**' is checked: **User Name** and **Password** needs to be provided at the register location mentioned followed by null termination or space.

Two options are available for configuring email in Cscape.

OPTION A: Click on **Email Server Configuration** to configure email address, server address, and authentication.

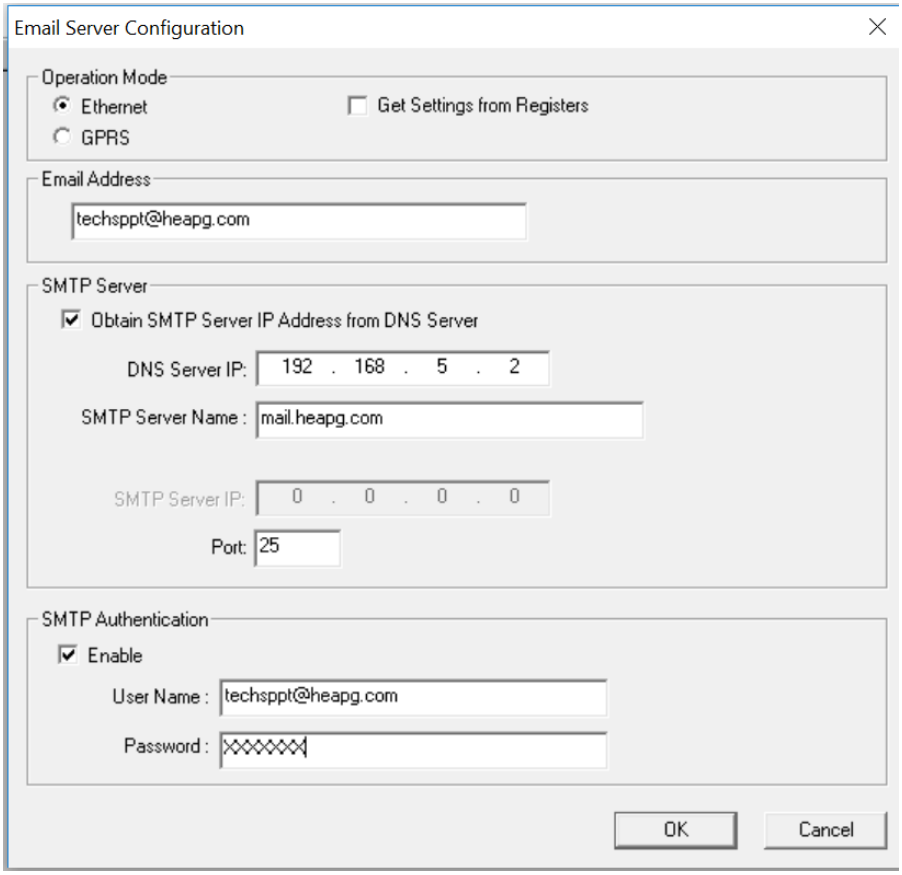


Figure 13.3 – Option A: Email Configuring Dialog

OPTION B: If the user wants to provide the email configuration through registers, '**Get Settings from Register**' checkbox should be checked. In this case the entire configuration data is taken from logic registers.

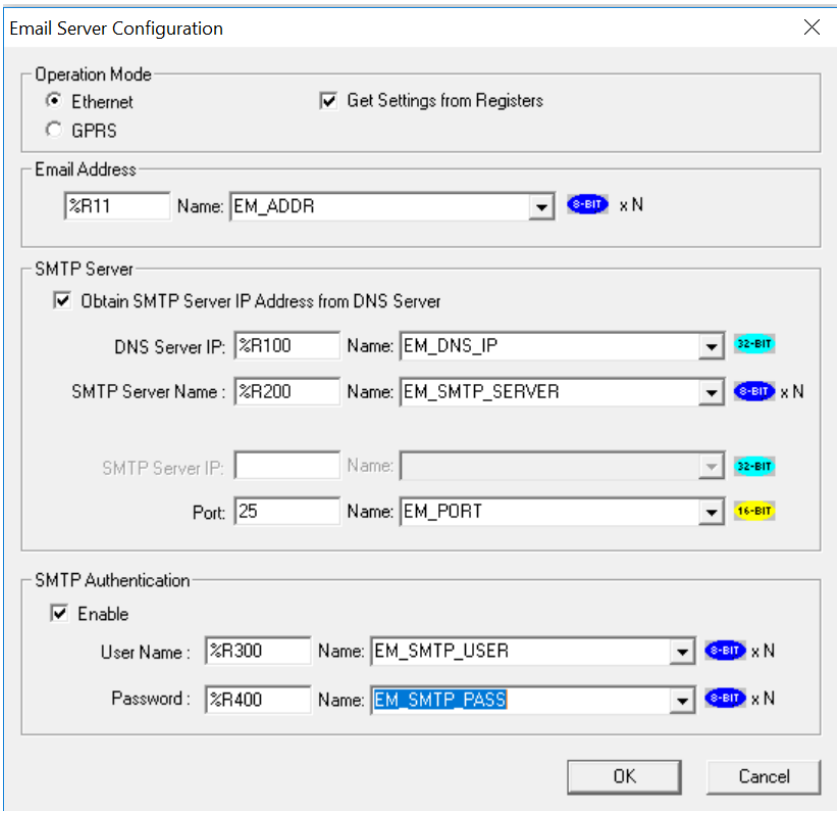


Figure 13.4 – Option B: Email Configuring Dialog

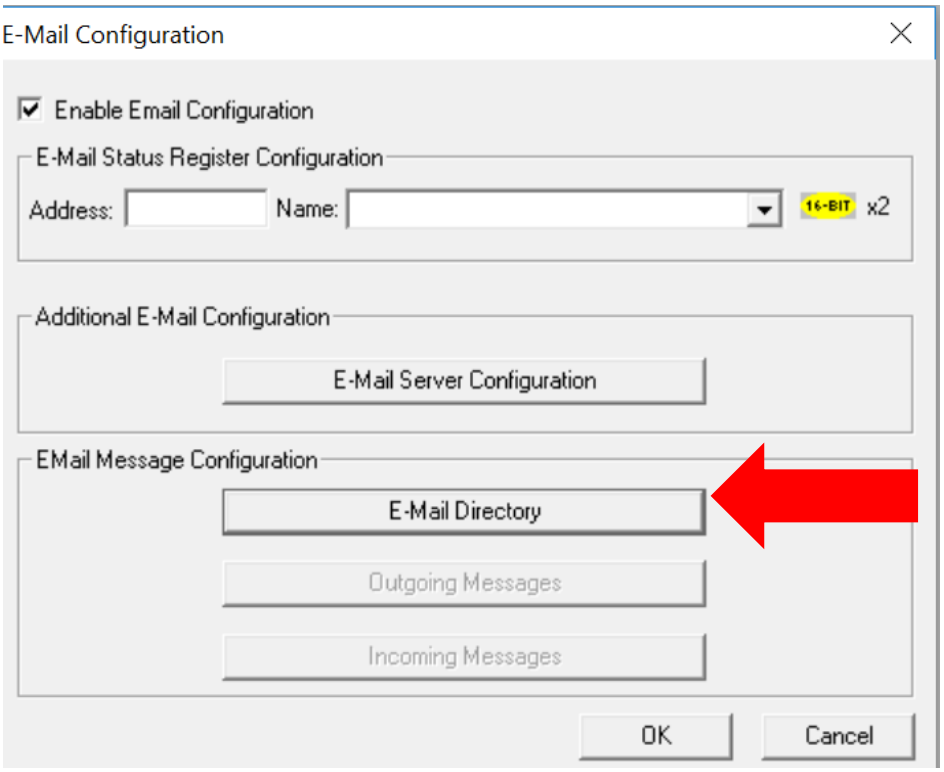
Following considerations must be made when using registers for configuration.

- DNS Server IP and SMTP Server IP need to be provided in 32-bit IP format in the defined register location.
- Port number needs to be provided in 16-bit numeric format at the defined register location.
- Email address needs to be provided at the register mentioned followed by null termination or space.
- SMTP Server Name (if required) needs to be provided at the register mentioned followed by null termination or space.
- User Name and Password (if required) need to be provided at the register location mentioned followed by null termination or space.

Once email server is configured, click **OK**, and return to the email

13.4 Email Directory Configuration

Before creating emails, a directory of email addresses needs to be created. Click on E-Mail Directory button to configure email addresses.



Email Target Directory

Email address configuration is very similar to SMS Address configuration with email addresses replacing phone numbers.

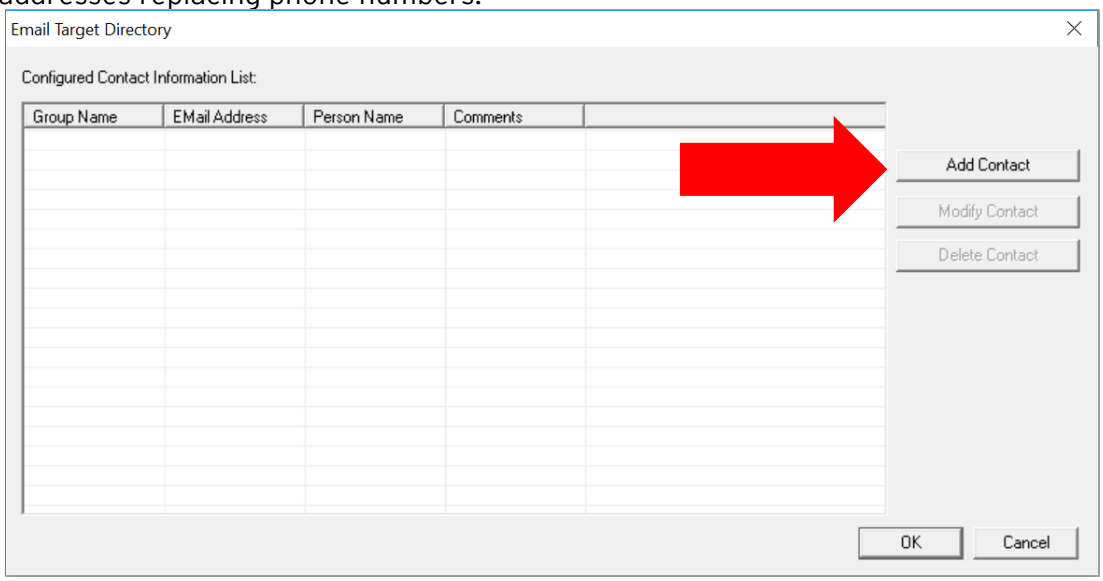


Figure 13.5 –Email Target Directory Dialog

Table 13.3 – Email Target Directory	
Add Contact	Add a new contact
Modify Contact	If a contact is already added in the directory. (You will see information about the contacts in the above screen.) To modify the contact, either double-click the row or highlight the row and click **Modify Contact**.
Delete Message	To remove a listing after highlighting the row.

Email Directory

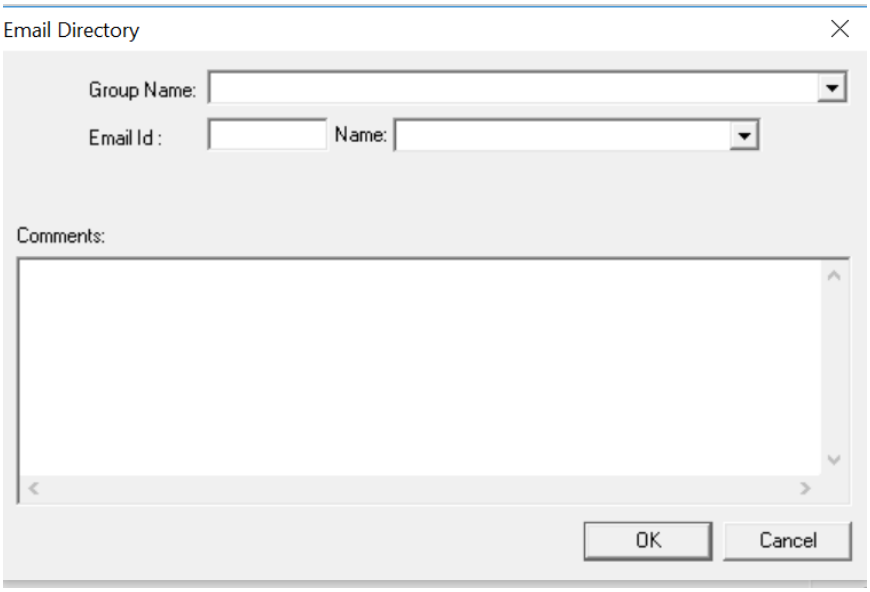
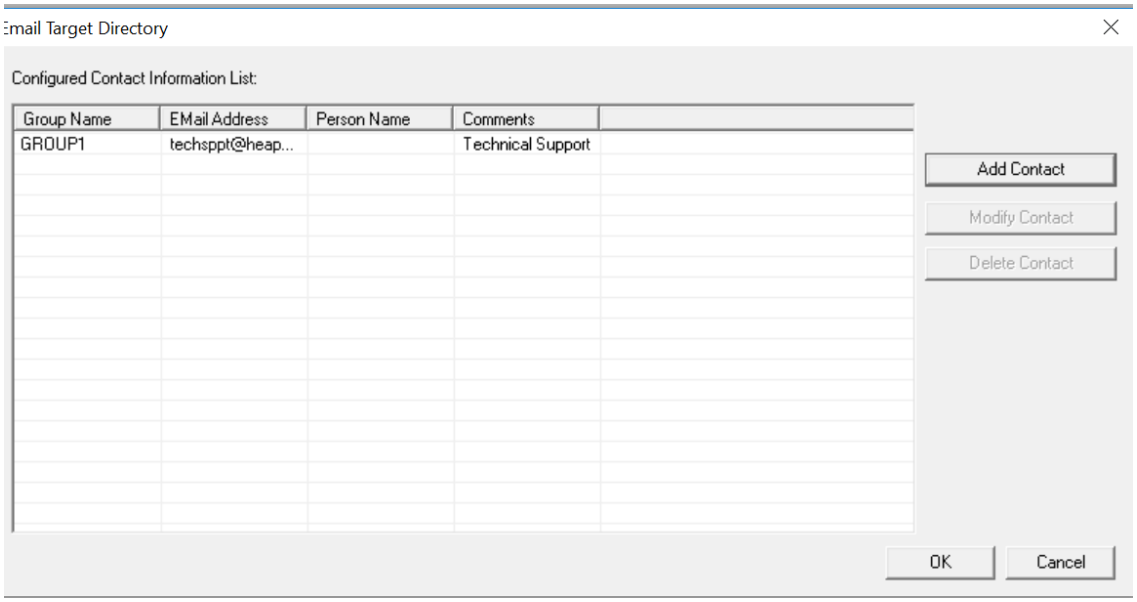


Figure 13.6 –Email Directory Dialog

Table 13.4 – New Contact Information	
Group Name	Enter or select a Group Name. 1. It can contain characters A-Z, a-z, 0-9, and the _ underscore character. 2. The first character must be A-Z, a-z, or the _ underscore character. 3. Do not use spaces or special characters. 4. Do not use two consecutive underscore characters.
Email ID	Either enter an Email ID or enter a register where the email ID is stored. The email address can be provided through register also wherever it is required to be changed the address in runtime. The register mentioned here should contain the email address followed by null termination or space.
Mail ID Name	If the Email ID is provided through register, the IP name also can be provided here.
Comments	Add description/comments for the email address here.

Click **OK** to see the contacts in the directory.

The following screen shows examples of contacts added to the directory.



With the directory being created, you are now ready to begin creating send email message formats.

13.5 Outgoing Messages

NOTE: Before creating send emails, a directory needs to be created. Up to 128 different emails can be configured.

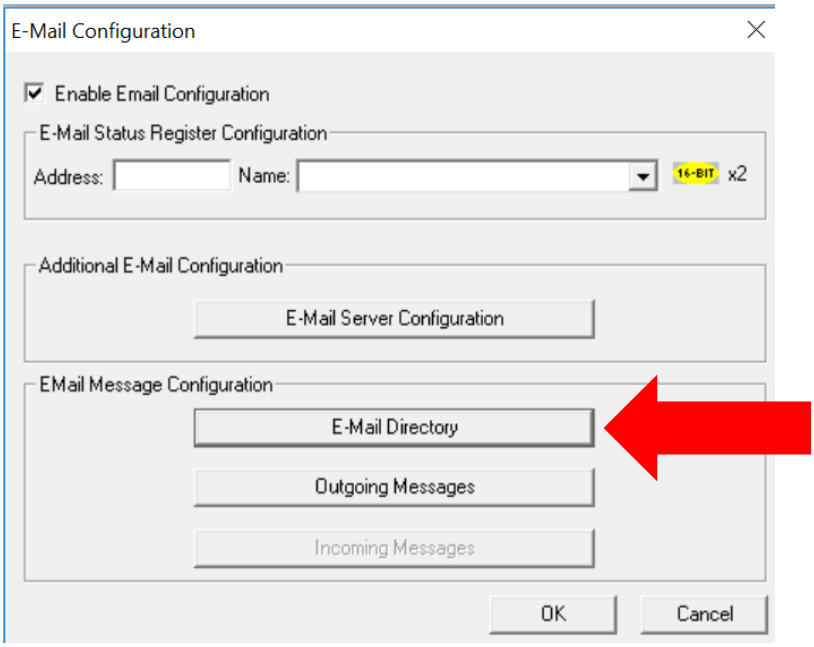


Table 13.5 – New Email Message	
Add New Message	Add a new email message format (i.e. email body)
Modify Message	Edit a message that is already in the list. Either double-click the row or highlight the row and click **Modify Message**.
Delete Message	Remove a message after highlighting the row.

When **Add/Modify Message** is clicked, the following screen appears.

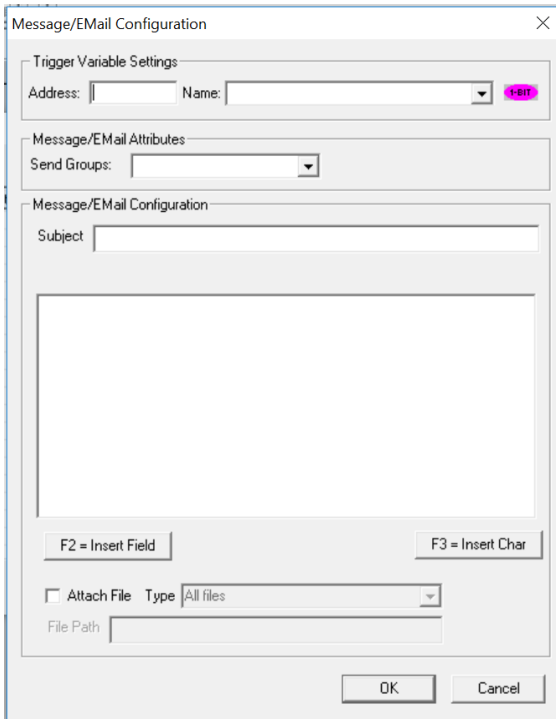


Figure 13.7 –Message/Email Configuration Dialog

Table 13.6 – Email Message Configuration	
Trigger Variable Settings	An event is needed to trigger an Email communication from the controller to the member(s) of a Group. Enter a bit reference in the **Address** field that (when set to HIGH) causes the Email message associated with the trigger to be sent to the specified Group member(s). An I/O name can be entered or selected in the **Name** field. On successful transmission of email, the trigger bit is made low by firmware. In case of error in transmission, the status register is updated, and firmware keeps on trying to send email till trigger bit is made low by the user.
Message/ Email Attributes	Enter or select the group that the Email message is sent to. Enter or select the **Send Group** that this email message is to be sent to.
Message / Email Configuration	Messages can contain **text** and **register data values,** which approved group members can read from the controller's data registers at runtime. • Enter the **Subject** of the email communication. • Enter the body of email. • To edit/add register data values, click **F2 = Insert Field** button and configure the **Insert Value Field** screen shown in Figure 13.8 below. • To have special characters added, click **F3 = Insert Char** button and configure the **Select a Character to Insert** screen shown in Figure 13.9 below.

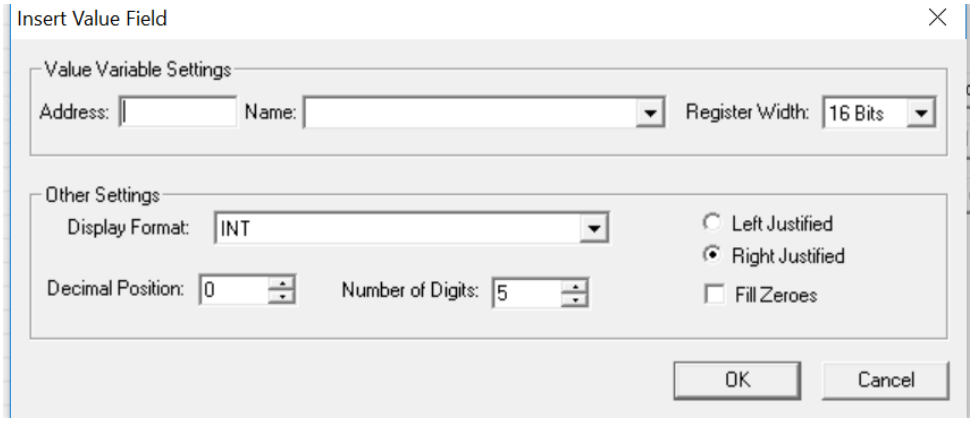


Figure 13.8 –Insert Value Field Dialog

Table 13.7 – Insert Value Field	
Value Variable Settings	Enter a register reference where the data embedded in the email to send will be stored. An I/O name can be entered or selected in the **Name** field. Select **Register Width** also. Up to 20 register value fields can be added to an email message.
Other Settings	Select the **Display Format** from the drop-down. In the **Decimal Position** field, enter the position of the decimal point. Enter the number of digits in the **Number of Digits**. **Example: xx.xx** shows that the decimal is in the 3rd position and the total number of digits (including the decimal) is 5. Click the **Fill Zeroes** box and the **Left Justified** or **Right Justified** box if desired.

- Click **F3 = Insert Char** button or press F3 to place a character to be place in email body.

Figure 13.9 –Select a Character to Insert Dialog

- Click **OK** to add registers/characters in the email body.
- Click **OK** to return to the Email Configuration screen.

13.6 Email Status

Email Status Registers are two 16-bit registers. The first is bitmapped as shown in Table 13.8, while the second status register reflects an SMTP error if one occurs.

Table 13.8 – Email Status – Troubleshooting

Bit #	Error	Reason	Remedy
1	Invalid Configuration Error	User-downloaded Cscape configuration is invalid.	Check and re-download email configuration.
2	Invalid IP or Port Address	Configured email server IP or port address is incorrect.	Verify configured email server IP port address.
3	Socket Error	Internal TCP Resource Error.	Reduce number of TCP connection in use by device and retry.
4	Bind Error	Internal TCP Resource Error.	Reduce number of TCP connection in use by device and retry.
5	Connection Timeout Error	Device is not able to connect specified email server within 30 seconds of timeout value.	Check configured email server name (or IP address) and port address.
6	Transmit Buffer Error	Internal memory resource error.	Reset the device and retry.
7	Transmit Error	Internal resource or connection break	Check configured email server name (or IP address) and port address.
8	UDP Resource Error	Internal resource error.	Reduce number of UDP connection in use by device and retry.
9	DNS Error Response	Erroneous response received from DNS server.	Check configured DNS server IP address and email server name.
10	DNS Response Timeout	No response received from DNS server.	Check configured DNS server IP address.
11	SMTP Command Timeout	No response received from email server for email command sent.	Check configured email server name (or IP address), port address, email address (to and from), username and password.
12	SMTP Error Response	Error response received from email server for email command sent.	Check configured email server name (or IP address), port address, email address (to and from), username password.
13 to 16	SMTP Error State Code	This field provides SMTP state (or command) code where error occurred.	1 = DNS Query State 2 = DNS Response State 3 = Establish Connection State 4 = SMTP HELO Command State 5 = SMTP Authentication State 6 = SMTP Username State 7 = SMTP Password State 8 = SMTP from Mail Address State 9 = SMTP Receiver Mail Address State 10 = SMTP Data Command State 11 = SMTP Send Data State 12 = SMTP End Data State 13 = SMTP Quit State 14 = Waiting for Response
Second E-mail Status Register	SMTP Server Error Response Code	This field provides SMTP Error response code received from SMTP server.	Details of different error codes can be found in the RFC 821 and RFC 2554 documents. Ex: '500' Syntax error, command unrecognized. An Internet search, i.e. "SMTP 500", is a good resource for troubleshooting error codes if the e-mail service provider does not have specific help with the error code.

NOTE: In Email feature, transmit trigger bit should get reset within 30s of time or error status should get updated indicating error information. If there is no status, it means either device is not in RUN mode or email feature is not configured or invalid email configuration is downloaded.

CHAPTER 14: ASCII over TCP/IP

14.1 Overview

This protocol is designed to send and receive ASCII data over Ethernet port of the controllers. Controller acts as a server while using this protocol.

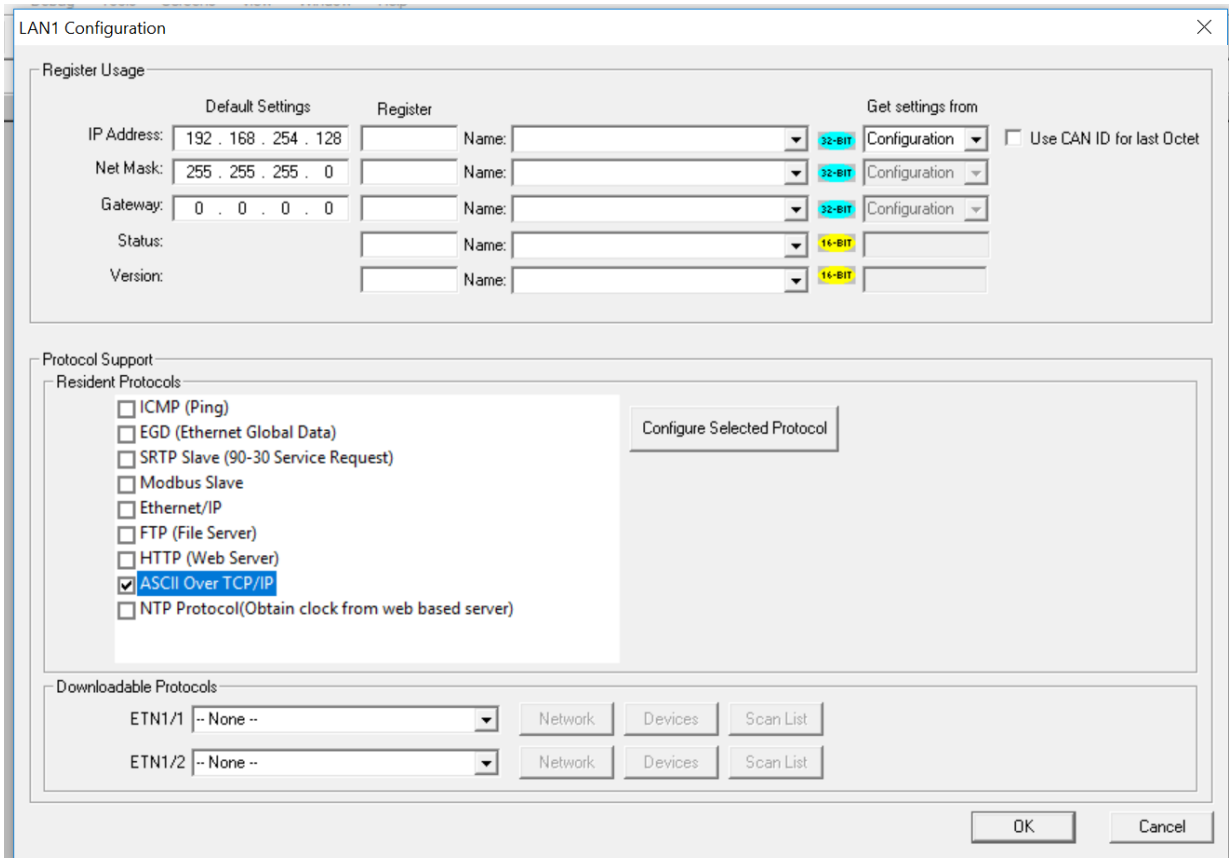
Minimum version requirements for ASCII over TCP/IP feature:

- Firmware 12.6 or later
- Cscape 9 or later

14.2 Configuration

a. Open CSCAPE and click on **Program → Hardware Configuration** (select a series and device type)→ **LAN1 / Config** button. (You can also access the Hardware Configuration dialog by clicking on the  menu option.)

In LAN1 configuration window set up IP address, Net mask, Gateway, and select **ASCII over TCP/IP** checkbox and the click on Configure Selected Protocol as shown below:



b. In **ASCII over TCP/IP Configuration** window configure Port Number, register for start address of Transmit data, register for start address for Receive data, Transmission Trigger Register and Status Address.

Port Number can be provided directly or by using register address. For port number entry using register user has to select **Get from Register** option.

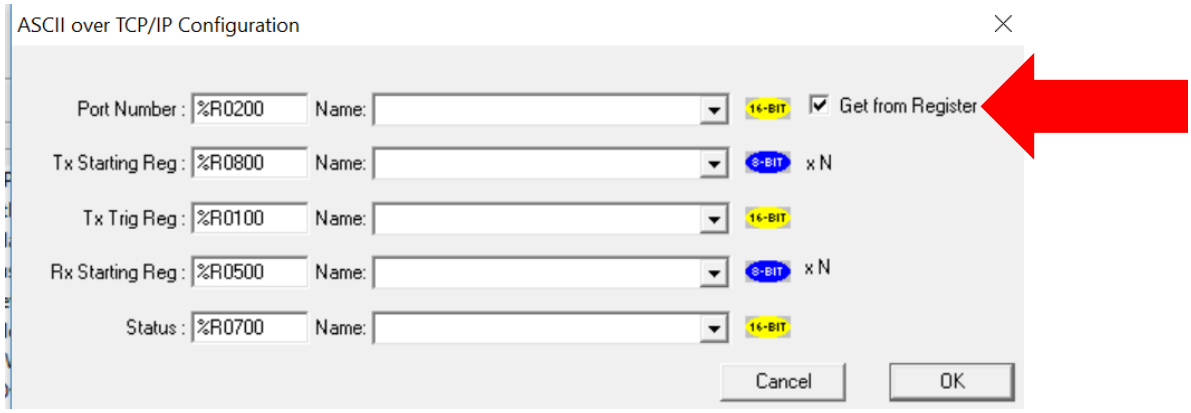


Figure 14.1 – ASCII over TCP/IP Configuration

14.3 Sending ASCII characters to the client

In above configuration dialog %R800 and %R100 registers are configured for sending data.

%R800 is start of transmit data and should contain data in following format.

Table 14.1 - %R800 Register and Data

Sl.No	Register Address	Data Content
1	%R800	First Register should contain 'Transmit Count' i.e. number of bytes (characters) to be transmitted.
2	%R801	Reserved.
3	%R802	Data to be transmitted. Each Register can hold 2 bytes of data.
4	%R803	Data to be transmitted.
	..	Data to be transmitted.
	..	Data to be transmitted.
n	%R(800+n-1)	Data to be transmitted.

NOTE: Maximum 2K ASCII characters can be sent in a single request.

%R100 is Transmit Trigger register used to start the transmission process. Setting R100 data to '1' will initiate transmission process and on successful transmission firmware will reset the R100 data to '0'.

14.4 Receiving ASCII characters from the client

In above configuration dialog %R500 register is used for receiving data. The receive data register format will be as below.

Table 14.2 - %R500 Register and Data		
Sl.No	Register Address	Data Content
1	%R500	First Register contains number of byes (characters) received.
2	%R501	User can provide valid termination character here (if any). If there is no termination character set this register value to zero.
3	%R502	Received data. Each Register can hold 2 bytes of data.
4	%R503	Received data.
	..	Received data.
	..	Received data.
n	%R(500+n-1)	Received data.

NOTE: Maximum 2K ASCII characters can be received in a single request.

14.5 Status Information

%R700 register is used for to display status information. Details of Status Register as below.

Table 14.3 - %R700 Register and Data	
Bit No.	Description
1	**Receive Data Overflow:** Set when number received characters are more than 255.
2	**Transmit Data Overflow:** Set when transmit character count is more than 255.
3	**Socket Creation Error:** Set if there is any error in Socket creation.
4	**Socket Allocate Error:** Set if there is any internal memory allocation error for transmitting data.
5	**Socket Send Error:** If error in transmitting TCP packet.
6	**Reserved**
7	**Reserved**
8	**Reserved**
9	**Reserved**
10	**Reserved**
11	**Reserved**
12	**Reserved**
13	**Reserved**
14	**Reserved**
15	**Reserved**
16	**Receive Data indication Bit:** Set when new packet is received. User can reset this bit to '0'.

CHAPTER 15: WEBMI

NOTE: Please refer to the WebMI Manual for full configuration details: MAN1036.

15.1 WebMI Overview

Web-Machine Interface, or WebMI, is a HTML5 based HTTP server, where access to data and visualizations is enabled from anywhere via web clients. WebMI empowers users to control everything onscreen from a computer, tablet, or other mobile device. It is not necessary for the user to create HTTP web pages. The pages are created by Cscape via the Graphic Editor and transferred to the OCS controller's removable media card. This turns the OCS controller into a web server that can be used with most internet browsers. WebMI support requires Cscape 9.70 and Firmware 14.14 at a minimum. It is recommended to use Cscape 9.80 SP2 and Firmware 15.0 and onwards to take advantage of improvements and additional features.

The web pages can be published from either the primary controller application or from an alternate 'Web Designer' application, which is selected in the Cscape hardware configuration. The typical method for transferring the published web pages to the OCS controller is via FTP over an Ethernet connection. Using FTP, the pages are transferred from Cscape to the Removable Media card. Once the published web page files have been successfully transferred to the Removable Media card in the controller, the interface can be remotely viewed from any PC, or wireless device connected to the same local network.

15.2 WebMI Security

WebMI supports non-secure HTTP on Port and secure HTTPS on Port 443. For HTTPS connections, TLS ver. 1.2 protocol is supported.

15.3 Setting up a Horner OCS for Use with WebMI

Removable Media / USB / microSD File System

X, XL, and RCC controllers:
The microSD card memory slot uses the PC-compatible FAT32 file system supporting up to 32GB cards.

ZX controllers:
Support USB 2.0 up to 32GB.

XL+ controllers:
The microSD card memory slot supports SDHC, SDXC IN FAT32 format up to 128GB max.

Ethernet Configuration - IP Parameters from Cscape Hardware Config / LAN Config

Open the **Cscape Hardware Configuration** window and verify that the Series and Device Type match the connected controller. Enter the desired network parameters in the Cscape Hardware Configuration / LAN Config.

For simplicity, the examples in this manual will show a local connection directly from PC to OCS. IP address 192.168.0.110 is used for the OCS IP. Hence, the PC IP address must be set on the same subnet, using an IP such as 192.168.0.x, where x is in the range of 1-254. Both subnet addresses should be set to 255.255.255.0.

Example OCS:
IP Address – 192.168.0.110
Subnet Mask – 255.255.255.0
Gateway – 0.0.0.0

Example PC:
IP Address – 192.168.0.99
Subnet Mask – 255.255.255.0
Gateway – 0.0.0.0

For OCS models with more than one Ethernet connection, these settings must be done on LAN1. Other OCS models have only one LAN port so this is not a concern.

NOTE: The OCS controller cannot open the door to make itself available to the World Wide Web; the path has to be set in place for it. For connections that are to be accessed remotely, use an IP address as assigned by your IT staff or Internet Service Provider. There are also hardware solutions available from companies such as MC Connect, Secomea, Ewon, and others that can allow an OCS and WebMI functionality to be accessible from the internet. For more details, contact your local distributor.

CHAPTER 16: SAFETY INFORMATION AND TECH SUPPORT

16.1 Safety

When found on the product, the following symbols specify:

Warning: Consult user documentation.

Warning: Electrical Shock Hazard.

WARNING: Remove power from the OCS controller, CAN port, and any peripheral equipment connected to this local system before adding or replacing this or any module.

WARNING: To avoid the risk of electric shock or burns, always connect the safety (or earth) ground before making any other connections.

WARNING: To reduce the risk of fire, electrical shock, or physical injury it is strongly recommended to fuse the voltage measurement inputs. Be sure to locate fuses as close to the source as possible.

WARNING: Replace fuse with the same type and rating to provide protection against risk of fire and shock hazards.

WARNING: In the event of repeated failure, do not replace the fuse again as a repeated failure indicates a defective condition that will not clear by replacing the fuse.

WARNING: Only qualified electrical personnel familiar with the construction and operation of this equipment and the hazards involved should install, adjust, operate, or service this equipment. Read and understand this manual and other applicable manuals in their entirety before proceeding. Failure to observe this precaution could result in severe bodily injury or loss of life.

For detailed installation and a checklist that covers panel box layout requirements and minimum clearances, refer to the hardware manual of the controller you are using.

- All applicable codes and standards need to be followed in the installation of this product.
- For I/O wiring (discrete), use the following wire type or equivalent: Belden 8917, 16 AWG or larger.

Adhere to the following safety precautions whenever any type of connection is made to the module.

- Connect the green safety (earth) ground first before making any other connections.
- When connecting to electric circuits or pulse-initiating equipment, open their related breakers. Do **NOT** make connections to live power lines.
- Make connections to the module first; then connect to the circuit to be monitored.

- Route power wires in a safe manner in accordance with good practice and local codes.
- Wear proper personal protective equipment including safety glasses and insulated gloves when making connections to power circuits.
- Ensure hands, shoes, and floors are dry before making any connection to a power line.
- Make sure the unit is turned OFF before making connection to terminals. Make sure all circuits are de-energized before making connections.
- Before each use, inspect all cables for breaks or cracks in the insulation. Replace immediately if defective.

16.2 TECHNICAL SUPPORT

For manual updates and assistance, contact Technical Support at the following locations:

North America:
Tel: (317) 916-4274
Fax: (317) 639-4279
Website: https://hornerautomation.com
Email: techsppt@heapg.com

Europe:
Tel: (+) 353-21-4321-266
Fax: (+353)-21-4321826
Website: www.horner-apg.com
Email: technical.support@horner-apg.com

INDEX

INDEX OF FIGURES AND TABLES