

CTI 2500P-ECC1
Communications Coprocessor
User Manual

Version 1.26

Copyright © 2021-2025 Control Technology Inc.
All rights reserved.

This manual is published by Control Technology Inc. (CTI) 5734 Middlebrook Pike, Knoxville, TN 37921. This manual contains references to brand and product names which are tradenames, trademarks, and/or registered trademarks of Control Technology Inc. Siemens®, SIMATIC®, and Series 505®, and 505® are registered trademarks of Siemens AG. Other references to brand and product names are tradenames, trademarks, and/or registered trademarks of their respective holders.

DOCUMENT DISCLAIMER STATEMENT

Every effort has been made to ensure the accuracy of this document; however, errors do occasionally occur. CTI provides this document on an “as is” basis and assumes no responsibility for direct or consequential damages resulting from the use of this document. This document is provided without express or implied warranty of any kind, including but not limited to the warranties of merchantability or fitness for a particular purpose. This document and the products it references are subject to change without notice. If you have a comment or discover an error, please call us toll-free at 1-800-537-8398 or email us at sales@controltechnology.com.

REVISION HISTORY		
V1.0	11/6/2012	Initial Release
V1.1	1/28/2013	Expanded description for use of STW267 (advanced tab) Expanded description for setting Broadcast Storm parameters (advanced tab) Expanded information regarding maximum amount of data that can be mapped to each protocol Documented CAMP server support for TC14 (Appendix C)
V1.2	4/17/2013	Added more information about Broadcast Storm Protection Documented CAMP server support for task code 11 and 12 Revised module startup text Added description of ECC1 Status Word bits Revised description of STW2048 bits to improve clarity Expanded description of the embedded Web Server pages.
V1.3	6/10/2013	Documented CAMP Server support for additional task codes: 05, 07, 09,0C, 0E, 19, and 59. Revised information regarding SD card selection. Added comments noting that CAMP Client UDP Multicast is restricted to the local area network. Documented configuration program error messages.
V1.4	9/24/2013	Documented User Switches and jumpers.
V1.5	7/29/2014	Eliminated reference to Communications Status and Fatal Error bits in the ECC1 status word and STW2048, which is an alias of the ECC1 status word. Added note that the Status word is not accessible when the Host Controller is disconnected or is in Fatal Error Mode. Revised text regarding SD card selection to improve clarity. Added description of “Display All Statistics” web server option.
V1.6	4/7/2015	Added recommendations for assigning error status and last error status to tagname database items. Added a list of all task code error codes.
V1.7	9/8/2016	Documented Port Isolation feature (SW4)

REVISION HISTORY		
V1.8	10/4/2016	Enabled hyperlinks in table of contents. Documented support for task codes 60, 64, 76, and 79 in Appendix C. Revised description of serial port to clearly indicate that it is unused.
V1.9	10/6/2016	Revised text to note that the 2500P-ECC1 Configuration Program V1.5 or greater will allow more bits of STW267 to be selected and will automatically default to a Broadcast Storm protection rate of 2% rather than 10%. Revised the text regarding storm protection to reflect field experience.
V1.10	10/13/2016	Deleted text indicating that the Configuration Program supports Windows XP. Added text to document that the 2500P-ECC1 Configuration Program also runs under Windows 10.
V1.11	5/31/2017	Documented error codes 4E and 4F. Documented error codes 5E and 5F. Added comments in Data Element Support table that DCC is read-only.
V1.12	11/2/2020	Added documentation for the IP Alias feature. Added documentation for web server reset of module.
V1.13	1/25/2021	Corrected Ethernet firmware update method. Eliminated text regarding display of IP address.
V1.14	3/16/2021	Deleted remaining references to Windows XP.
V1.15	4/12/2021	Corrected comments for Error 320. Created a hyperlink to error 320. Added comments to TC19 description indicating only the force bit is updated. Added hyperlink to Ethernet Port Isolation feature.
V1.16	5/4/2021	Added a note regarding the source of the Product Clock displayed in web page statistics and the Date/Time shown in product web page headings.
V1.17	8/4/2021	Increased the maximum number of Modbus Server data blocks allowed. Added a section in Appendix regarding handling of Host Controller error conditions. Clarified mapping of multiple tagname database items to a single CAMP client request.
V1.18	8/23/2021	Replaced "AFM" in Use Case #3 of Appendix F with "ECC1 Module".
V1.19	12/4/2023	Revised descriptions for NITP errors 70, 71, and 72.
V1.20	1/2/2024	Added shipping configuration of switches and jumpers at Section 1.4.3
V1.21	4/10/2024	Added information about new Remote Reset feature to Section 5.3.2.
V1.22	5/10/2024	Added ⚠ to all WARNING notices. Corrected some formatting problems and incorrect cross references.
V1.23	7/30/2024	Corrected a problem with firmware version required for Remote Reset feature. It previously showed firmware > V2.21 was required. Now shows V2.26 and above.
V1.24	2/17/2025	Changes to Section 4.4 (DOCS-235)
V1.25	4/10/2025	Added note on function of Gateway IP address in Section 5.3.2 (CTI-167)
V1.26	5/29/2025	Updated Modbus Client Error 11 description in Appendix A (CTI-170)

PREFACE

This ***User Manual*** provides reference information for the CTI 2500P-ECC1 Ethernet Communications Coprocessor module. The information in this manual is directed to individuals who will be installing and operating the product, configuring the product for a specific application, and those who will be designing systems that use it.

USAGE CONVENTIONS

NOTE

Notes alert the user to special features or procedures.

CAUTION

Cautions alert the user to procedures that could damage equipment.

 WARNING

Warnings alert the user to procedures that could damage equipment and endanger the user.

TABLE OF CONTENTS

PREFACE	1
USAGE CONVENTIONS	2
TABLE OF CONTENTS	3
CHAPTER 1 INTRODUCTION	6
1.1 Definition of Terms.....	7
1.2 Getting Started.....	7
1.3 Front Panel Indicators and Connectors	8
1.3.1 <i>Status Indicator LEDs</i>	8
1.3.2 <i>Multi-Segment Display</i>	8
1.3.3 <i>Reset Button</i>	8
1.3.4 <i>Ethernet Indicators</i>	9
1.3.5 <i>Ethernet Ports</i>	9
1.3.6 <i>Ethernet Port LEDs</i>	9
1.3.7 <i>Serial Port</i>	9
1.4 User Switches and Jumpers.....	10
1.4.1 <i>User Switches</i>	10
1.4.2 <i>User Jumpers</i>	10
1.4.3 <i>Shipping Configuration of Switches and Jumpers</i>	11
CHAPTER 2 PROTOCOLS	13
2.1 CAMP Server	13
2.2 CAMP Client.....	14
2.3 Modbus Server.....	14
2.4 Open Modbus Client.....	15
2.5 Network Data Exchange Publish and Subscribe.....	16
CHAPTER 3 INSTALLATION	17
3.1 Installation Planning.....	17
3.1.1 <i>SD Card Requirements</i>	17
3.1.2 <i>Ethernet Cabling</i>	18
3.1.3 <i>Communications with a Host Controller</i>	18
3.1.4 <i>CTI 2500 Series Controller Firmware Update</i>	19
3.1.5 <i>2500P-ECC1 Firmware Update</i>	19
3.1.6 <i>Configuration Planning</i>	19
3.1.7 <i>Power Requirements</i>	19
3.2 Installing the 2500P-ECC1 module	19
3.2.1 <i>Unpacking the Module</i>	19
3.2.2 <i>Physical Installation</i>	20
3.2.3 <i>Connecting to the Host Controller</i>	20
3.3 QuickStart Configuration	21
CHAPTER 4 OPERATION	25
4.1 Module Startup.....	25
4.2 Normal Operation.....	27

4.3	Cache Update Performance Tuning	27
4.3.1	<i>Configuration Parameters That Affect Performance</i>	27
4.3.2	<i>Determining Operating Status</i>	29
4.3.3	<i>Tuning the 2500P-ECC1 Operation</i>	30
4.4	Troubleshooting Operational Problems	31
CHAPTER 5	CONFIGURATION	35
5.1	Configuration Overview	35
5.2	Installing the 2500P-ECC1 Configuration Program.....	35
5.3	Using the 2500P-ECC1 Configuration Program.....	35
5.3.1	<i>Configuration Program Main Window</i>	36
5.3.2	<i>Entering ECC1 Settings</i>	38
5.3.3	<i>Entering Tagname Database Data Items</i>	42
5.3.4	<i>Selecting and Configuring Protocols (Overview)</i>	46
5.3.5	<i>Configuring the CAMP Server</i>	50
5.3.6	<i>Configuring the CAMP Client</i>	51
5.3.7	<i>Configuring the Open Modbus Server</i>	60
5.3.8	<i>Configuring the Open Modbus Client</i>	63
5.3.9	<i>Configuring the Network Data Exchange Publisher</i>	71
5.3.10	<i>Configuring the Network Data Exchange Subscriber</i>	74
CHAPTER 6	UPDATING FIRMWARE.....	77
6.1	Overview	77
6.2	Ethernet Firmware Update Method.....	77
6.3	SD Card Firmware Update Method	79
6.4	Firmware Update Status Codes.....	80
APPENDIX A:	ERROR CODES	81
	Initial Startup Error Codes	81
	Operational Error Codes	81
	Host Controller Status Error Handling	86
	Protocol Error Codes.....	87
	<i>CAMP Server Error Codes</i>	87
	<i>CAMP Client Error Codes</i>	89
	<i>Task Code Error Codes</i>	90
	<i>Modbus Server Error Codes</i>	93
	<i>Modbus Client Error Codes</i>	94
	<i>Network Data Exchange Subscriber Error Codes</i>	95
	Firmware Update Error Codes.....	95
	Configuration Error Codes	97
	Configuration Program Error Messages.....	102
APPENDIX B:	IP ADDRESS INFORMATION	103
	IP Address Nomenclature	103
	Using the Subnet Mask	104
	Selecting an IP Address	105
	Selecting a Multicast Address.....	106
APPENDIX C:	CAMP SERVER SUPPORT	107
	Overview.....	107
	Task Code Support	107
	CAMP Message Support.....	108
	Data Element Support.....	109
	Status Word 2048.....	113

APPENDIX D: DATA CONSISTENCY	115
Overview	115
How Data Consistency Works	115
Guidelines for Using Data Consistency	116
APPENDIX E: DATA CACHE OPERATION.....	117
Overview	117
Cache Access.....	117
Cache Membership	117
Cache Update.....	118
APPENDIX F: ALIAS IP FEATURE.....	119
Overview	119
Use Cases	119
<i>Communicating on Two Ethernet Networks.....</i>	<i>119</i>
<i>Communicating on an Ethernet Network with Multiple IP Subnets</i>	<i>121</i>
APPENDIX G: PRODUCT SPECIFICATIONS	123
Hardware Specifications.....	123
LIMITED PRODUCT WARRANTY	125
REPAIR POLICY	127

CHAPTER 1 INTRODUCTION

The 2500P-ECC1 Ethernet Communications Coprocessor module extends the communications capabilities of CTI 2500 Series® controllers, offering increased connectivity and additional protocols. By offloading the processing of communications protocols from the 2500 Series® controller, the 2500P-ECC1 module minimizes the impact of network communications on controller performance.

The 2500P-ECC1 module communicates with the 2500 Series® controller via a high speed Ethernet link. Data is transferred between the controller and module using a high density Ethernet messaging protocol, which is capable of transferring a large amount of data while minimizing the impact on the 2500 Series® controller. The controller processes data transfer requests from the 2500P-ECC1 in a configurable time slice, allowing the user to limit the amount of scan time used to service the requests. Up to four 2500P-ECC1 modules can communicate concurrently with a 2500 Series® controller.

Client requests to read data from the 2500 Series® controller are serviced from a data cache maintained on the 2500P-ECC1 module. The frequency of cache updates is user configurable. This data cache design allows protocols such as CAMP Server to provide rapid response to a high volume of requests from SCADA workstations, HMI equipment, and operator panels while providing a means to manage data quality.

Client protocols, such as Open Modbus, can be used to access data in other controllers and devices. PLC logic is not required to initiate client requests, since the 2500P-ECC1 can automatically initiate requests based on a change in data value or on a designated time interval. However, client requests can be triggered using PLC logic, if desired.

The 2500P-ECC1 module supports the following communications protocols:

- CAMP Server
- CAMP Client
- Modbus Server
- Modbus Client
- Network Data Exchange

See CHAPTER 2 for information on these protocols.

Using a CTI-supplied configuration program, you can select the protocols to be used and designate how they will operate. Configuration data is saved on a removable SD memory card, located on the 2500P-ECC1 circuit board. If a module ever fails, a replacement can quickly be placed in service by transferring the SD card from the failed unit to the new one.



1.1 Definition of Terms

CRI	Cache Refresh Interval. The CRI specifies how often a data cache item is updated.
Host Controller	The CTI 2500 Series® controller with which the 2500P-ECC1 module is exchanging data. The ECC1 module performs communications tasks on behalf of the host controller.
IP	Internet Protocol: A suite of protocols used to relay data packets across networks.
LAN	An acronym for Local Area Network. In this manual it refers to an Ethernet local area network. All devices on a local area network can directly communicate with each other and are members of the same broadcast domain.
Mapping	The association of protocol data with Host Controller memory addresses, enabling data to be transferred between the protocol and Host Controller memory.
MSD	Multi-Segment Display, located on the front panel of the ECC1 module.
Network Device	General term for equipment with which the ECC1 can communicate, such as PLCs, workstations, operator interfaces, weighing stations.
PLC	An acronym for a Programmable Logic Controller
Protocol	A system of digital message formats and rules for exchanging messages between systems. Also, the task that implements the protocol.
VLAN	An acronym for Virtual Local Area Network. VLANs provide a means of subdividing a physical Ethernet LAN into logical isolated LANs, each with its own broadcast domain. VLANs are implemented in Ethernet switches.
SD Card	A <i>Secure Data</i> Card is a portable non-volatile memory card with a data format and form factor that complies with standards maintained by the SD card association. The 2500P-ECC1 uses an SD card to store configuration files.

1.2 Getting Started

If you are not familiar with the 2500P-ECC1 Communications Coprocessor module, you should read the remainder of this chapter and CHAPTER 2, which describes the communications protocols supported by the ECC1 module.

To begin using your 2500P-ECC1 Ethernet Communications Coprocessor module, you will need to install the module, make the necessary Ethernet connections to the Host Controller and the network, and configure the module to meet your application requirements. You may also need to update the firmware of the 2500 Series® controller that you will be using as the Host Controller and/or your 2500P-ECC1 module.

- **Installation:** The 2500P-ECC1 installs in a CTI 2500 Series® base or a Siemens Series 505® base. See CHAPTER 3 for installation planning tips and instructions for installing the module.
- **Ethernet Connections:** To function properly, the module must have an Ethernet path to the Host controller and to the network devices with which it will communicate. See section 3.2.3 for connection alternatives.
- **Configuration:** Before you can use the module, you must configure it using the 2500P-ECC1 Configuration Program, which is available for download from the CTI web site. CHAPTER 5 provides comprehensive instructions regarding the module configuration. See Section 3.3 for a simple QuickStart configuration, which sets the IP parameters for the module and enables the CAMP server protocol.
- **Host Controller and ECC1 Module Firmware Update:** See Sections 3.1.4 and 3.1.5 for help on determining whether an update is required and obtaining an update, if required. CHAPTER 6 of this manual contains

instructions on updating the 2500P-ECC1 firmware. See the *CTI 2500 Controller Installation and Operation Guide* for instructions regarding the controller firmware update.

1.3 Front Panel Indicators and Connectors

1.3.1 Status Indicator LEDs

At the top of the module front panel are three status LEDs. These are turned on briefly when the module starts up to test the LEDs. The function of the each LED is described in the following table.

LED	State	Indication
GOOD	Off	Loading Application Firmware
	Flashing	Processing Configuration File
	On	Module Startup Successful
HOST	Off	Not Attempting to Communicate with Host Controller
	Flashing	Attempting to Communicate with Host Controller
	On	Successfully communicating with the Host Controller
ACTIVE	Off	Protocols are not Activated
	Flashing	At least one protocol has reported an error.
	On	All Configured Protocols are Activated and Operational

1.3.2 Multi-Segment Display

The Multi-Segment Display (MSD) is located below the status LEDs. The MSD is used to display status and error codes. During normal operation the MSD will display the TCP/IP address of the product, one octet at a time. When an error is encountered, the MSD will also display an Error Code. See *APPENDIX A: ERROR CODES* for a list of error codes and descriptions.

1.3.3 Reset Button

The reset button allows you to initiate a “soft reset” for the 2500P-ECC1 module. A “soft reset” restarts the module after closing all Ethernet TCP/IP connections. The button, which is recessed to prevent inadvertent reset, can be depressed using a pointed object such as a ball point pen.



1.3.4 Ethernet Indicators

The Ethernet LEDs indicate the state of the TCP/IP interface and whether the module is transmitting and receiving data via the Ethernet as shown in the following table.

LED	State	Indication
NS (Network Status)	Off	TCP/IP is not operational.
	On-Red	TCP/IP is operational. A device with the same IP address as this 2500P-ECC1 module has been detected.
	On-Green	TCP/IP is operational. No duplicate IP address has been detected.
XMT (Transmit)	Flashing	The 2500P-ECC1 is transmitting data via an Ethernet port
RCV (Receive)	Flashing	The 2500P-ECC1 is receiving data via an Ethernet port.

1.3.5 Ethernet Ports

The 2500P-ECC1 provides two Ethernet ports capable of operating at 10 or 100Mb, half or full duplex. The speed and duplex mode are automatically negotiated with the device connected to the port. Each port supports auto-crossover capability, allowing the port to be connected to an external Ethernet switch or directly to a device, such as a laptop or 2500 Series® controller. Both ports are functionally equivalent.

The Ethernet ports are connected to an Ethernet switch incorporated into the 2500P-ECC1 module. This switch is also connected to the Ethernet controller on the ECC1 microprocessor. This arrangement allows devices connected to either port to communicate with the microprocessor and allows communications between ports. The switch also provides hardware protection against excessive broadcast/multicast traffic.

1.3.6 Ethernet Port LEDs

Each Ethernet port connector contains two embedded LEDs. The **LINK** LED indicates whether the Ethernet port is successfully connected to another Ethernet device, such as a network switch. The **ACTIVITY (ACT)** LED provides visual indication that Ethernet packets are being received or transmitted via the port. See the following table below for more information.

LED	State	Indication
Link	Off	Ethernet link is not available.
	On	Ethernet link is available.
Act (Activity)	Off	There is no Ethernet frames are being transmitted on the network to which the port is connected..
	Flashing	Ethernet frames are being transmitted on the network to which the port is connected

1.3.7 Serial Port

This port is not used on this product.



1.4 User Switches and Jumpers

User switches and jumpers provide a way to enable certain product functions. These switches and jumpers are located on the product printed circuit board (PCB). The following illustration shows the location of the user switches and jumpers.

1.4.1 User Switches

There are 8 user switches located on a common switchblock. A switch may be Open or Closed. The Open position is indicated on the switchblock. The Closed position is the opposite of the Open position.

- **Switch 1 (SW1): Firmware Update**
The position of this switch determines whether the module will start up in Normal Operation mode or Firmware Update mode. To start up in Normal Operation mode, set the switch to the Open position. To start up in Firmware Update mode, set the switch to the Closed position. See CHAPTER 6 for more information regarding firmware update.
- **Switch 2 (SW2): Firmware Update Method**
When the SW1 is set to the Firmware Update position, this switch selects whether the Ethernet port or an SD card is used for firmware update. To update firmware using the Ethernet port, set the switch to the Open position. To update firmware using an SD card, set the switch to the Closed position. See CHAPTER 6 for more information regarding firmware update.
- **Switch 3 (SW3): Unused**
This switch is currently unused and reserved for future use. It should be set to the Open position to prevent a future firmware enhancement that is enabled by one of these switches from inadvertently changing the operation of the module.
- **Switch 4 (SW4): Ethernet Port Isolation**
(Firmware Version 2.19 and above)
When set to the Closed position, this switch enables the Ethernet Port Isolation feature. When the Ethernet Port Isolation feature is enabled, the switch will not forward Ethernet frames between Ethernet Ports 1 and 2. Ethernet frames will be forwarded between Port 1 and Port 3 (module microprocessor) and between Port 2 and Port 3. This capability is useful for preventing Ethernet traffic from a network attached to one port from entering a network connected to the other port. When the switch is in the Open position, Ethernet frames are forwarded among all ports (1, 2, and 3).
- **Switch 5 (SW5) – Web Server Reset Enable**
When set to the Closed Position, this switch enables the module to be reset via the Module Reset web page. When the switch is in the Open position, reset via the web page is prevented.
- **Switches 5 – 8 (SW6 – SW8) – Unused**
These switches are currently unused and reserved for future use. They should be set to the Open position to prevent a future firmware enhancement that is enabled by one of these switches from inadvertently changing the operation of the module.

1.4.2 User Jumpers

User jumpers are not used on this product.

1.4.3 Shipping Configuration of Switches and Jumpers

Switch Position	Use	Shipping Position	Shipping Configuration
1	Firmware Update	Open	Normal Startup
2	Firmware Update Method	Open	Update using Ethernet Port
3	Unused	Open	
4	Ethernet Port Isolation	Open	Disabled
5	Web Server Reset Enable	Open	Disabled
6	Unused	Open	
7	Unused	Open	
8	Unused	Open	

There are no jumper options currently used on the product. All jumpers should remain in the DISABLED position.

CHAPTER 2 PROTOCOLS

The 2500P-ECC1 module supports a collection of commonly used communications protocols. Multiple protocols can be executed concurrently. Although the configuration software does not limit the number of protocols you can run at a time, the certain configurations may not be able to meet your application requirements. For example, combining a high transaction SCADA/HMI application using the CAMP server with an Open Modbus client application used to update a large I/O network would likely yield unacceptable results.

The following sections describe each protocol and present typical applications that might use the protocols. Before a given protocol can be used the 2500P-ECC1 module must be configured to use it. Protocol configuration is explained in CHAPTER 3.

2.1 CAMP Server

The CAMP Server enables client applications using the CAMP (Common ASCII Messaging Protocol) to access most controller data element types supported by the CTI 2500 Series® controllers, including loop and alarm data. First implemented on the CTI 2572 Ethernet network module, the CAMP protocol is widely used by HMI (Human Machine Interface) panels and SCADA (Supervisory Control and Data Acquisition) equipment to communicate with CTI 2500 Series® controllers and Siemens Series 505® controllers.

The CAMP protocol includes two sub-protocols: Data Transfer and Packed Task Code. The Data Transfer sub-protocol is an efficient means to read or write large blocks of data (up to 256 words). The Packed Task Code sub-protocol allows up to 15 task code requests and responses to be transferred in a block of Ethernet data. The CAMP server supports task codes commonly used to access data in the PLC and most data element types. See APPENDIX C: CAMP SERVER SUPPORT for more information.

NOTE:

*The CAMP server on the 2500P-ECC1 does **not** service task codes that program the 2500 Series® controller. Programming the 2500 Series® controller over Ethernet TCP/IP can be accomplished by connecting to the IP address of the controller's local Ethernet port rather than to the IP address of the 2550P-ECC1 module.*

The CAMP Server provides a means for client applications to monitor the status of the ECC1 module. Status Word 2048 has been designated as the ECC1 Status Word. Data for this status word is obtained from the ECC1 module rather than the Host Controller. By reading STW2048, the client can obtain status of the data caches, connection with the Host Controller, and the operational status of the Host Controller. The format of the status word is described in APPENDIX C: CAMP SERVER SUPPORT.

The 2500P-ECC1 dynamically maintains a local cache of Host Controller data items for the CAMP server. Data items are added to the cache when they are first accessed. They remain in the cache until they are no longer being accessed. Cache members that have not been accessed within 60 seconds are removed from the cache. Requests to read data are serviced directly from the cache, resulting in a substantially reduced response time. Requests to write data are always transferred immediately to the Host Controller, where the request is serviced. After the write request is completed, the corresponding cache members are updated with the new data values. The number of Host Controller data items that can be dynamically cached depends on the type of access and the diversity of the data. Worst case, the maximum number of dynamically cached items is equal to 10,250 less the number of Host Controller data addresses referenced in the Tagname Database.

The CAMP Server will allow up to 16 TCP connections. In addition, it supports UDP and UDP Multicast communications. See *Section 5.3.5* for instructions regarding configuration of the CAMP Server. Error codes that may be returned by the CAMP server are listed in Appendix A under CAMP Server Error Codes.

2.2 CAMP Client

The CAMP Client protocol provides a common means to communicate with CTI 2500 Series® controllers and Siemens Series 505® controllers via a TCP/IP network. If you are communicating with a Siemens 505® controller, it must be connected to the network using a CTI 2572 or 2572-A Ethernet TCP/IP module. If you are communicating with another CTI 2500 Series® controller, you may communicate via another 2500P-ECC1 module, a CTI 2572 or 2572-A Ethernet TCP/IP module, or directly to the Ethernet port on the 2500 Series® controller.

The CAMP client enables you to read data from or write data to a block of up to 256 V memory addresses in a CTI 2500 Series® controller or Siemens Series 505® controller. Requests to read or write data can be automatically initiated on a user-specified time interval (for example, every 500ms) or triggered by user logic in the 2500 Series® Host Controller. In addition, requests to write data can be initiated when the cached Host Controller data associated with the request changes in value.

The CAMP client can support up to 16 concurrent communication sessions using TCP, UDP, or UDP Multicast. UDP multicast enables one CAMP write request to be consumed by multiple devices.

NOTE:

The 2500P-ECC1 module implementation of UDP Multicast requires that all participating devices be connected to the same Ethernet local area network. Routing of Multicast packets is not supported.

See *Section 5.3.6* for instructions regarding configuration of the CAMP client protocol. Error codes that may be returned by the CAMP Client are listed in Appendix A under CAMP Client Error Codes.

2.3 Modbus Server

The Open Modbus Server provides a common means for a wide variety of automation equipment that support the Modbus protocol to communicate with CTI 2500 Series® controllers. You can configure one or more blocks of Host Controller data addresses to represent a block of Modbus addresses of a given type. For example, you can create an item in the Tagname Database referencing a block of V memory from V600 – V699 and associate these addresses with Modbus Holding Registers 100-199. Consequently, a Modbus request to read Holding

Register 100 would be supplied from V600. Blocks representing the following Modbus data types can be created.

Modbus Data Type	Comments
Coils	Read or Write
Holding Registers	Read or Write
Discrete Inputs	Read Only
Input Registers	Read Only

By mapping a block of Modbus addresses to a block of memory addresses in the Host controller, you can allow products that implement the Modbus TCP client protocol to access to specific data in the controller while preventing Modbus access to the remaining memory.

The Open Modbus Server can service the Modbus Function Codes indicated in the table below.

Function Code	Description	Function Code	Description
01	Read Coils	05	Write Single Coil
02	Read Discrete Inputs	06	Write Single Holding Register
03	Read Holding Registers	15	Write Multiple Coils
04	Read Input Registers	16	Write Multiple Holding Registers

The Open Modbus server supports up to 16 concurrent TCP connections. UDP is not supported. The number of data items that can be read or written in one Modbus request depends on the request Function Code (see next section for details).

See *Section 5.3.7* for instructions regarding configuration of the Open Modbus Server protocol. Error codes that may be returned by the Open Modbus Server are listed in Appendix A under Modbus Server Error Codes. For more information regarding Open Modbus go to the Modbus Organization website, <http://www.modbus.org/>

2.4 Open Modbus Client

The Open Modbus Client provides a means to communicate with products implementing a Modbus TCP server that complies with specifications published by the Modbus Organization. The Open Modbus TCP protocol is an adaptation of the popular Modbus RTU protocol network version of the Modbus RTU that allows it to communicate using TCP/IP.

Since the Modbus protocol is widely supported by a wide variety of PLCs, automation controllers, monitoring equipment, remote terminal units, and other industrial devices, it offers a common method for the exchanging data between devices from different manufacturers. In addition, the Modbus protocol is often used to communicate with process control I/O devices, such as motor starters and variable frequency drives.

The Open Modbus Client allows you to establish up to 64 TCP connections to server devices that comply with the Open Modbus standard. Alternately, you can use UDP to communicate with Modbus devices that support this IP protocol. If the server device is a Modbus Ethernet to Serial Gateway, the Open Modbus Client can communicate with serial Modbus slave devices attached to the gateway.

The Open Modbus Client can initiate requests using the Modbus Function Codes listed in the table below. The amount of data that can be transferred in a single request depends on the Function Code.

Function Code	Description	Maximum Items
01	Read Coils	2,000 Coils
02	Read Discrete Inputs	2,000 Inputs
03	Read Holding Registers	125 Holding Registers
04	Read Input Registers	125 Input Registers
05	Write Single Coil	1 Coil
06	Write Single Holding Register	1 Holding Register
15	Write Multiple Coils	1,968 Coils
16	Write Multiple Holding Registers	120 Holding Registers

See *Section 5.3.8* for instructions regarding configuration of the Modbus Client. Error codes that may be returned by the Open Modbus Client are listed in Appendix A under Modbus Client Error Codes. For more information regarding Open Modbus go to the Modbus Organization website, <http://www.modbus.org/>

2.5 Network Data Exchange Publish and Subscribe

The Network Data Exchange protocol provides an efficient means for CTI 2500 Series® controllers to coordinate control of a manufacturing process. Network data exchange uses a Publish – Subscribe model. Each 2500P-ECC1 participating in the Network Data Exchange protocol can publish data and subscribe to published data on behalf of its Host Controller.

When performing the function of a publisher, the 2500P-ECC1 monitors specified Host Controller data items, sending a particular data item to subscribers only when the value of the data item changes. When performing as a subscriber, the 2500P-ECC1 receives data to which it has subscribed and writes the data to user designated memory locations in the Host Controller.

A Network Data Exchange Publisher can support up to 20 concurrent subscribers. A Network Data Exchange Subscriber can subscribe to data from up to 20 publishers. The maximum number of data items that can be published and subscribed to is limited by the maximum number of Host Controller data items that can be in the Tagname Database (approximately 10,000).

See *Section 5.3.9* and *Section 5.3.10* for instructions regarding the configuration of the Network Data Exchange Publisher and Subscriber. Error codes that may be returned by a Network Data Exchange subscriber are listed in APPENDIX A: ERROR CODES Appendix A under Network Data Exchange Subscriber Error Codes.

CHAPTER 3 INSTALLATION

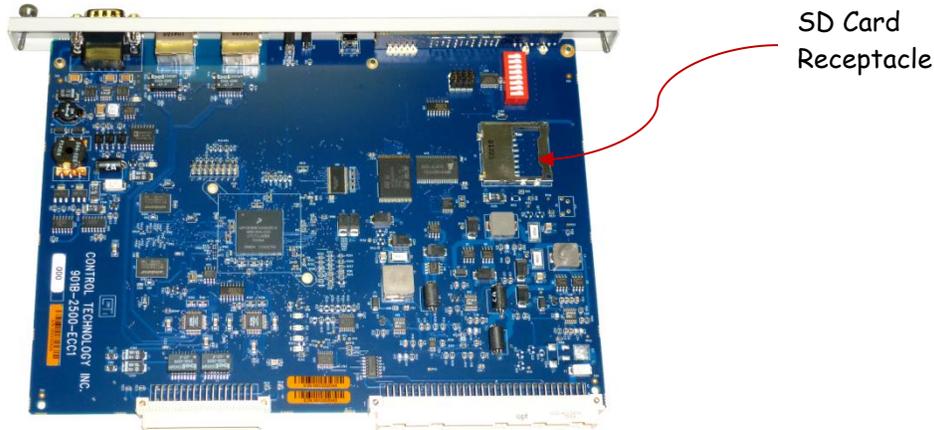
The installation of the 2500P-ECC1 consists of the following steps:

- 1) Preparing for the module installation,
- 2) Unpacking and installing the module,
- 3) Connecting the module to the Host Controller.

3.1 Installation Planning

3.1.1 SD Card Requirements

Configuration data is stored on an SD (Secure Data) card, which is inserted in a receptacle located on the 2500P-ECC1 printed circuit board. See the illustration below. The product is shipped from CTI with an SD card installed. If a card is not installed, see the following paragraphs.



Installing an SD Card

To install the SD card in the 2500P-ECC1 receptacle, remove the module from the base. Then insert the card in the receptacle face up, with the beveled edge facing the receptacle. Continue insertion until you hear a click, then release. To remove the card, apply insertion pressure until you hear a click, then release pressure.

Obtaining an SD Card

SD cards are readily available at retail outlets. CTI recommends a standard size SDHC card with 4GB or greater storage capacity and a speed rating of 4 or greater. See the following sections for more details.

Physical Size

There are three sizes of SD cards: standard, mini, and micro. The SD card receptacle on the 2500P-ECC1 module is designed for a standard size card, which is the largest form factor. A passive adapter can be used to accommodate the smaller mini or micro sizes, if necessary.

Family

There are four families of SD cards: Standard Capacity (SDSC), High Capacity (SDHC), extended capacity (SDXC), and SDIO, which combines input/output functions with data storage. The 2500P-ECC1 module can use either a standard SDSC or an SDHC card. SDSC cards use the SD logo; SDHC cards are marked SDHC. CTI recommends the SDHC card, which is a newer design.

Data Storage Capacity

Standard capacity SD cards (SDSC) provide up to 2GB (gigabytes) of data storage. SDHC cards range in capacity from 4GB to 32GB. The ECC1 Module ships with a 4GB SDHC card, which is more than adequate for storing ECC1 configuration data. However, you can use a larger capacity SDHC card, if necessary.

If you must use an SDSC card, select one with a capacity of 256MB or larger. Although the configuration data can fit on smaller capacity cards, most of these cards were produced when the SD specification was immature and may not work with the ECC1 module.

Speed

Modern SD cards indicate the access speed by a class rating, which indicates the minimum continuous write speed in MB/sec. For example, a class rating of 4 indicates an access speed of 4 MB/sec. When used to store ECC1 configuration files, the primary effect of card speed is the time it takes to transfer a new configuration to the card. Card speed also has a small effect on the time required for the module to start up, since the configuration is read from the SD card. Card speed does not affect the operating performance of the ECC1 module. Although a speed rating of 4 is adequate for most ECC1 applications, you can use cards with a faster rating.

Older SDSC cards use the CD-ROM speed rating, where 1x is the speed of an audio CD (150 Kb/s).

3.1.2 Ethernet Cabling

You will need an Ethernet cable to connect the 2500P-ECC1 module to the network. You may also need a short Ethernet cable to connect the module to the CTI 2500 Series® controller. See Section 3.2.3 for connection options. For best results you should use cables that are rated Category 5e.

3.1.3 Communications with a Host Controller

To allow the 2500P-ECC1 module to communicate with a CTI 2500 Series® controller acting as a Host Controller, the following conditions must exist:

1. The CTI 2500® Series controller must be running a firmware version that supports communications with the 2500P-ECC1 module (see the following section).
2. The module and the Host Controller must be on the same IP network, as determined by the module IP address and Network Mask. See APPENDIX B: IP ADDRESS INFORMATION for additional information regarding IP address selection,
3. The module and the Host Controller must be on the same Ethernet LAN or, if VLANs are used, the same VLAN.

3.1.4 CTI 2500 Series Controller Firmware Update

To support communications with a 2500P-ECC1 module, your CTI 2500 Series® controller may require a firmware update. The installed firmware version may be determined by accessing the controller Product Information web page or by reading Status Words 260 and 261. The Firmware Revision History for the 2500P-ECC1, located on the CTI website http://www.controltechnology.com/support/software_revision/, contains a cross reference to the firmware version required for your CTI processor. A firmware update file for the controller can be downloaded from the CTI web site, <http://www.controltechnology.com/downloads/>.

3.1.5 2500P-ECC1 Firmware Update

Although CTI installs the latest available firmware version prior to shipping a product, it is possible that a new firmware version has been released after your ECC1 module was shipped. CTI recommends that you check the firmware revision history on the CTI web site http://www.controltechnology.com/support/software_revision/ to determine whether you need to upgrade to newer firmware. If needed, a firmware update file can be downloaded from the CTI web site, <http://www.controltechnology.com/downloads/>.

3.1.6 Configuration Planning

Before the 2500P-ECC1 module can be used, you must create a configuration using the 2500P-ECC1 Configuration Program. The configuration contains the general module parameters (such as the module IP address) and operational parameters for each of the selected protocols. After creating the configuration file, you must transfer the file to your 2500P-ECC1 module. There are two methods for transferring the configuration file to the module:

- Save the configuration from your PC to an SD card; then, insert the card into the 2500P-ECC1 SD card slot.
- Connect to the 2500P-ECC1 via Ethernet TCP/IP and save the file directly to the module.

For the initial power up test, you will probably want to create a simple configuration containing the IP address of the module and the Host Controller, save it to an SD card, and then install the SD card in the module SD card slot. See Section 3.3 for instructions on creating a QuickStart configuration. See CHAPTER 5 for complete information regarding module configuration.

3.1.7 Power Requirements

The CTI 2500P-ECC1 module consumes a maximum of 5 watts of +5 VDC power. To calculate the total power required for a base, you need to add the power requirements for the other modules you will install in the base.

3.2 Installing the 2500P-ECC1 module

3.2.1 Unpacking the Module

Open the shipping carton and remove the special anti-static bag that contains the controller. After discharging any static build-up, remove the unit from the static bag. Do not discard the static bag; use it for protection against static damage when the module is not inserted into the I/O base.

CAUTION

The components on the CTI 2500P-ECC1 printed circuit card can be damaged by static electricity discharge. To prevent this damage, the module is shipped in a special anti-static bag. Static control precautions should be followed when removing the module from the bag and when handling the printed circuit card during configuration.

3.2.2 Physical Installation

Before installing the module, remove AC power from the rack. Using the guides, align the circuit board with one of the I/O slot connectors in the base. Slide the 2500P-ECC1 module into the rack until the connector seats. Then use the thumbscrews to secure the module in the rack.

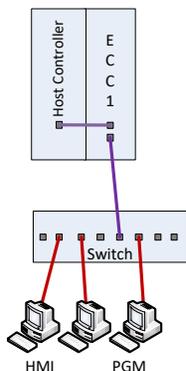


3.2.3 Connecting to the Host Controller

The 2500P-ECC1 must have an Ethernet connection to the Host Controller (the 2500 Series® controller for which it will provide communications services) and to the network containing the devices with which it will communicate.

Direct Connection to the Host Controller

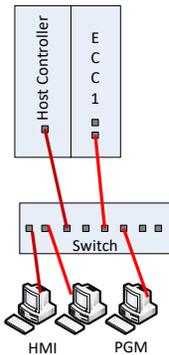
If the 2500P-ECC1 is installed in the same base as the Host controller, you may choose to connect a short Ethernet cable (Cat 5 or Cat 5e) from one of the Ethernet ports on the 2500P-ECC1 module to the Ethernet port on the 2500 Series® controller and connect a second Ethernet cable from the other port to a port on a network switch that provides a path to the HMI or other devices as shown in the accompanying illustration. In this configuration, Ethernet packets between the programming workstation and the host controller are switched between the ECC1 Ethernet ports.



An advantage of this method is that the embedded switch on the 2500P-ECC1 can provide broadcast storm protection for the 2500 Series® programmable controller. See Section 5.3.2 Entering ECC1 Settings Advanced Tab.

Alternately, you may choose to connect both the ECC1 module and the Host Controller to the network Ethernet switch as shown in the next section.

Connection via a Network Switch



If you are connecting more than one 2500P-ECC1 module to the same Host Controller, you should connect through an external Ethernet switch rather than daisy chaining the connections between modules. Although the modules will communicate successfully if daisy chained, this could cause unnecessary communications interruptions, since resetting one module (required after a configuration download) will temporarily interrupt communications with the network.

Using this method, all communications between the HMI and the 2500P-ECC1 module and between the 2500P-ECC1 module and the 2500 Series® controller pass through the network switch. In addition, communications between the programming workstation and the 2500 Series® controller pass through the network switch.

CAUTION:

When this method is used, the other Ethernet port on the 2500P-ECC1 SHOULD NOT be connected to the network unless [Ethernet Port Isolation](#) is enabled (SW4). Doing so could create a loop, potentially disrupting network communications.

3.3 QuickStart Configuration

The following procedure provides a quick means for creating a configuration that can be used to test the 2500P-ECC1 using the CAMP server. It consists of the following steps.

1. Entering the IP address and subnet mask of the ECC1 module,
2. Entering the IP address of the Host Controller,
3. Enabling the CAMP server,
4. Saving the configuration to an SD card.

See CHAPTER 5 for information regarding the CTI 2500P-ECC1 Configuration Program

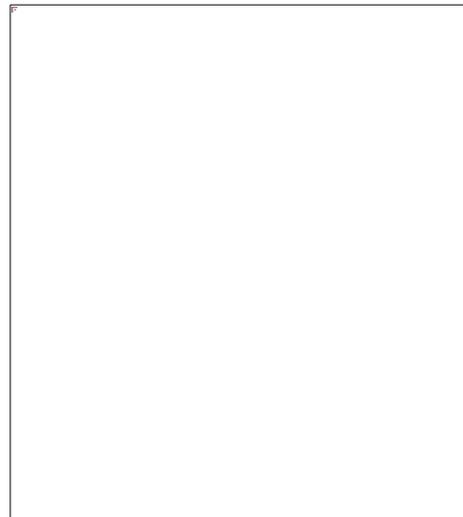
After starting the configuration program you should be presented with a dialog box that asks you to Start a New Project, Open an Existing Project, or Cancel. Select the “Start a New Project” option. If the dialog box does not appear, select File/New from the menu bar.

Once you have created a new project, you will be presented with the ECC1 Settings dialog box.

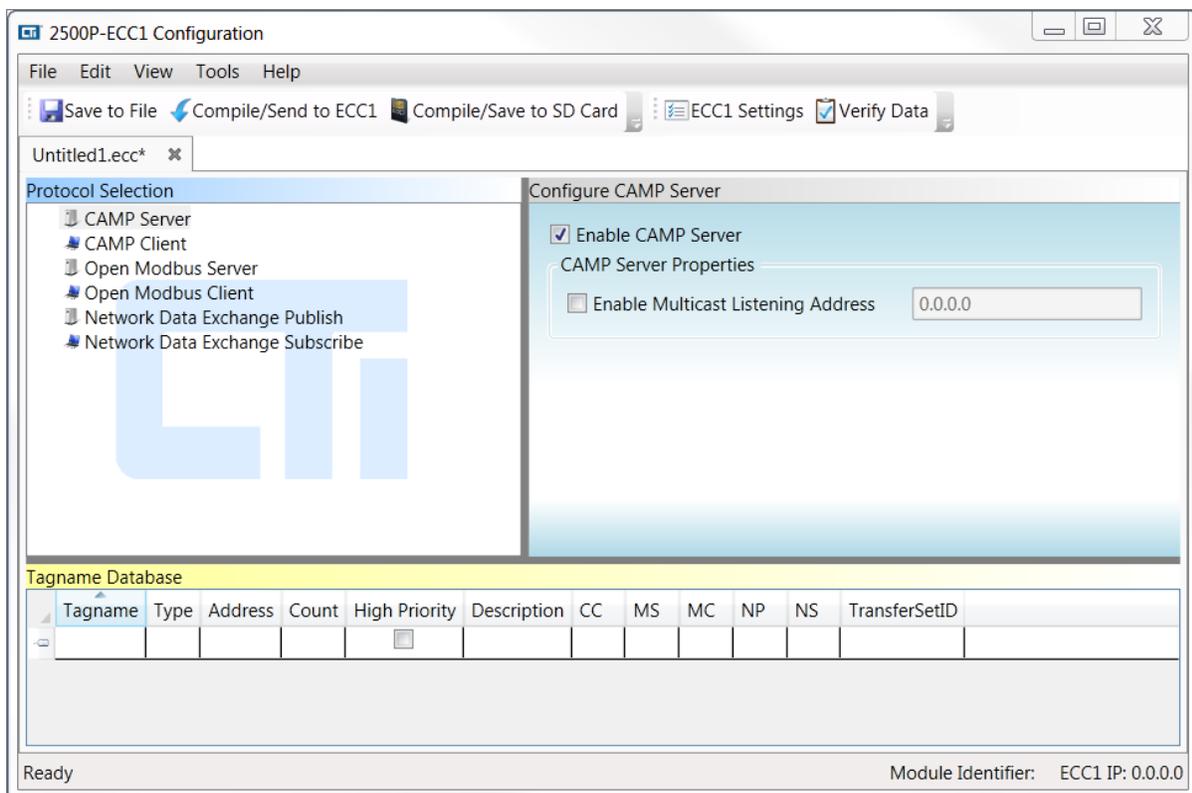
Under “ECC1 Network Settings” heading enter the IP address and the subnet mask for your ECC1 module.

Under the Host Controller Settings heading, enter the IP address assigned to the Ethernet port of your CTI 2500 Series® controller. *Note that the 2500P-ECC1 module and the CTI 2500 Series® controller must be on the same IP network.*

When you have finished entering this information, click on the OK button.



Next, click on the CAMP Server object in protocol selection panel. The Configure CAMP Server configuration parameters will appear in the Configuration panel. Click on the Enable CAMP Server check box to enable the server as shown below.



Click on the Save to File option in the toolbar to name and save the configuration project. Then insert the SD card into your PC and click on the Compile/Save to SD Card. If you don't have an SD card slot on your PC (most don't), you will need to obtain an SD card to USB adapter, available at most stores that carry computer supplies. This adapter allows you to plug the SD card into your computer's USB connector.

After saving the configuration files to the SD card, insert the card into the SD card receptacle on the 2500P-ECC1 module. See Section 3.1.1 for the location of the SD card.

As an alternate to saving the configuration to an SD card, you can transfer the configuration directly to the 2500P-ECC1 module via Ethernet. However, this requires that the IP address of your PC and the IP address of the module be on the same IP network or that there be a gateway path between the two IP networks. If this is not the case, it is much easier to configure the ECC1 module the first time using the SD card. Once the IP address has been set to an IP address that is accessible from your PC, subsequent configurations can be easily downloaded via the Ethernet port.

If the module IP address is not accessible from your PC and saving the configuration to an SD card is not a viable option, you must change the IP address and subnet mask of your PC to match the IP network of the module. See APPENDIX B: IP ADDRESS INFORMATION for more information.

CHAPTER 4 OPERATION

4.1 Module Startup

Firmware Loading and Execution

When power is applied or the ECC1 module is reset, the 2500P-ECC1 module will attempt to load the module firmware from flash and begin execution. At the beginning of this process, the module will turn on the software controlled LEDs and all multi-segment display (MSD) segments to allow you to verify that the LEDs and MSD are functioning correctly. After one second, the LEDs and MSD segments will be turned off.

During this process, any IP parameters that have been previously stored in flash will be read. If no IP parameters have been stored in flash, a temporary link-local IP address will be automatically generated. If the flash contains an IP address, the module will attempt to determine whether another on the local area network is using this address. If a duplicate IP address is detected, the module will alert you by turning the Network Status (NS) LED red. Otherwise, the NS LED will remain green. In either case, the startup process will continue.

If an error is encountered while loading or initializing the firmware, an error code will be displayed on the MSD and startup process will be suspended. See *Initial Startup Error Codes* in Appendix A for a description of the startup error codes. See Section 1.2 for front panel locations of the MSD and other indicators. To proceed, you must correct the problem and restart the module

Configuration File Processing

Once the application firmware is successfully executing, the ECC1 module will begin flashing the **Good** LED and attempt to read the configuration file from the SD card. If an error is encountered while reading or initializing the configuration, the ECC1 module will alternate between displaying an error code and the module IP Address on the Multi-Segment Display (MSD) and module startup will be suspended. See *Operational Error Codes* in Appendix A for a description of the error codes. To proceed, you must correct the problem.

If a configuration file can be read, the ECC1 module IP parameters obtained from the configuration will be compared with the parameters previously obtained from flash memory. If they are different, the new values will be written to flash memory and the module will restart using the new parameters. If they are the same, the IP address will be displayed on the MSD and the startup process will continue. **When the configuration has been successfully read and initialized, the module will turn the Good LED on.**

NOTE:

Even though no duplicate IP address is detected, it is possible for a network host device to be using the same IP address as the 2500P-ECC1.

Host Controller Communication

If the configuration file does not contain an IP address for the Host Controller, startup procedure will halt, the Host LED will remain off, and an error code will be displayed on the MSD. You will need to download a configuration containing a Host Controller IP Address to proceed.

When the configuration file contains a Host Controller IP address, the ECC1 module will begin flashing the Host LED and attempt to connect to the Host Controller. If the ECC1 module is unable to connect to, register with, or access data in the Host Controller, an error message will be displayed on the MSD and the startup process will be suspended. See *Operational Error Codes* in Appendix A for a description of the error codes. To proceed, you must correct the problem. You must correct the error before the 2500P-ECC1 module will attempt to activate the configured protocols. **When the module is successfully communicating with the Host Controller, the Host LED will be turned on.**

Protocol Activation

Once the 2500P-ECC1 module has successfully registered with the Host Controller, it will attempt to execute protocols that have been configured. If no protocols were configured, the **ACTIVE** LED will remain off. The Active LED also will remain off while the Host Controller is in Fatal Error mode or while the Host Controller is in Program mode when the option to “Disable Protocols when Host Controller is in Program Mode” has been selected. Otherwise, the ECC1 module will begin activating the configured protocol(s).

If any of the configured protocols reports an error, the Active LED will begin flashing and the protocol reporting the error will be identified by displaying an error code MSD.

NOTE:

Multiple protocols could be reporting an error, however only one of the protocols will be identified on the MSD. To see all protocols that are reporting errors, browse to the Error Code Descriptions page of the ECC1 module web server.

Potential errors include:

- The module is configured to access a memory address not available in the Host Controller,
- The protocol cannot start up successfully,
- A client protocol cannot connect to a network device as configured,
- A Network Data Exchange subscriber cannot connect to a publisher as configured.

When all configured protocols are operational (not reporting an error), the module will set the ACTIVE LED on.

NOTE:

Client Protocols do not report all errors, especially those included in error replies from the network device with which the client is communicating. Thus, it is possible for a data access error to exist, even though no error is displayed on the MSD. These errors can be detected by mapping the “Error Status” word to a Host Controller memory location. See Section 5.3.6 and 5.3.8 for more information.

4.2 Normal Operation

When the 2500P-ECC1 module is fully operational, the front panel indicators and displays should be in the state indicated in the following table:

Indicator Display	State	Comments
Good LED	On (solid)	The ECC1 module is running using a valid configuration.
Host LED	On (solid)	The ECC1 module is successfully communicating with Host Controller.
Active LED	On (solid)	All configured protocols are operational.
NS LED	On (Green)	TCP/IP stack is operational.
XMT/RCV LEDs	Flashing	The 2500P-ECC1 is transmitting and/or receiving data
Multi Segment Display	Displaying Module IP Address	
	No error message is displayed	
Port 1 Link	On, if this Ethernet link is connected.	An Ethernet transmit/receive path is available.
Port 1 Act	Flashing, if there is network activity	Data is being transmitted by a device on the network, not necessarily by this module.
Port 2 Link	On, if this Ethernet link is connected.	An Ethernet transmit/receive path is available.
Port 2 Act	Flashing, if there is network activity	Data is being transmitted by a device on the network, not necessarily by this module.

4.3 Cache Update Performance Tuning

Whether or not you will need to be concerned about performance tuning will depend on your particular application needs. For many applications, the default settings of the 2500P-ECC1 will not need to be changed. The communications protocol used by the 2500P-ECC1 module to access Host Controller data is designed to minimize the workload on the Host Controller. In addition, all the workload related to communicating with external devices, such as HMI, SCADA systems, and automation controllers is offloaded to the 2500P-ECC1 module.

Nevertheless, some applications where Host Controller scan time is critical or where very fast data updates are important may require adjustments of certain parameters to achieve desired results. This section is intended to provide guidance to achieving your process objectives.

4.3.1 Configuration Parameters That Affect Performance

- Host Controller Time Slice.** The Host Controller time slice is the *maximum* amount of time per scan that the Host controller will allocate to processing requests from this module. This places an upper bound on the increase in scan time caused by communications with the module. If communications processing consumes less than the time slice, the scan will be extended only by the amount of time needed. Once the maximum time is used, further communications processing by the Host controller is deferred until the next scan. The Host Controller time slice is set during module configuration. See

-
- *Entering ECC1 Settings* in the following Chapter.
- **Host Controller Scan Time.** Because Host Controller services data access requests in a time slice that occurs once per scan, the frequency of cache updates is limited by the scan time. For example, if the Host Controller scan time is 150ms, the actual cache refresh interval can be no faster than 150ms. If you set the Cache Refresh Interval to 100ms, the cache will be marked as not current, even though you may be updating the cache every scan.
- **Cache Refresh Interval.** The Cache Refresh Interval (CRI) determines how often the cache on the 2500P-ECC1 module will be updated with new data from the Host Controller. The smaller the CRI, the more often the module requests data to update the cache from the Host Controller, which increases the communications processing load on the Host Controller and the ECC1 module.

The 2500P-ECC1 module uses two user specified Cache Refresh Intervals, Normal and High Priority. As the names imply, the Normal CRI is intended to be used to update most of the items in the data cache and the High Priority CRI is intended for those items you need to be updated faster than normal. You can specify the value of the Cache Refresh Intervals during module configuration. See

Entering ECC1 Settings in the following Chapter. You can also specify which items will be updated using the High Priority CRI. See *Entering Tagname Database Data Items* in the next Chapter.

Number of Data Cache Members. For all Protocols except CAMP Server, the number of data cache members is determined by the number of Tagname Database items entered during configuration and the value of the Count parameter of each item. See *Entering Tagname Database Data Items* in the next Chapter. For the CAMP Server, the number of cache members is dynamically managed based on usage. An increase in the number of data cache members increases the amount of work required by the Host Controller to process data requests to keep the cache current.

4.3.2 Determining Operating Status

There are several facilities available to determine operating status.

ECC1 Status Word

The ECC1 Status Word contains a set of bits which indicate the status of various ECC1 operating properties, including the status of the caches. The CAMP server allows external clients to access the ECC1 status word as STW2048. See APPENDIX C: CAMP SERVER SUPPORT for more information. Other ECC1 protocols can provide access to the ECC1 status word by mapping the tagname “_ECC1_Status_Word” to a protocol data word. For example, when using Modbus, you could map this tagname to a Holding Register address.

The format of the ECC1 Status Word is shown below. Bit 1 is the least significant bit. Unused bits, indicated by 0, are reserved for future use.

Bit	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	0	0	0	0	0	0	0	0	0	0	A	B	C	D	0	0

A	Invalid Host Controller Memory Address	0 = All items in the Tag Database reference valid Host Controller memory addresses. 1 = Tag Database contains one or more items that reference memory addresses exceed the address range of the Host Controller.
B	Priority Cache Status	0 = Current – All members of the cache are being updated within the designated cache refresh interval. 1 = Stale – some members of the cache are not being updated within the designated cache refresh interval
C	Normal Cache Status	0 = Current – All members of the cache are being updated within the designated cache refresh interval. 1 = Stale – some members of the cache are not being updated within the designated cache refresh interval
D	Host Controller Mode	0 = Host Controller is in Run Mode 1 = Host Controller is in Program Mode or status is unknown

NOTE:
The ECC1 Status Word is not accessible when the Host Controller is disconnected or is in Fatal Error Mode, since either of these conditions disables the communications protocols.

See Section 5.3.2 for information on enabling mapping of the ECC1 Status Word.

2500P-ECC1 Web Server Page

The 2500P-ECC1 module contains an embedded web server facility. You can access this facility by typing the IP address of the target module into the URL window of your browser. The Host Controller Data Cache Statistics page provides statistics regarding the data cache. Among the information presented is the status of the various data caches, which allow you to determine whether cache members are being updated within their specified cache refresh interval. If not, you can determine the number of items that are not being updated in time.

2500 Series PLC Web Server Page

The 2500 Series® controller also contains an embedded web server facility. The PLC Scan Statistics page allows you to determine the impact of servicing requests from the 2500P-ECC1 module(s) on the controller scan time. The time used consists of the DataServControl element and one element for each connected ECC1 module.

4.3.3 Tuning the 2500P-ECC1 Operation

For most applications, it is not necessary to tune the ECC1 module operation. However, for those applications requiring tuning, the approach you should take depends on whether you need to minimize the impact on the Host Controller scan time or not. See the applicable section below.

Applications Requiring Minimum Scan Time Impact

If you need to limit the impact of ECC1 communications on Host Controller scan time, you should start by setting the Host Controller Time Slice to the maximum time you will allow a scan to be extended. See *Section 5.3.2 Entering ECC1 Settings*. There is approximately 1ms of fixed scan overhead incurred when one or more 2500P-ECC1 modules are communicating with the Host Controller. Thus, if you want to limit the scan time impact to 10ms and are using one ECC1 module, you should set the Host Controller Time Slice to 9ms.

After transferring the configuration and allowing the module to complete startup, connect to the Host Controller web server and navigate to the PLC Scan Statistics page in the will allow you to verify that the peak scan time is within bounds. Observe whether the cache of items assigned to the Normal CRI and High Priority CRI are current. If both caches are current, no additional changes are necessary.

If the cache of items assigned to the High Priority CRI is stale, then you will need to increase the High Priority CRI or reduce the number of items assigned to the High Priority CRI. Confirm that the High Priority CRI is not smaller than the Host controller scan time. If the High Priority CRI is smaller, you should increase it to a value greater than the Host Controller peak scan time.

If the cache of items assigned to the Normal CRI is stale, then you will need to increase the Normal CRI until the cache is current. If this does not result in an acceptable Normal CRI value, then you will need to compromise on the Host Controller time slice or the High Priority CRI.

Applications where Scan Time Impact is not a Concern

In this situation, at least one of the data caches is reporting stale, which indicates that some items not being updated within the configured Cache Refresh Interval (CRI) and you are willing to dedicate additional Host Controller scan time to correct the problem. In this case, you should set the Host Controller Time Slice to the maximum possible value. See *Section 5.3.2 Entering ECC1 Settings*. After making this change, download the configuration, restart the module, and access the Host Controller Data Cache Statistics web page to determine whether the cache is current. If the caches are current, you will need to make no further changes.

If the cache is still stale, then you will need to increase one or both of the Cache Refresh Intervals until the caches are current. Alternately, you could split the workload between two 2500P-ECC1 modules.

4.4 Troubleshooting Operational Problems

This section provides guidance in troubleshooting and correcting operational problems. Operational errors include:

- Errors in loading and executing module firmware,
- Errors in reading user configuration files,
- Errors in connecting to and communicating with the Host Controller,
- Errors in accessing the network and establishing network connections,
- Errors in updating module firmware.

NOTE:

Errors returned by the ECC1 protocols are described in the sections of this manual related to the specific protocol

There are several sources of information that can be used to troubleshoot problems.

MSD Error Codes

When an operational error occurs, an error code will be displayed on the Multi-Segment Display. A list of error code descriptions and associated recommendations for resolving the errors can be found in the Operational Error Codes section in Appendix A.

Web Server

The 2500P-ECC1 module contains an embedded web server, which can be accessed by typing the IP address of the module in your browser's URL box. Each of the web server pages can be saved to a file by clicking on the File menu item and selecting the "SAVE AS" option. Once the page has been saved, it can be emailed to CTI. Alternately, if the PC you are using to view the file has email access, you can email the page without first saving it by clicking on the File menu item and selecting the "SEND" option.

Starting with firmware version 2.08, you can select the "Display All Statistics" option from the web page menu to consolidate all web server files. After this has been done, you can save or email all web pages by selecting the "SAVE AS" or "SEND" option.

Event Log Page

The event log is a file of significant events ordered by time. This can be used to determine what was happening when the error occurred, often providing clues to the cause. CTI customer support representatives will usually need this information when assisting you.

Product Information Page

This page contains information about the module, including serial number, MAC address, current IP address and subnet mask, firmware version, and switch settings. CTI customer support representatives will usually need this information when assisting you.

TCP/IP Statistics (V2.05 and above)

This page displays summary data regarding the TCP and UDP communications of each ECC1 protocol. This information is useful when diagnosing TCP/IP communications problems.

Ethernet Port Statistics

This page displays statistics related to the Ethernet packets received and transmitted by the ECC1 Ethernet controller. This information is useful when diagnosing network problems.

Active Communications Sessions

This page contains information about each active TCP and UDP session initiated by the CAMP client, Modbus client, and Network Data Exchange subscriber. This information is useful when diagnosing problems communicating with a particular device. **This page contains information about each active TCP and UDP session initiated by the CAMP client, Modbus client, Modbus Server, Network Data Exchange subscriber, and the Network Data Exchange publisher. Note that the Camp Server has a separate web page.**

Communications Session History

This page contains data regarding previously active communications sessions that have been terminated for some reason. This information is useful when diagnosing problems communicating with a particular device.

CAMP Server Statistics (V2.05 and above)

This page contains data regarding the operation of the CAMP server. It also keeps a count of common CAMP and Task code errors returned to the client, providing a means to diagnose problems related to the client requests. This page is very useful in diagnosing errors encountered when communicating with SCADA and HMI systems, which often provide limited error code feedback to the user.

Host Controller Data Cache Statistics

This page contains information about the Host Controller, communications between the ECC1 module and the Host controller, and the status of the data caches. Although much of this information is intended for CTI developers and support personnel, the data cache information allows you to determine the status of the data cache.

Error Code Description Page

This page lists the Operational Error codes along with a description of the error. In addition, it indicates which errors are currently being reported and maintains a cumulative count of occurrences by error code. Unlike the Multi-Segment Display, which shows only the highest priority error, this page allows you to determine all the operational errors that active.

Switch Statistics

This page reports the diagnostic statistics maintained by the ECC1 Ethernet switch and the value of the switch registers. This page is primarily for use by CTI developers.

Display All Statistics (V2.08 and above)

This option collects and displays information in all web pages. It is used to consolidate web pages so that diagnostic information can be saved and emailed to CTI in one operation.

Product Support

This is a link to the CTI web site.

NOTE:

The "Product Clock" value displayed in web page statistics is derived from the clock of the Host Controller.

The time and date shown in web page headings is obtained from the requesting PC.

2500P-ECC1 Configuration Program Help Text

The help text source is essentially a copy of this user manual with links from the program. Appendix A contains a list of all error codes with a description of the error and recommendations for resolving the problem causing the error.

CHAPTER 5 CONFIGURATION

5.1 Configuration Overview

Before the 2500P-ECC1 module can be used, it must be configured for your particular application requirements. This is accomplished using the 2500P-ECC1 Configuration Program.

The 2500P-ECC1 Configuration Program allows you to:

- Create a configuration project, specifying the module and protocol specific parameters,
- Save the configuration project file to a local or network drive,
- Compile and send the configuration to the 2500P-ECC1 via an Ethernet link,
- Compile and save the configuration to an SD card, which can be inserted into the ECC1 SD card slot.
- Update the 2500P-ECC1 module firmware.

The program will run under the following Microsoft Windows versions:

- Windows Vista client (32 or 64 bit),
- Windows 7 client (32 or 64 bit),
- Windows 8 client (32 or 64 bit),
- Windows 10 client (32 or 64 bit).

5.2 Installing the 2500P-ECC1 Configuration Program

You can obtain the latest copy of the 2500P-ECC1 configuration program from the CTI web site www.controltechnology.com/downloads.

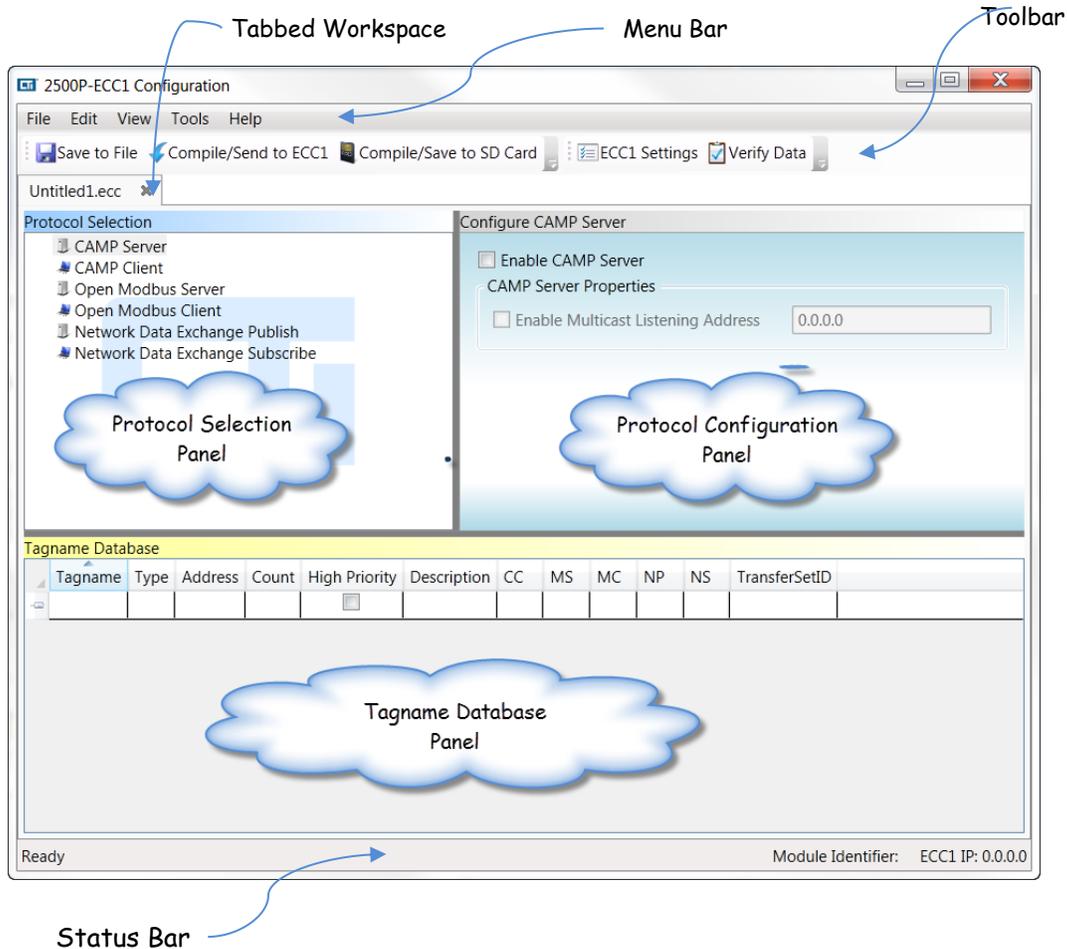
After copying the related Setup file to a local drive, click on the file and select Run. The installation program will create the required files on your PC.

5.3 Using the 2500P-ECC1 Configuration Program

This section describes the configuration program facilities and how to use them. It also explains how to configure the protocols supported by the ECC1 module.

5.3.1 Configuration Program Main Window

After the 2500P-ECC1 Program starts, the main application window will be displayed. The window contains a menu bar, a toolbar, a tabbed workspace consisting of three panels, and a status bar as shown below.



Menu Bar

The menu bar includes contains pull down menu items that allow you to save and restore file, perform editing tasks, view configuration program items, access utility tools, and to access program help.

- **File**
 - **New:** Creates a new (unnamed) configuration project file
 - **Open:** Opens an existing configuration project file on a hard drive, an ECC1 Module, or an SD card
 - **Close:** Closes the currently active configuration project file
 - **Save:** Saves the active configuration project file to a hard drive, ECC1 Module, or an SD card
 - **Save As:** Saves the active configuration project file to a hard drive with a new name

- **Edit**
 - **Undo:** Reverses previous action
 - **Redo:** Cancels the previous Undo
 - **Cut:** Deletes the selected item and copies it to the Clipboard
 - **Copy:** Copies the selected item to the Clipboard
 - **Paste:** Pastes an item previously saved to the Clipboard
 - **Delete:** Deletes the selected item

- **View**
 - **Settings:** Displays the Settings dialog box
 - **Compilation Report:** Displays a detailed report of the previous compilation

- **Tools**
 - **Verify:** Checks the current configuration for errors
 - **Firmware Update:** Updates the ECC1 Module firmware
 - **Options:** Sets program options, such as default folders

- **Help**
 - **Help:** Displays the help text
 - **About:** Displays information about the 2500P-ECC1 Configuration Program

Toolbar

The toolbar provides a shortcut to commonly invoked actions.

- **Save to File:** Saves the active configuration project to a file
- **Compile/Send to ECC1:** Compiles the configuration project and transfers the resulting files to a 2500P-ECC1 module via Ethernet TCP/IP
- **Compile/Save to SD Card:** Compiles the configuration project and saves it to an SD card
- **ECC1 Settings:** Displays a dialog box that allows you to specify global ECC1 parameters, such as the module IP address.
- **Verify Data:** Checks the current configuration project for errors.

Tabbed Workspace

The tabbed workspace allows you to create and modify configuration data. Multiple tabs can be displayed, one for each named configuration file. The tabbed workspace consists of three panels, the Protocol Selection Panel, the Configuration Panel, and the Tagname Database Panel.

- **Protocol Selection Panel**

The Protocol Selection Panel allows you to select a protocol and related objects to be configured.
- **Configuration Panel**

The Configuration Panel displays the configuration parameters related to the configuration object that is selected in the Protocol Selection panel. For example, when the CAMP Server is highlighted, the configuration panel displays parameters that allow you to enable the server and to specify a multicast listening address.

- **Tagname Database**

The Tagname Database Panel allows you to assign tagnames to Host Controller data items. The tagnames are used to map the Host Controller data to the protocol data. A tagname may be assigned to a single Host Controller memory address such as V100, or a block of Host Controller memory addresses, by specifying a count representing of data elements in the block. This panel also allows you to select particular data items for high priority update and to enter a short description for the data items.

Status Bar

The Status Bar provides information about the active configuration project.

5.3.2 Entering ECC1 Settings

The ECC1 settings window allows you configure the global module operating parameters. This section describes the use of the various settings.

General Tab

The General tab allows the user to enter descriptive information, specify the module IP parameters, and to configure the connection to the Host Controller.

Module Identifier: This optional field allows you to assign a unique identifier (up to 16 characters) to the module. This identifier will be displayed in the 2500P-ECC1 Web Product Information Page and the Host Controller “PLC Scan Statistics” Web page.

Location: This field allows you to document the product location or other pertinent application information. This will be displayed in 2500P-ECC1 Web Product Information page. This field is optional. You may enter up to 40 characters, which are displayed in the 2500P-ECC1 Product Information Web page.

Configuration Description: This field allows you to describe the configuration file being used. This is often helpful in the startup phase of an automation project where some communications may be disabled. This field is optional. You may enter up to 40 characters, which are displayed in the 2500P-ECC1 Product Information Web page.

ECC1 Network Settings

IP Address: This parameter specifies the IP address you wish to assign to the 2500P-ECC1. It must be a valid IPV4 unicast address or 0.0.0.0, which will cause the IP address of the ECC1 module to be cleared. See APPENDIX B: IP ADDRESS INFORMATION for further information regarding IP addresses.

Subnet Mask: After you enter a valid IP unicast address, the 2500P-ECC1 will automatically assign a subnet mask that corresponds to the address class of the IP address you entered. You may change this value, if necessary, to match your network requirements.

Default Gateway: This parameter specifies the IP Address of the default gateway. A value of 0.0.0.0 indicates no gateway is used. A gateway address is necessary only if you are communicating with devices on another IP network or wish to allow access from another network to download configuration files or browse the module web server diagnostic information.

Alias IP Address: This parameter allows you to assign an additional IP address to the Ethernet interface.

Alias Subnet Mask: This is the subnet mask associated with the Alias IP Address. When an Alias IP address and Alias Subnet mask is assigned, the module will be able to communicate on the Alias IP subnet in addition to the primary IP subnet defined by the Module IP address and subnet mask. For details see APPENDIX F: ALIAS IP FEATURE.

Important Note on Default Gateway: the gateway will operate with either the Primary or Alias IP, but not both – since by specification the primary and alias IPs must be on different subnets. If the gateway IP is a member of the primary subnet, it will forward to the primary address. If it's a member of the alias subnet, it will forward to the alias IP.

Host Controller Settings

Interface Type: This parameter specifies how the ECC1 module is connected to the Host Controller. When connecting with CTI 2500 Series® controllers model C100 – C400, you should always select LAN option.

IP Address: This parameter specifies the IP address of the Host Controller with which the ECC1 will communicate. The ECC1 module and the Host Controller must be on the same Ethernet network and IP subnet.

Time Slice: The Host Controller processes data access requests from 2500P-ECC1 modules in a dedicated time slice. The Time Slice parameter allows you to limit the effect of processing the requests on the Host Controller scan time. The time slice value represents the *maximum* amount of time (in milliseconds) that the controller scan will be extended to process data requests from this ECC1 module. If the time slice maximum is reached and additional requests are pending, the Host Controller will defer processing the additional requests from this module until the next scan. If requests can be serviced in less time, the Host Controller scan will be extended only by the time required. When multiple 2500P-ECC1 modules are installed, each module will be allotted scan time based on the module's time slice value.

If the scan time extension is not critical to the process being controller, CTI recommends that you use the default time slice (15ms). If the time slice needs to be reduced or the data cache cannot be maintained in a current state, see Section 4.3 for additional information regarding performance tuning.

Advanced Tab

ECC1 Cache Refresh Intervals

The cache refresh interval (CRI) specifies how often members of the ECC1 data cache are updated with new data. There are two CRI categories, Normal and High Priority. Data cache members are assigned to the Normal CRI category unless the High Priority is selected for the associated Tagname Database item. *You should avoid setting the CRI values to arbitrarily small values, since this creates unnecessary workloads for the module and the Host Controller and may result in configurations that cannot maintain the cache at a current state.*

Normal: This parameter specifies the requested cache refresh interval (in milliseconds) for the Normal CRI Category. Most of the items should be assigned to this category. For SCADA applications, this should be set to a value ranging from 1000ms to 500ms.

High Priority: This parameter specifies the requested cache refresh interval (in milliseconds) for the High Priority Category. Only items that require an update faster than the Normal category CRI should be assigned to High Priority category.

Host Controller Connection Status Bit (STW267): Host Controller user logic can monitor the connection status of 2500P-ECC1 modules by reading STW267. In this status word, bits 1 – 8 are used to monitor the data cache connection status of up to eight Advanced Function (AF modules) AF modules include the 2500P-ECC1 and 2500P-ACP1 modules. When a bit corresponding to a particular ECC1 module is set to 1, the module is successfully communicating with the controller. This parameter designates which bit (1-8) of STW 267 will be used to monitor the connection status of **this** ECC1. If you are using more than one AF module, each module must select a different bit. If more than one AF module is attempting to use the same connection bit, bit 16 of STW267 will be set. If Host Controller monitoring is not required, select the “Not Used” option. Note that bit 1 is the most significant bit of a status word (STW).

Enable Mapping of ECC1 Status Word: In the event that devices communicating with the 2500P-ECC1 need to monitor the status of the module, this can be accomplished by mapping the reserved tagname “_ECC1_Status_Word” to the protocol data. Checking this box makes the reserved tagname appear in the list of tagnames that can be mapped. STW2048 in Appendix C.

Disable Protocols when Host Controller is in Program Mode: Checking this box will cause the ECC1 to shut down all active protocols (except Camp server) when the Host controller is placed in Program mode. This option is intended for applications where the ECC1 module is used to update network based I/O, allowing the process to be placed in a safe state while the Host Controller is in program mode. When this box is checked, [Error 320](#) will be displayed when the Host Controller is in Program mode.

The screenshot shows the 'ECC1 Settings' dialog box with the 'Advanced' tab selected. The 'ECC1 Cache Refresh Intervals' section has 'Normal' set to 1000 ms and 'High Priority' set to 500 ms. The 'Host Controller Connection Status Bit (STW267)' is set to 'Not Used'. Under 'Ethernet Port Properties', 'Use Defaults', 'Enable Broadcast Storm Protection', and 'Include Multicast' are checked, with a 'Rate Limit (% of 10/100 bandwidth)' set to 2%. Under 'Remote Reset', 'Enable' is checked and 'Control Relay' is set to 100. 'OK' and 'Cancel' buttons are at the bottom right.

NOTE:
This action will affect all active protocols except CAMP server.

Ethernet Port Properties - Broadcast Storm Protection

An Ethernet Broadcast Storm is a condition where an excessive rate of broadcast packets is being transmitted on the network. A broadcast storm can adversely affect the operation of equipment connected to the network, since additional resources are consumed to process the broadcast packets.

Broadcast Storm Protection limits the rate at which broadcast packets arriving at one of the external ports are forwarded to the ECC1 module microprocessor and to the other external port. Packets not forwarded are discarded. If you are connecting the Host Controller directly to an ECC1 Ethernet port, the rate at which broadcast packets are forwarded to the Host Controller Ethernet port is also limited. Unicast packets are unaffected by the storm protection.

When the Use Defaults box is checked, the storm protection parameters are set to enable storm protection and to include multicast frames when calculating the rate. Version 1.5 and above of the 2500P-ECC1 Configuration program will set the default rate limit to 2%. Previous versions set the default rate limit to 10%.

When the Use Defaults box is unchecked, you can choose to enable or disable Broadcast Storm Protection by checking/unchecking the associated box. When storm protection is enabled, you can choose to include multicast packets in the storm protection algorithm and set the rate limit.

Note:
Firmware versions prior to V2.05 used the following static broadcast storm parameters: Broadcast Protection Enabled, Rate Limit = 10%, Include Multicast.

The rate limit is expressed as a percent of the network bandwidth (roughly equal to the network speed). The default rate is currently set to 10%. When the ECC1 port is connected to a 100Mb Ethernet network, a value of 10% would limit the broadcast rate to 10Mb (1.25 MB/sec). Considering inter-packet overhead, this translates to approximately 15,000 64 byte packets per second. If you are encountering network storm problems, a value of 1 – 2% should provide adequate protection.

Remote Reset

Remote reset allows the Host PLC program to trigger a reset of the ECC1 module on a FALSE to TRUE transition of a Control Relay.

To use this feature, check the Enable Box and enter the Control Relay number to be used.

Note:
Remote Reset is only available on ECC1 Firmware versions V2.26 and above. Also, a working connection between the ECC1 and Host PLC is required for the Remote Reset to function.

5.3.3 Entering Tagname Database Data Items

Contents

Entries in the Tagname Database panel identify Host Controller data items that will be used by the ECC1 protocols. For each data Item, you designate a tagname, which will be used to map the controller data to the protocol data during protocol configuration. A tagname can represent a single data address in the Host Controller or a block of contiguous data addresses. The Tagname Database can contain numerous entries representing a maximum of 10,000 Host Controller data memory addresses.

Items accessed by the CAMP Server protocol do not require an entry in the Tagname Database, since the CAMP Server does not require mapping to access the Host Controller data elements. Data accessed by the CAMP Server is dynamically cached when accessed and uses the Normal Cache Refresh Interval. ***If you want specific Host Controller data accessed by the CAMP server to be assigned to the High Priority cache refresh interval, you must create an entry for the data items and check the High Priority box.***

Tagname Database											
Tagname	Type	Address	Count	High Priority	Description	CC	MS	MC	NP	NS	TransferSetID
My_PLC_V100_Block	V	100	20	<input type="checkbox"/>		<input checked="" type="checkbox"/>					0
				<input type="checkbox"/>							

For each Host Controller item that you want to access, enter the following information:

- **Tagname:** The Tagname is a name used to identify the data item. It is used to associate (map) the data item to protocol data. The tagname must comply with the following rules:
 - A tagname can consist only of upper and lower case letters (a-z, A-Z), numbers (0-9), and the underline character (_). Note: Tagnames are case-insensitive (“ABC” and “abc” are the same tagname).
 - A tagnames can contain no more the 40 characters.
 - A tagnames cannot start with the underline character or contain consecutive underline characters.
 - A tagname cannot start with a number.
 - A tagname must be unique.
 - Different tagnames cannot reference the same Host controller data.
- **Type:** The **TYPE** specifies the Host Controller data element type to be accessed. For example, a Type of V specifies V memory. The supported types are described in the following section.
- **Address:** The **ADDRESS** specifies the memory address of the data element in the Host Controller. For items with a count greater than 1, this represents the address of the first data element.
- **Count:** The **COUNT** specifies the number of consecutive data addresses to be included. A count of 1 specifies that a single data address will be accessed. For Types CP and XYP, the maximum count is 16. For all other Types the maximum count is 256.
- **High Priority:** Checking this box will apply the High Priority cache refresh interval to the item. If unchecked, the item will be updated using the Normal cache refresh interval. See section 0
-
- *Entering ECC1 Settings* for information regarding the cache refresh interval.

- **Description:** This field allows you to enter an optional description, which can be used to further identify the data item. The description will be displayed when the Tagname is mapped to the protocol data.
- **CC|MS|MC|NP|NS:** After the item is mapped to a protocol, a check mark will be displayed in the column corresponding to the protocol. CC= Camp Client, MS = Modbus Server, MC = Modbus Client, NP = Network Data Exchange Publisher, NS = Network Data Exchange Subscriber.
- **Transfer Set ID:** A transfer set is a group of data items that will be updated in the same scan. All members of a particular transfer set are assigned the same transfer set ID. Transfer sets are created when you specify data consistency (See APPENDIX D: DATA CONSISTENCY). This information is displayed to assist in correcting configuration errors related to data consistency.

Data Types

This section lists the Data types that may be selected and describes how they are used.

Discrete Data Types

Data Type C

Type C designates Control Relays, which are typically used by an RLL program to store intermediate states of contacts and coils. Control Relays can also be used by external devices, such as HMI or SCADA workstations to signal the RLL program to perform a certain action, such as turn on a motor or start a process. These data items are usually mapped to Boolean/discrete data items in a protocol. Control relays are not cleared by the 2500 Series® controller when transitioning from Run to Program mode but are cleared during a complete program load.

Type XY

Type XY designates discrete I/O image register items. Discrete image register addresses are usually tied to external I/O points that are either in an off or on state, such relays, indicators, limit switches. The controller designates a particular address as an X or a Y, depending on how it is used in the control program or the type of I/O it is associated with. An address preceded by an X is used for input while an address preceded by a Y is used for output. For example, X10 designates discrete image register address 10 used as an input and Y10 designates discrete image register address 10 used as an output. *It is important to understand that X10 and Y10 refer to the same data point.* While the PLC I/O system enforces that X items are tied to discrete process inputs and Y items are tied to discrete process outputs, external devices can both read and write to a particular I/O address without regard for its use in a PLC program or I/O system.

An important feature of the discrete image register is that all values are cleared when the controller transitions from Run to Program mode. This is done so that the discrete outputs that enable the process are turned off. If you are using the 2500P-ECC1 to control discrete I/O, you should map this cache item type to the protocol outputs you want to clear when the controller is placed in Program mode. For example, when using the Open Modbus Client protocol you should map this cache item type to Modbus coils. If your Modbus Slave device places control bits in a Modbus Holding Register, see Type XYP below.

16 bit Word Data Types

Type V

Type V designates V memory, a collection of 16 bit registers used to store internal data. V memory is used by the PLC program for intermediate data storage and by external sources, such as an HMI terminal, to exchange analog to with the PLC. Because V memory is not strongly typed, a V memory address can be interpreted as a

signed integer, unsigned integer, BCD value, ASCII characters, or a field of bits. This type is commonly used to as an efficient means to transfer blocks of data between controllers and other devices.

Type WXWY (Word I/O Registers)

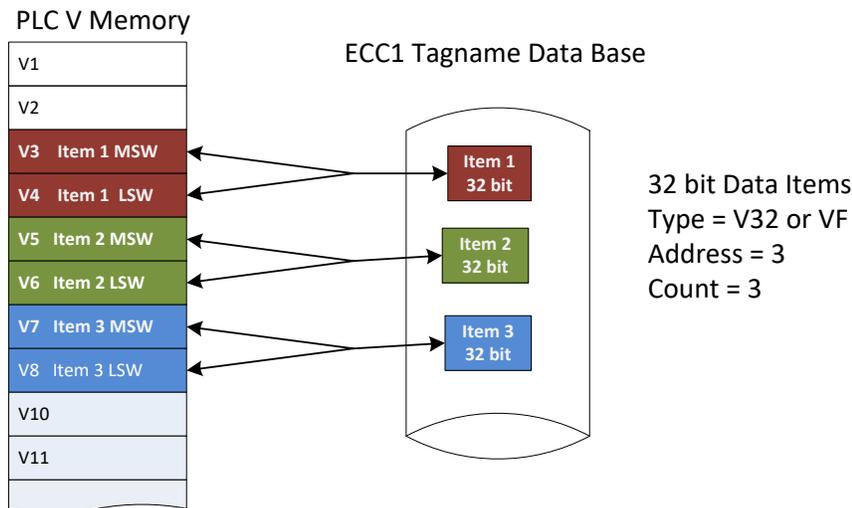
Type WXWY designates a Word I/O register item. Word I/O is typically used as to control and monitor analog process analog such as temperature or pressure. Consequently, it is usually mapped to protocol word representing analog I/O. However, it may also be used as a field of 16 bits for reading status information or writing to a set of discrete outputs. When doing so, you should be aware the Host PLC does not automatically clear word I/O when transitioning from Program to Run mode. Also, when a complete program is loaded, these words may be restored to their saved values.

32 bit Data Types

These data types are used to read and write 32 bit entities such as long integers and floating point numbers. In the CTI 2500 Series® controller, the data types are stored in two consecutive V memory locations. To maintain data integrity, The ECC1 ensures that both V memory locations are read from the PLC or written to the controller in the same scan.

When a 32 bit data type is selected, the address specified in the Address field represents the first V memory address to be read or written. The most significant word (MSW) of the first data item will be read from or written to this V memory address. The least significant word (LSW) will be read from or written the next higher V memory address. For example, when writing a single data item to the Host Controller where the Address field is 3, the MSW will be written to V3 and the LSW to V4. For a block of data items (see the following section), the additional data items will be read from or written to consecutive V memory locations. For example, when writing a block of 3 V32 data items starting at V3, the first item will be stored in V3 and V4, the second in V5 and V6, and the third in V7 and V8. See the illustration below.

The **COUNT** field represents the number 32 bit data items (Type V32 or VF). If a block of 3 data items is needed, the **COUNT** field must be set to 3, even though the Host Controller stores the data in 6 V memory locations.



Type V32

Type V32 designates a 32bit integer (signed or unsigned), which is stored in two consecutive V memory addresses. When reading a type V32 data item from the Host Controller, the data from two consecutive V

memory addresses are read and subsequently written to the ECC1 Tagname Database as a Doubleword. When writing Type V32 items to the Host Controller, a Doubleword is read from the ECC1 Tagname Database and subsequently written to two consecutive V memory addresses in the host controller.

Type VF

Type VF designates a 32 bit floating point number, which is stored in two consecutive V memory addresses. When reading a type VF data item from the Host Controller, the data from two consecutive V memory locations are read and subsequently written to the ECC1 Tagname Database as a real number. When writing a VF type to the Host Controller, a real number is read from the ECC1 Tagname Database and subsequently written to the two consecutive V memory locations in the host controller.

Discrete to Bit-of-Word Data Types

Type XYP (Discrete I/O Register Packed)

Type XYP allows up to 16 contiguous discrete I/O register values (X inputs or Y outputs) to be packed into a 16 bit word. It is primarily used when controlling devices that present discrete control interfaces as bits of a word rather than as individual discrete data items. **Associating control data bits with discrete I/O register outputs ensures that the control bits are cleared when the host controller transitions to Program mode or when a complete user program is loaded to the 2500 Series® controller.**

When this data item type is selected, the Address field designates the address of the first discrete image register point to be included in the word and the Count field designates the number of consecutive image register addresses to be included. The following table specifies range and default value field.

Field Name	Minimum Value	Maximum Value	Default Value
Address	1	16,384*	1
Count	1	16	1

* This represents the maximum address that can be entered. The maximum address that can be accessed depends on the Model of the 2500 Series® controller that you are using as the host controller.

The values of the designated Discrete Image Register addresses are packed into the word starting with the least significant bit (LSb). Unused bit positions will be filled with a value of 0. Following is an example when the user specifies an address of 50 and a count of 4.

MSb																LSb
0	0	0	0	0	0	0	0	0	0	0	0	0	X53	X52	X51	X50

When data is read from the Host Controller using this data type, the ECC1 ensures that all discrete I/O register addresses contained in the word are updated in the cache with data from the same scan, since it is likely that this data type will be used to control a set of discrete outputs. For example, when reversing a motor, the Run forward bit may be cleared at the same time the Run Reverse bit is set. When this data type is written to the Host controller, the associated the Discrete I/O register values will be written as a group.

Type CP (Control Relay Packed)

Type CP allows up to 16 contiguous control relay (C) values to be packed into a 16 bit word. It provides a more efficient method of transmitting control relay values than sending them individually.

When this data item type is selected, the Address field designates the address of the first control relay to be packed into the word and the **COUNT** field designates the number of consecutive control relay addresses to be included. The following table specifies range and default value field.

Field Name	Minimum Value	Maximum Value	Default Value
Address	1	32,767*	1
Count	1	16	1

* This represents the maximum address that can be entered. The maximum address that can be accessed depends on the Model of the 2500 Series® controller that you are using as the host controller.

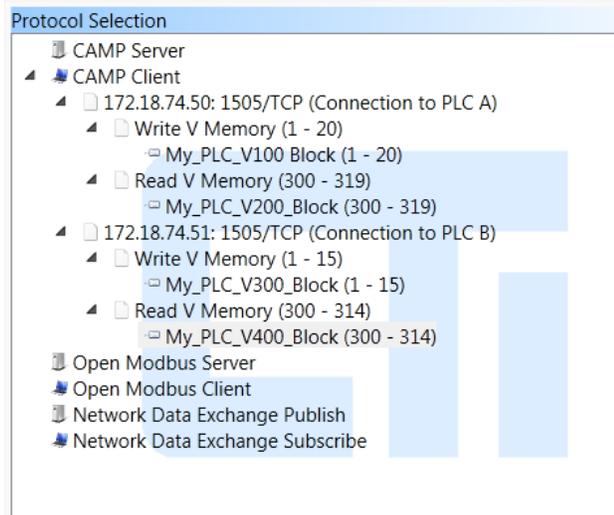
The values of the designated Control Relay addresses are located in the packed discrete word starting with the least significant bit (LSb). Unused bit positions will be filled with a value of 0. Following is an example when the user specifies an address of 9 and a count of 4.

Word Bits																
MSb															LSb	
0	0	0	0	0	0	0	0	0	0	0	0	0	C12	C11	C10	C9

5.3.4 Selecting and Configuring Protocols (Overview)

The protocol selection panel allows you to select a protocol and, if necessary, to create configuration objects required by protocol application. When configuration objects are required, they are presented in a tree structure, where dependent configuration objects are displayed as children of the higher level (parent) objects. The CAMP Client configuration will serve to illustrate the configuration process. Since the intent of this section is to illustrate the process, portions of the CAMP client configuration are abbreviated. For detailed instructions see Section 5.3.6.

The screen capture to the right shows the completed example where there are two connection objects, each with two requests objects and associated tagnames.



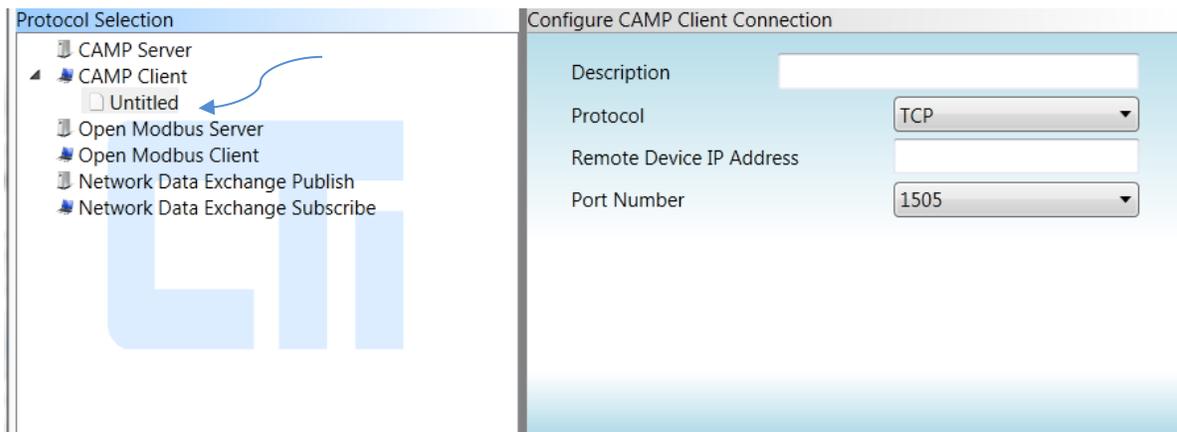
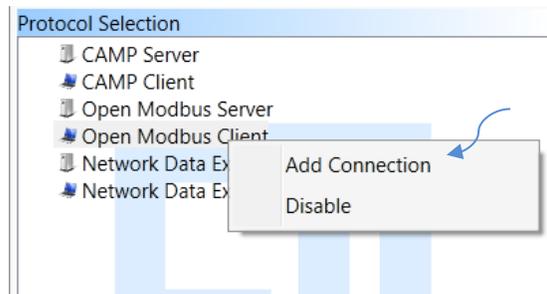
To create a dependent (child) object, you right click in the higher level (parent) item. When you right click on the parent, you will be presented with a list of options. The options will vary depending on the parent level and the state of the child objects.

Possible options are:

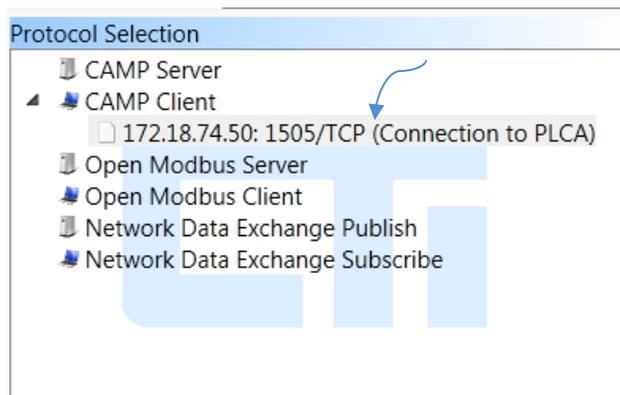
- **Add (a child object):** The child object may be a connection, request, tagname, or other applicable object, depending on the parent.
- **Disable:** This option allows you to disable the object (and all children). When an object is disabled, it will not be included in the resulting ECC1 execution file. This option can be used to eliminate error messages in situations where the target device is not yet installed, such as installation, testing and commissioning activities.
- **Insert:** This option allows you to insert an object above the one you have selected.

- **Expand All Children:** This option shows all dependent objects of the selected object.
- **Collapse All Children:** This option allows you to hide all dependent objects of the selected object.

To begin configuring the CAMP Client, right click on the CAMP Client object and select the “Add Connection” option. After clicking on the “Add Connection” selection, the Protocol Selection and Configuration panels appear as shown below. Note that a child connection object, “Untitled”, has been added to the selection tree and the Configuration panel displays the related configuration items.

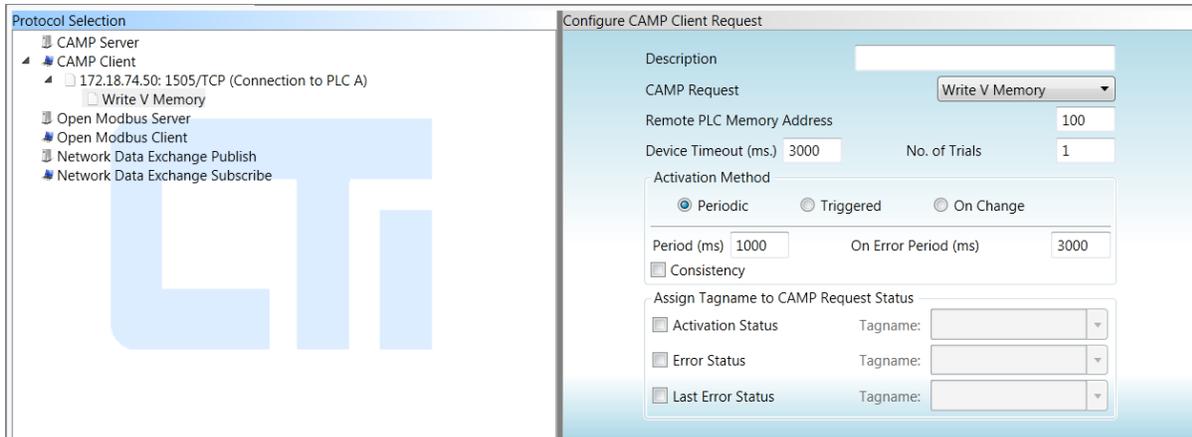
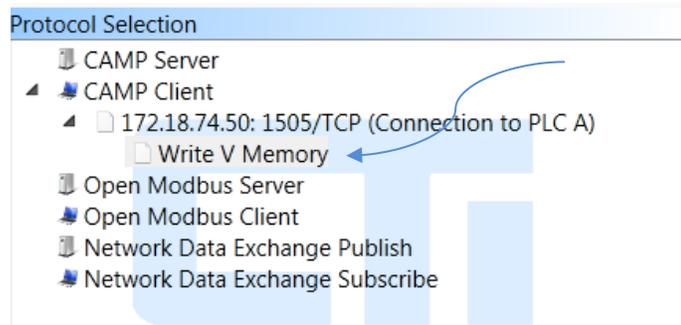


After the connection object configuration parameters have been entered, the Protocol Selection Pane will reflect the configuration data previously entered as show in the accompanying illustration.

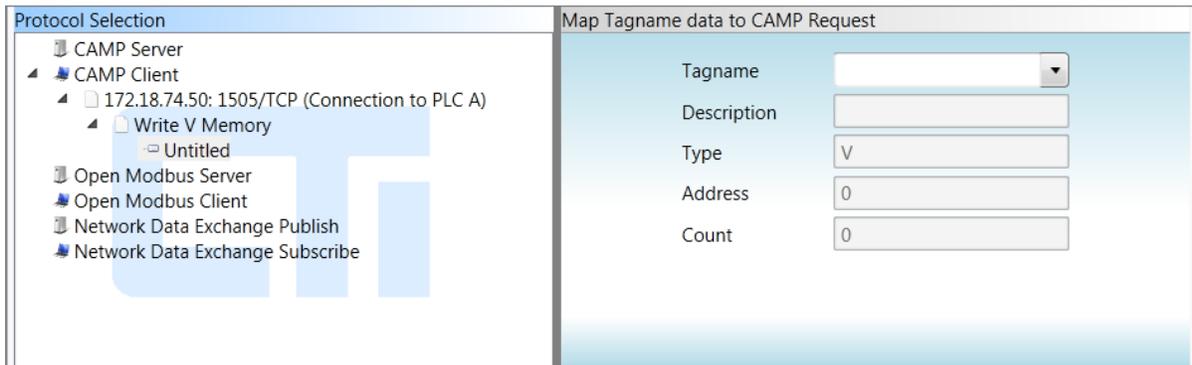


Right clicking on the connection object and selecting the “Add Request” option allows you to add a request object, which specifies a particular read or write operation.

When the request object is added, the Configuration Panel will display the configuration parameters for the request as illustrated below.

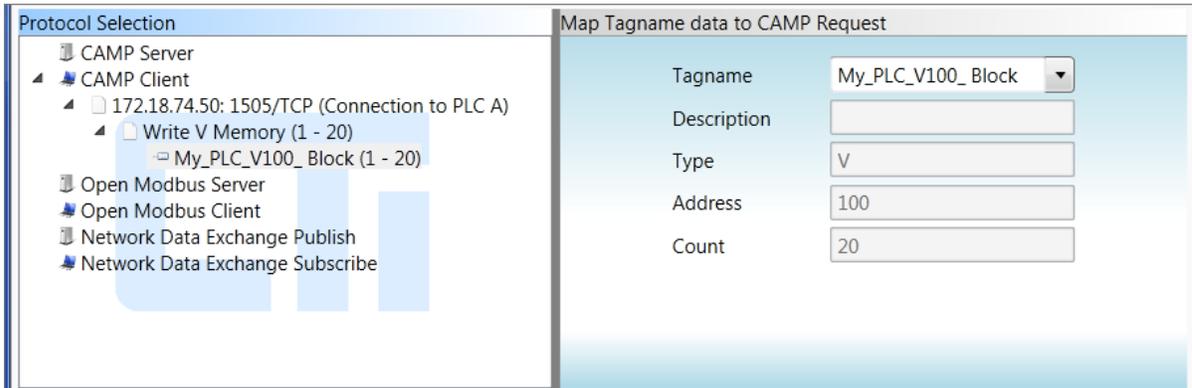


After entering parameters for the request in the Configuration panel, right click on the “Write V Memory” request object and select the “Add Tagname” option. This will cause the tagname mapping window to be displayed in the Configuration panel as shown below.



The Tagname selection box allows you map a Host Controller data to the protocol by selecting a tagname previously entered into the Tagname Database. Once the tagname is selected, the Description, Type, and Count for the Tagname Database item will be displayed.

The following illustration shows the Protocol Selection and Configuration panels after selecting a tagname.



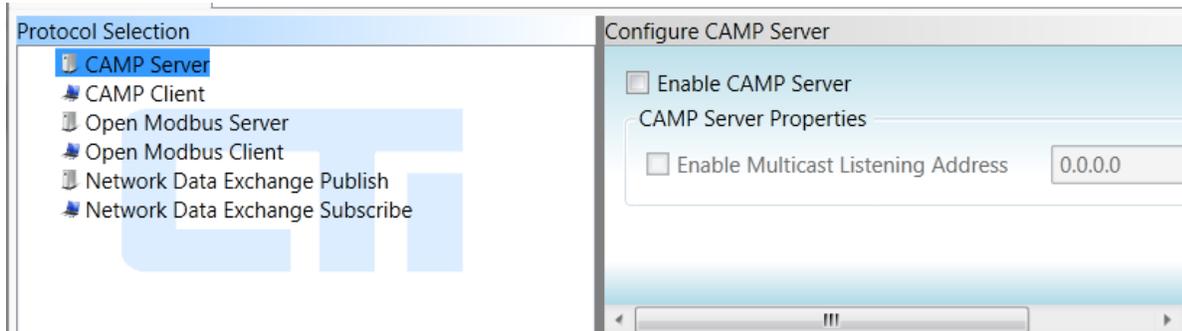
In the example above, one CAMP client TCP connection to IP address 172.18.74.50 has been created with one request to write data to V memory locations 1 – 20 where the data will originate from the a block of 20 V memory locations starting at V100 in the Host Controller.

The same procedure as above would be used to create and configure the additional configuration objects in the example.

You can incrementally build and test a configuration project by compiling the project and transferring the resulting configuration project files to the ECC1 module at various stages of the configuration. Before you can compile the project, the configuration must be free of errors. Although the validation test is performed during the compile process, you can check the validity at any time by selecting the “Verify Data” toolbar item.

5.3.5 Configuring the CAMP Server

To configure the CAMP Server, click on the CAMP server item in the Protocol Selection Panel. The configuration panel will then display the CAMP server configuration items. Click on the Enable CAMP Server check box to enable the CAMP server. In many cases, this is the only configuration required, since the CAMP server will use the module IP parameters.



If you plan to receive data from CAMP clients using multicast, you will need to click on the “Enable Multicast Listening Address” checkbox and then the IP address of the multicast transmission you wish to receive. See APPENDIX B: IP ADDRESS INFORMATION for information about assigning a multicast address

You do not need to enter anything into the Tagname Database for data items accessed by the CAMP server unless you want them to be updated using the High Priority cache refresh interval. Host controller data items are automatically added to the cache when they are first accessed by a client. Items remain in the cache until they are no longer being accessed. After a period of 60 seconds with no access, an item is removed from the cache. By default, data cache members accessed by CAMP server are updated using the Normal cache refresh interval (CRI).

If you want certain data items to be updated using the High Priority CRI, you must create one or more entries in the Tagname Database representing these data items and check the High Priority box for each entry. These items will be permanently cached. See Section 5.3.3 for information about entering Tagname Database items.

Client requests to write data to the Host Controller are immediately transferred to the Host Controller upon receipt. As soon as the write request completes successfully, the corresponding cache members are updated. Consequently, there is no need to assign the related data item(s) to the High Priority CRI in order to ensure adequate performance. However, if the client application is monitoring a different data item to confirm the success of a write request, you may wish to assign the High Priority CRI to the monitored data item. For example, if a SCADA application is using a pushbutton tied to C100 to turn a motor on and an indicator tied to C200 to indicate the motor is on, you might create a Tagname Database item representing C200 and check the High Priority box for the item.

5.3.6 Configuring the CAMP Client

To configure the CAMP Client, you must:

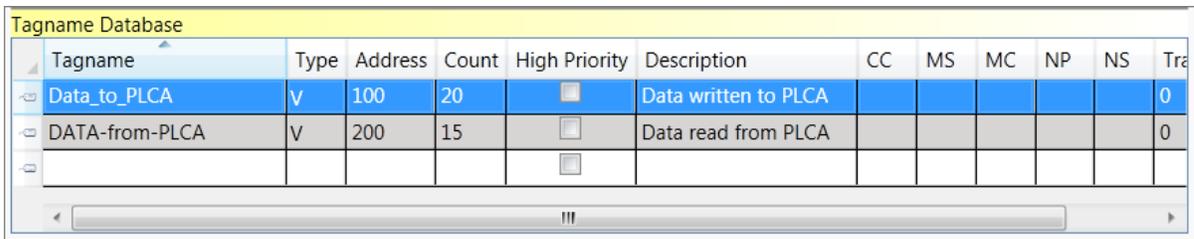
- Enter items into the Tagname Database,
- Configure at least one CAMP client connection,
- Configure at least one CAMP request per connection,
- Map at least one Tagname to each CAMP request.

These steps are described in the following sections.

Creating Tagname Database Items

You will need to specify the V memory locations that will be used as the source of the data used for each write requests or as the destination for the data obtained from read requests. This is accomplished by adding entries to the Tagname Database. See *Entering Tagname Database Data Items* for more information about the Tagname Database.

In the example below, a block of 20 host controller V memory locations (V100-V119) has been assigned a tagname of “Data_to_PLCA”. The data in these V memory locations will be written to the remote PLC. In addition, a block of 15 host controller V memory locations (V200-V219) is assigned a tagname of “Data_from_PLCA”. The data in these V memory locations will be read from the remote PLC. These tagnames will be associated with the CAMP requests later in the configuration process.



Tagname	Type	Address	Count	High Priority	Description	CC	MS	MC	NP	NS	Tr
Data_to_PLCA	V	100	20	<input type="checkbox"/>	Data written to PLCA						0
DATA-from-PLCA	V	200	15	<input type="checkbox"/>	Data read from PLCA						0
				<input type="checkbox"/>							

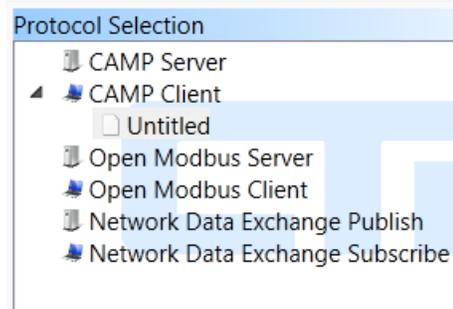
Configuring a CAMP Client Connection

You must add a client connection object for each remote device you want to communicate with. For a given connection, requests are serialized; one request must complete (either successfully or with error) before the next request will be sent. Each connection sends requests independently of other connections; a connection does not need to wait on another connection to complete a request. Connections may send requests concurrently.

In certain applications where the performance using a single connection is not adequate or when you need to configure more than 8 requests per remote device, making a second connection to a remote device can resolve the problem. If you are considering this option, please refer to the note below.

NOTE:
*When making multiple connections to the same device use the TCP protocol. Many TCP/IP servers, including the CTI 2572 and 2572-A, do not handle multiple request streams from the same IP address when UDP is used.
This technique consumes additional connection resources on the remote device and the ECC1 module. The device you are using may not support multiple connections. You should test thoroughly before implementing this technique.*

You may add up to 16 CAMP client connection objects. To add a connection object, right-click on the CAMP Client in the Protocol Selection panel, then click on the “Add Connection” item. An “untitled” connection object will be displayed.



Clicking on the “untitled” connection object will display the Configure CAMP Client Connection parameters in the Configuration panel, as shown below.

Description: You may enter up to 40 characters describing the connection, for example, the name of the PLC you are communicating with. This information will appear in the connection object in the Protocol panel.

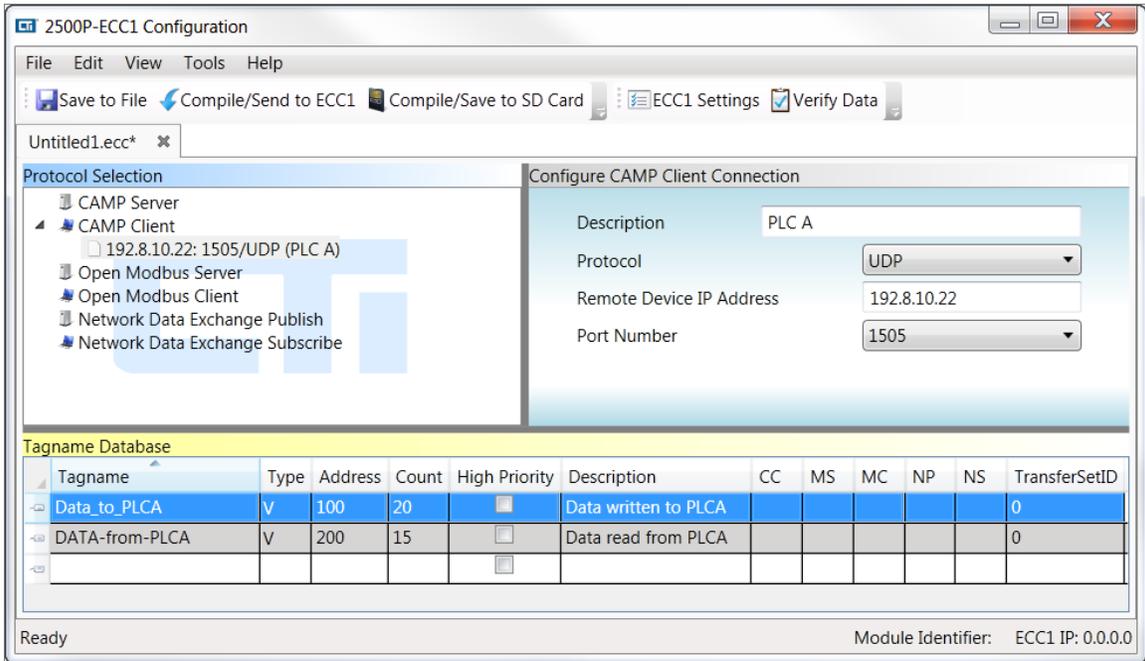
The image shows the 'Configure CAMP Client Connection' configuration panel. It has a light blue header and a white body. There are four fields: 'Description' (a text input field), 'Protocol' (a dropdown menu with 'TCP' selected), 'Remote Device IP Address' (a text input field), and 'Port Number' (a dropdown menu with '1505' selected).

Protocol: Select the IP protocol to be used. You can choose TCP, UDP, or UDP Multicast. If you choose the multicast option, you will be limited to write requests when configuring the CAMP request.

Remote Device IP Address: Enter the IP address of the remote device or, if you selected UDP Multicast, enter the multicast address. See APPENDIX B: IP ADDRESS INFORMATION for more info.

Port Number: Select the IP port number to be used (1505 or 4450). Port 4450 is a CTI registered port for the CAMP protocol. Port 1505 was originally established as the default port for the CTI 2572 and 2572-A Ethernet at a time when the port number was unused. If you are communicating with these products in an existing installation, you will likely use port 1505. Newer CTI products, such as the 2500 Series® controller and the 2500P-ECC1, listen on both port 1505 and 4450. Configuring 2572 and 2572-A modules to use port 4450 may be acceptable in a new installation. When feasible, use the registered port 4450.

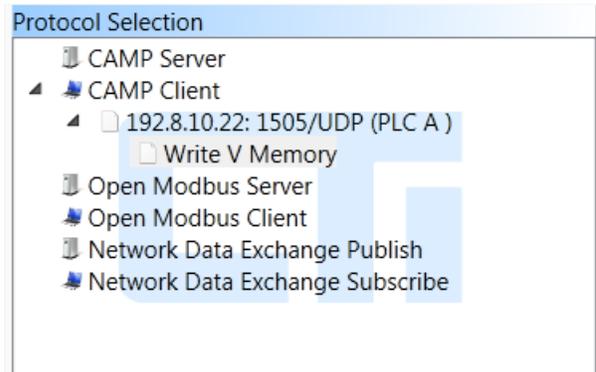
Following is an example illustrating the configuration program display after example data has been entered.



Configuring a CAMP Client Request

For each CAMP Client connection, you must add at least one CAMP Client request object. To accomplish this, right click on the client connection object and click on the “Add Request” item.

After performing these actions, the Protocol Selection panel should look like the illustration to the right of this paragraph. You may add up to 8 requests per connection.



The configuration parameters related to the CAMP Client request will be displayed in the configuration panel shown below.

Description: You may enter up to 40 characters describing this request.

CAMP Request: Select the CAMP request (**WRITE V MEMORY** or **READ V MEMORY**). If you selected the Multicast protocol, you should select the **WRITE V MEMORY** request.

Remote PLC Memory Address: Enter the initial V memory address in the remote PLC

Device Timeout: Except for multicast transmission, the CAMP Client protocol expects the remote PLC to reply to a request. The **DEVICE TIMEOUT** value is the number of milliseconds the ECC1 will wait for a reply from the device. If the remote PLC replies within this time interval, processing continues normally. If the remote PLC fails to respond within this interval, the current attempt to communicate with the device will be terminated. If you are using TCP, the connection will be closed (and re-opened on the next attempt). Additional attempts to communicate with this device during the same activation period are determined by the “Number of Trials” parameter described in the next section.

The screenshot shows the 'Configure CAMP Client Request' dialog box. It has a light blue background and a white border. The title bar says 'Configure CAMP Client Request'. The fields are as follows: Description (text input), CAMP Request (dropdown menu showing 'Write V Memory'), Remote PLC Memory Address (text input showing '1'), Device Timeout (ms) (text input showing '3000'), No. of Trials (text input showing '1'). Under 'Activation Method', there are three radio buttons: 'Periodic' (selected), 'Triggered', and 'On Change'. Below these are two text inputs: 'Period (ms)' showing '1000' and 'On Error Period (ms)' showing '3000'. There is a checkbox for 'Consistency'. At the bottom, there is a section 'Assign Tagname to CAMP Request Status' with three rows: 'Activation Status' with a 'Tagname:' dropdown, 'Error Status' with a 'Tagname:' dropdown, and 'Last Error Status' with a 'Tagname:' dropdown.

Setting the **DEVICE TIMEOUT** value too small will result in timing out before the device has had a chance to respond, which will have a major effect on device communications. Setting the **DEVICE TIMEOUT** value too large will result in some unnecessary delays when the device is offline or having problems communicating.

The timeout value should be set to a value slightly larger than the maximum response time of the device. Since the maximum response time is often difficult to determine, you may start with a value that is 2 - 3 times the normal response time of the device. If it is impractical to determine the device response time, start with the default value of 3000ms. This value should allow sufficient time for devices on the local area network while introducing a moderate delay when the device is offline or faulty. If you are communicating over the Internet or if the device is very slow to respond, you may need to increase the timeout value.

No. of Trials: This is the maximum number of times that the CAMP Client protocol will attempt to send a request to the device during the current activation of the request. Setting the number of trials greater than 1 allows you to immediately retry sending the request during the activation period. If the request is a **WRITE** request, additional requests will attempt to send the **same** data each trial.

Additional trials should be used only if the application demands it. Note that each additional trial in which the device fails to respond delays sending other requests on this connection. The amount of delay is approximately equal to the value of the Device Timeout parameter.

NOTE:

The Device Timeout and Number of Trials parameters are not used when UDP Multicast is selected.

Activation Method: The activation method specifies the event that will initiate the request. The methods include **PERIODIC**, **TRIGGERED**, and **ON CHANGE**.

Periodic: This method initiates the request on a specified time period. When the **PERIODIC** method is selected, you can specify the **PERIOD** and the **ON-ERROR PERIOD** and select **CONSISTENCY** as shown below:

Period: This parameter specifies the normal time interval used to initiate requests. All requests for a particular connection are placed in an execution queue, which is serviced sequentially. Each request is allowed to complete (either successfully or with error) before the next request will be serviced. Depending on the number of pending requests and the time required to service previous requests, requests may not be serviced as often as the PERIOD value specifies.

On Error Period: This parameter specifies the time interval used after the device fails to reply to a request within the timeout period. Specifying an **ON ERROR PERIOD** value that is greater than the **PERIOD** value reduces the overhead incurred when polling a device that is off-line or not present. *The On Error Period is not used when UDP Multicast is selected.*

Consistency: The **CONSISTENCY** checkbox is displayed only for write requests. Checking the **CONSISTENCY** box indicates that you want all the data cache members associated with this request to be updated as a group in the same host controller scan. Since consistency is usually not required for applications using periodic activation, the default is unchecked (no consistency). See APPENDIX D: DATA CONSISTENCY for more information regarding consistency.

Triggered: This method allows user logic in the Host Controller to initiate the request. When the triggered method is selected, the following parameters are displayed.

One Shot: This trigger type initiates a request only when the trigger value transitions from **OFF** to **ON**. The one-shot trigger allows you to initiate a transaction when using user logic in the Host Controller. If you are using the triggered method, this is the trigger you will most likely use.

Enable: This trigger type initiates requests as often as possible while the trigger value is **ON** (non-zero). While this method is applicable in certain situations, in most cases it is easier to achieve a similar result using the periodic method.

Tagname: This field allows you to select the tagname representing the host controller data address that will be used to set the trigger value.

NOTE:
*When a **one-shot** trigger is used, the Tagname Database item associated with this tagname will automatically be assigned a Cache Refresh Interval (CRI) of 100ms.*

Consistency: The Consistency checkbox is displayed only for write requests. Checking the **CONSISTENCY** box indicates that you want all the data cache members associated with request to be updated in the same host controller scan. When used with a one shot trigger it has special meaning: *when consistency is selected, the data mapped to a write request will be obtained from the Host Controller immediately after the trigger transitions from low to high, rather than being read from the cache.* Since this behavior is desirable for most one shot trigger applications, consistency is selected by default. If you are using an enable trigger, you will likely want to disable consistency. See APPENDIX D: DATA CONSISTENCY for additional information regarding consistency.

On Change: The **ON CHANGE** method initiates a request only when the Host Controller data mapped to the request changes in value. *This method can be used only for write requests.* Using the **ON CHANGE** method provides a more timely initiation while reducing unnecessary requests.



Activation Method

Periodic Triggered On Change

Consistency

Consistency: Checking the **CONSISTENCY** checkbox indicates that you want all the data cache members associated with this request to be updated as a group in the same host controller scan. Since data consistency is not required for most applications using periodic activation, the default is unchecked (no consistency). See APPENDIX D: DATA CONSISTENCY for guidelines for using data consistency.

NOTE:

When the On Change method is used, all the current value of all data items mapped to the request is written when any data item mapped to the request changes in value.

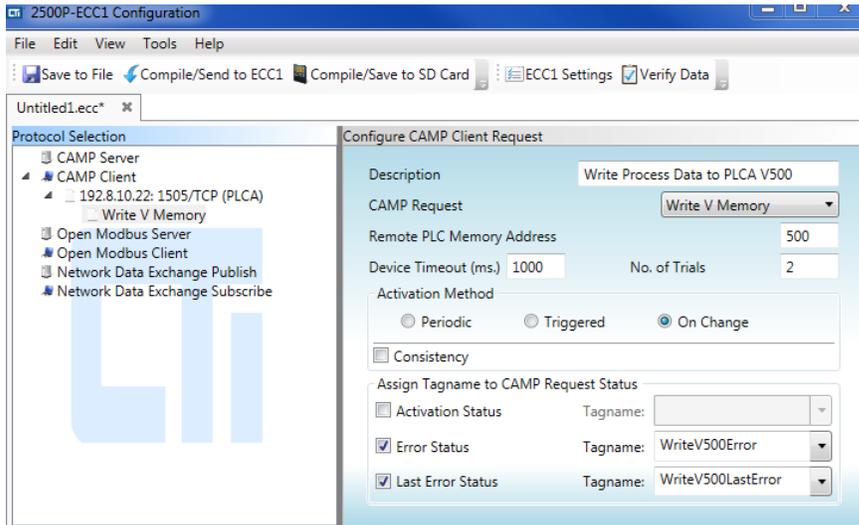
Assign Tag name to CAMP Request Status: These fields allow you to monitor the status of a request by mapping the tagname of an item in the Tagname Database to the execution status of the request. Use of these fields highly recommended. **NOTE:** *The related tagname database item must have a count of 1.*

Activation Status: The **ACTIVATION STATUS** changes to true when the request begins execution. It remains true until the request is completed successfully or terminates with an error, when it is set to false. **ACTIVATION STATUS** is primarily used with the triggered activation method.

Error Status: When a request terminates because of an error, an error code is written to the **ERROR STATUS**. When the request completes successfully, the **ERROR STATUS** is set to 0, clearing any previous error code. *You should assign the Error Status to a tagname when PLC logic is used to process the error condition.*

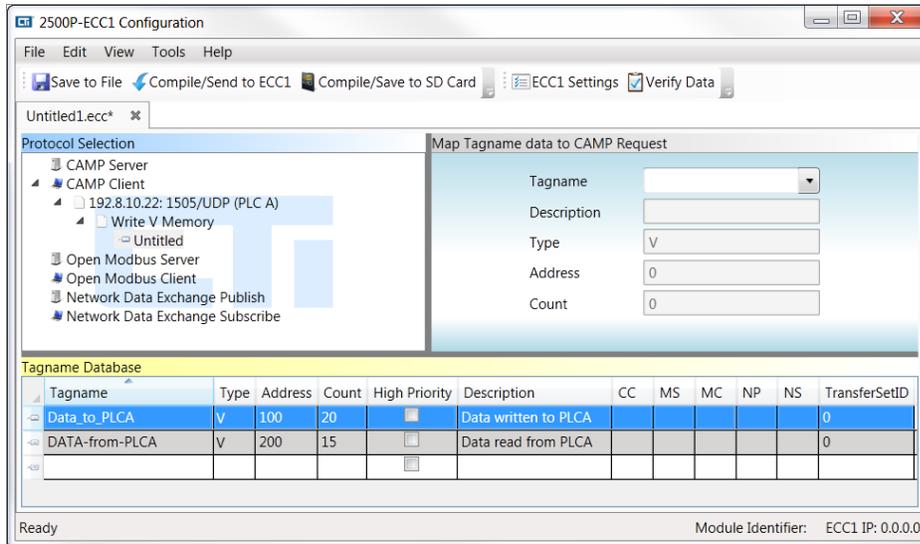
Last Error Status: The **LAST ERROR STATUS** contains the error code for the last error that occurred when this request was executed, which may be on a previous activation cycle. Unlike the **ERROR STATUS**, the **LAST ERROR STATUS** is not set to 0 when a request completes successfully. ***If you are not using PLC logic to handle an error condition, you should always assign a tagname to the Last Error Status. The last error status is extremely valuable in diagnosing networking problems that may occur.***

Following is an example illustrating the configuration program display after sample data has been entered. In this example, the ECC1 will write data to PLCA starting at V500. The CAMP **WRITE V MEMORY** request will be activated when the data to be written changes value. After activation, the ECC1 module will wait up to 1 second (1000ms), for a reply. If a reply is not received within this time, the ECC1 module will immediately resend the request. If a no reply to the second attempt is received within 1 second, the ECC1 module will declare a timeout proceed to service the next request, if one is pending. The Error Status and Last Error Status are mapped to the Tagname Database.

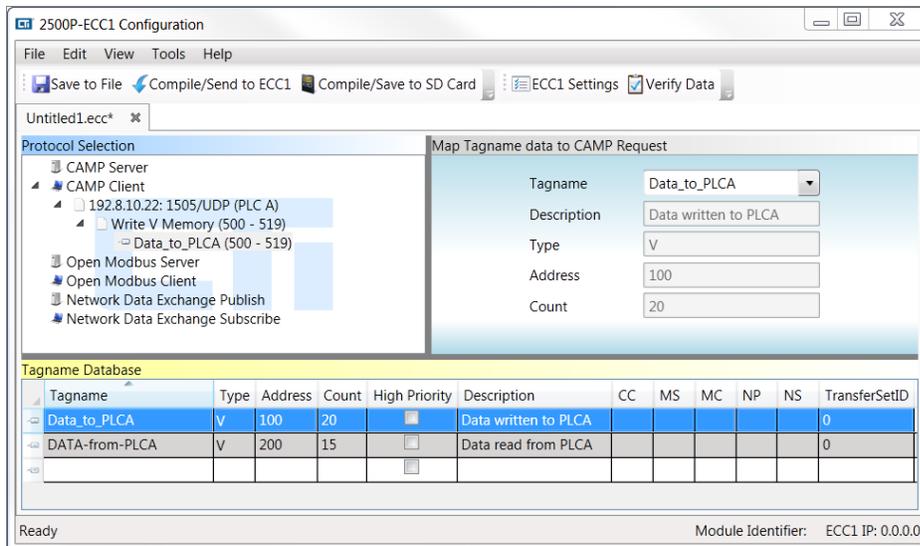


Mapping a Tagname to a CAMP Client Request

After configuring the request, the final step is to map Host Controller data to the request by selecting one or more tagnames from the Tagname Database. To accomplish this task, right click on the request object and select the “Map Tagname” item. The resulting configuration panel should appear as illustrated below.

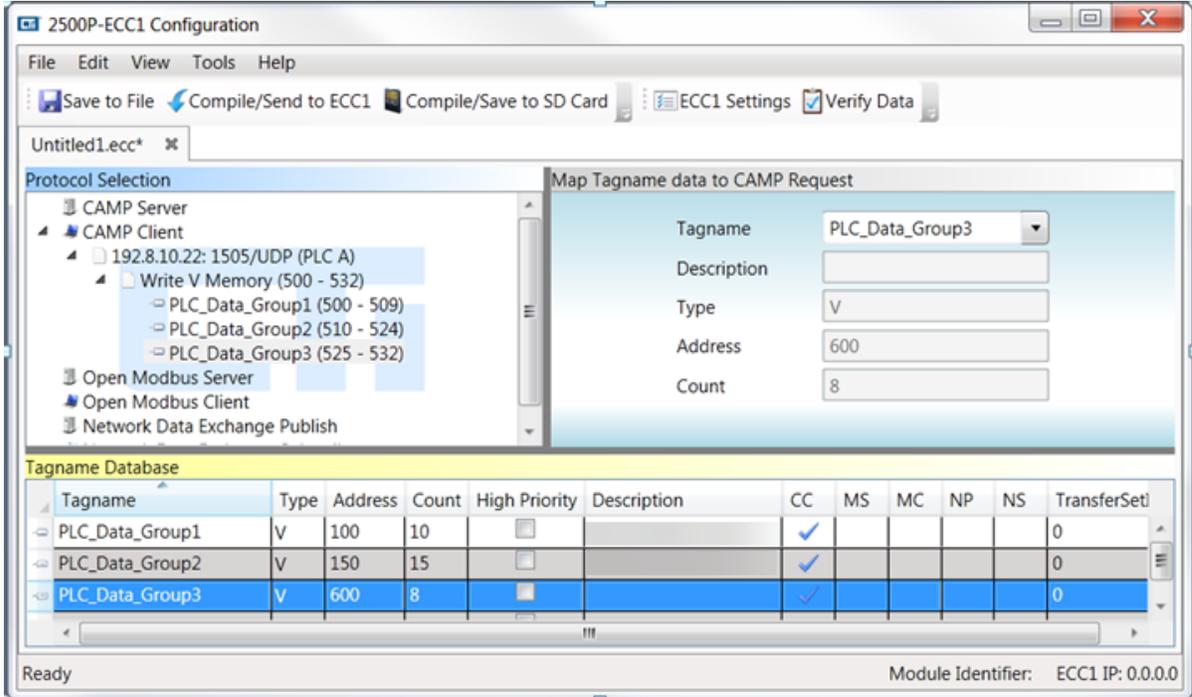


To map Host Controller data to the request, click on the Tagname field and select a tagname from a list of tagnames contained in the Tagname Database. In this example, after selecting the “Data_to_PLCA” tagname, the configuration window should look like the following illustration.



When the request is executed, the data contained in the Host Controller memory locations V100-V119, represented by the tagname “Data_to PLCA”, will be written to V500-V519 in the PLCA.

Although it is usually easier (and more efficient) to map a tagname representing a contiguous block of host controller memory to a CAMP request, you can map multiple tagnames, each representing a separate block of host controller memory to a CAMP request as shown in the following illustration. When mapping multiple items, the Host Controller data is mapped to the CAMP protocol data in the order in which the tagnames are added to the request object.



In the example above, the host controller data represented by the **PLC_DATA_GROUP1** tagname will be written to V500-509 in PLCA, the host controller data represented by the **PLC_DATA_GROUP2** tagname will be written to V510-524 in PLCA, and the data represented by **PLC_DATA_GROUP3** tagname will be written to V525-532 in PLCA.

5.3.7 Configuring the Open Modbus Server

Using the Open Modbus Server protocol, you can provide a means for a wide range of industrial control devices to access specific blocks of memory in 2500 Series® controllers.

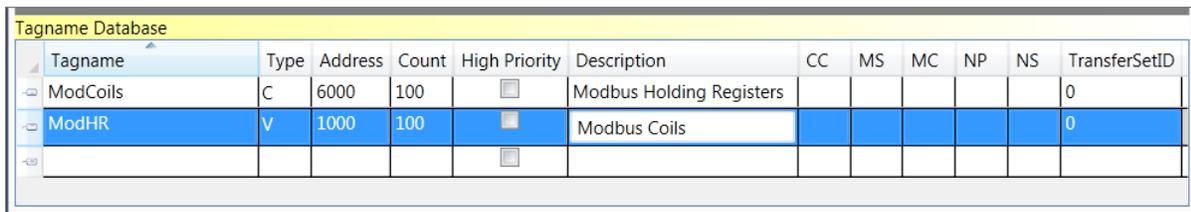
To configure the Open Modbus Server protocol, you must:

- Specify the Host Controller memory that can be accessed using the open Modbus protocol.
- Configure the Open Modbus Server Properties
- Configure one or more Modbus data blocks, representing Modbus data types and corresponding addresses will be accessible,
- Map one or more Tagnames to each configured data block.

Creating Tagname Database Items

You will need to specify blocks of Host controller memory addresses that will be used by the Open Modbus Server protocol. This is accomplished by adding entries to the Tagname Database. You must reserve at least one contiguous block of V or C memory for each Modbus data type that you will support. Although you can use other data item types, CTI recommends that you allocate V memory for Holding Registers and Input Registers and C memory for Discrete Inputs and Coils.

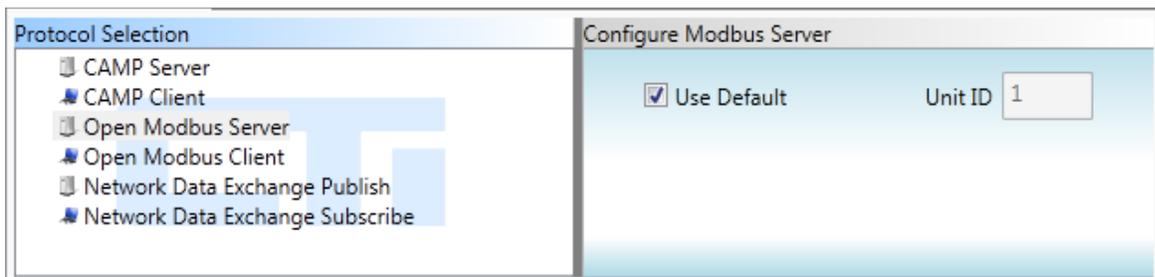
In the example below, a range of 100 Host Controller V memory locations (V1000 – V1099) has been assigned a tagname of “ModHR”. As the description denotes, this memory will be used for Modbus Holding registers. In addition, a range of Host Controller C memory locations (C6000-6099) has been assigned a tagname of “ModCoils” This will be used for Modbus Coils.



Tagname	Type	Address	Count	High Priority	Description	CC	MS	MC	NP	NS	TransferSetID
ModCoils	C	6000	100	<input type="checkbox"/>	Modbus Holding Registers						0
ModHR	V	1000	100	<input checked="" type="checkbox"/>	Modbus Coils						0
				<input type="checkbox"/>							

Configuring the Modbus Server Properties

To configure the Open Modbus Server Unit Identifier, click on the Open Modbus Server in the Protocol selection panel. This action displays the Configure Modbus Server parameters as shown below.

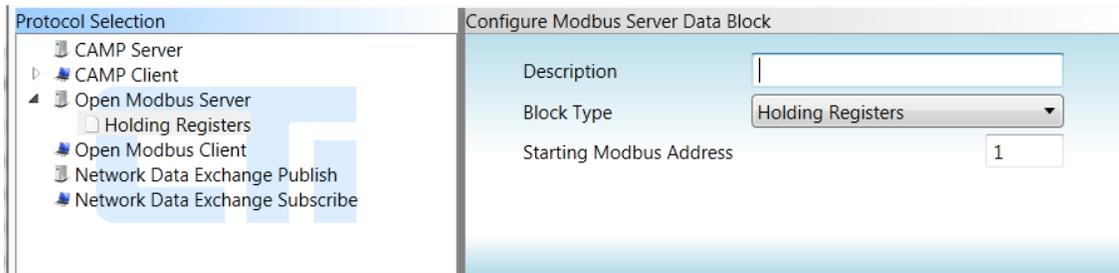


There is only one configuration item at this level, the Unit ID. The Unit ID is equivalent to the Modbus slave address. Since the IP address uniquely identifies Modbus devices directly to an Ethernet network, the Unit ID is redundant for Modbus servers. All Modbus servers can use the same Unit ID, unless the Modbus client requires different Unit ID values. This parameter is set by default to 1. The default should be used unless a particular Modbus client requires a different value.

Configuring a Modbus Server Data Block

A Modbus data block represents a block of contiguous **Modbus** addresses of a particular Modbus data type, such as Holding Registers. To use the Open Modbus Server, you must add at least one Modbus data block. You may add up to 64 data blocks, if necessary. Multiple data blocks of the same data type can be created, as long as the address ranges cannot overlap.

To add the data block right click on the Open Modbus Server object in the Protocol selection Configuration panel and click on the **ADD DATA BLOCK** option. The following illustration shows the Modbus Server Data Block configuration parameters.

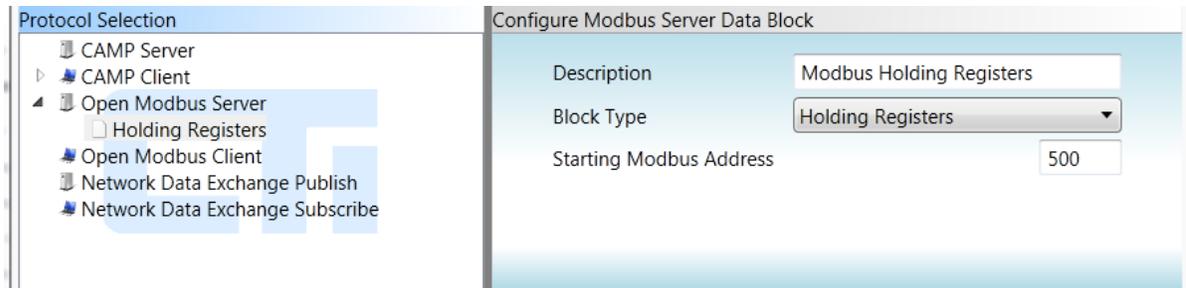


Description: You may enter up to 40 characters describing the data block.

Block Type: Select the Modbus Data type of the block. You may choose Holding Registers, Input Registers, Coils or Discrete Inputs.

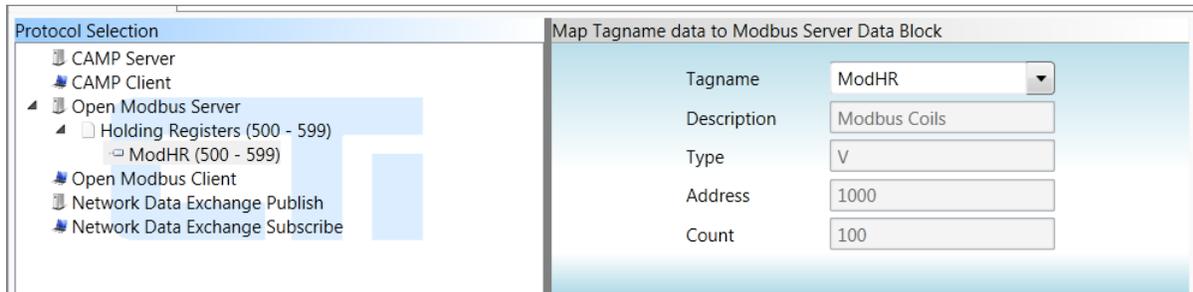
Starting Modbus Address: Enter the address of the first address in the block. The maximum Modbus address is 65,535.

In the example below a block of Modbus Holding Registers starting at Modbus Address 500 has been specified.



Mapping a Tagname to a Modbus Server Data Block

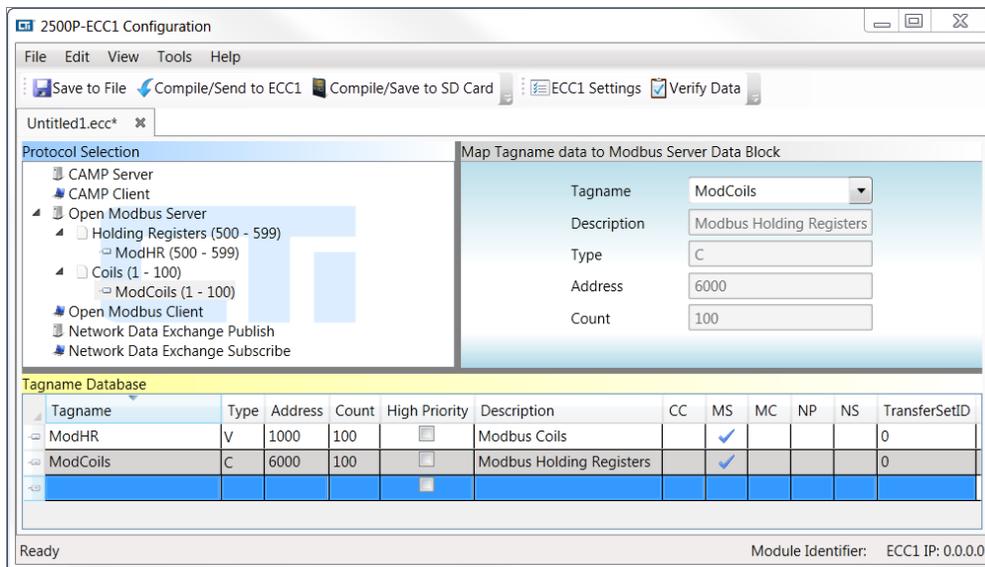
After configuring the data block, you must map host controller data to the data block by selecting a tagname contained in the Tagname Database. To accomplish this task, right click on the “Holding Registers” data block object and select the “Add Tagname” option. Then click on the Tagname field and select the ModHR tagname. After completing these tasks, the configuration program window should look like the illustration below.



Using this configuration, Modbus requests for Holding Registers 500 – 599 will access PLC V memory 1000-1099.

Creating Additional Server Modbus Data Blocks

Additional blocks can be configured in a similar manner. The following illustration shows the configuration window after a block of 100 Modbus coils starting with Modbus address 1 has been created and mapped to the host controller C memory starting at C6000. In this configuration, a Modbus request to read Coil address 1 will access C6000 in Host Controller.



5.3.8 Configuring the Open Modbus Client

To configure the Open Modbus Client protocol, you must:

- Specify the Host Controller memory that will be used,
- Configure at least one Open Modbus client connection,
- Configure at least one Open Modbus request per connection,
- Map at least one Tagname to each Open Modbus request.

Creating Tagname Database Items

You will need to specify the memory locations that will be used as the source of the data used for each write request or as the destination for the data obtained from read requests. This is accomplished by adding one or more entries to the Tagname Database.

In the example below, a block of 20 host controller V memory locations (V500-V519) has been assigned a tagname of “Data_from_ModbusDeviceA_HR”. This data will be read from the holding registers in the Modbus device. In addition, a block of 10 Host Controller C memory locations (C600-609) been assigned a tagname of “Data_to_ModbusDeviceA_Coils”. The data on these C memory addresses will be written to coils in the Modbus device.

Tagname Database											
Tagname	Type	Address	Count	High Priority	Description	CC	MS	MC	NP	NS	TransferSetID
Data_from_ModbusDeviceA_HR	V	500	20	<input type="checkbox"/>							0
Data_to_Modbus_DeviceA_Coils	C	600	10	<input type="checkbox"/>							0
				<input type="checkbox"/>							

Configuring a Modbus Client Connection

For each device you want to communicate with, you must add an Open Modbus client connection object. You may add up to 64 Open Modbus client connection objects. For a given connection, requests are serialized; one request must complete (either successfully or with error) before the next request will be sent. Each connection sends requests independently of other connections; a connection does not need to wait on another connection to complete a request. Connections may send requests concurrently.

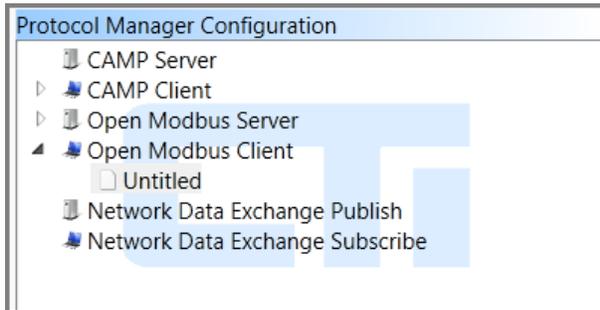
In certain applications where the performance using a single connection is not adequate or when you need to configure more than 8 requests per remote device, making a second connection to a remote device can resolve the problem. If you are considering this option, please refer to the note below.

NOTE:

When making multiple connections to the same Modbus device use the Modbus TCP protocol. Most Modbus Devices that support The Modbus UDP protocol do not handle multiple request streams from the same IP address.

This technique consumes additional connection resources on the remote device and the ECC1 module. The device you are using may not support multiple connections. You should test thoroughly before implementing this technique.

To add a connection object, right-click on the Open Modbus Client in the Protocol selection panel, then click on the “Add Connection” item. After performing these actions, Protocol selection panel should look like the one below.



Clicking on the untitled connection object will display the Configure Modbus Client Connection panel.

Description: This field allows you to describe the connection, for example, the name of the Modbus device you are connecting to. The description can be up to 40 characters in length.

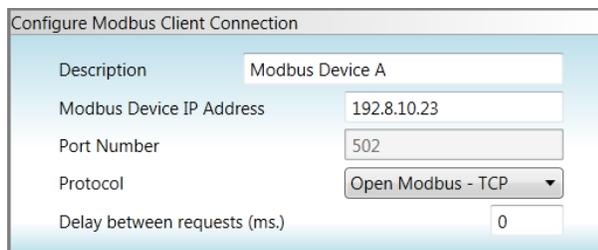
Modbus Device IP Address: Enter the IP address of the Modbus server device with which you want to communicate.

Port Number: IP port number 502 is the registered port for the Open Modbus Protocol. This port number is fixed and cannot be changed.

Protocol: Select the IP protocol to be used. You can choose either **OPEN MODBUS-TCP** or **OPEN MODBUS-UDP**. **OPEN MODBUS TCP**, which sends Modbus requests using TCP, complies with the specifications maintained by the Modbus Organization and is supported by most devices. **OPEN MODBUS-UDP** is a variant that sends Modbus request using UDP. Some Modbus devices support this protocol because it introduces less overhead.

Delay between Requests: This specifies the minimum time interval between Modbus requests initiated on the same connection. The primary use of this parameter is to allow devices on an RS-485 serial network time to prepare for receiving a new request after replying to the previous request. Modbus devices attached directly to Ethernet very rarely require this delay. Serial Modbus slaves attached using a gateway usually don't require a delay, since the gateway handles the timing on the serial network. ***Unless you have a reason to do otherwise, set the delay to 0.***

Following is an example illustrating the configuration program display after sample data has been entered.

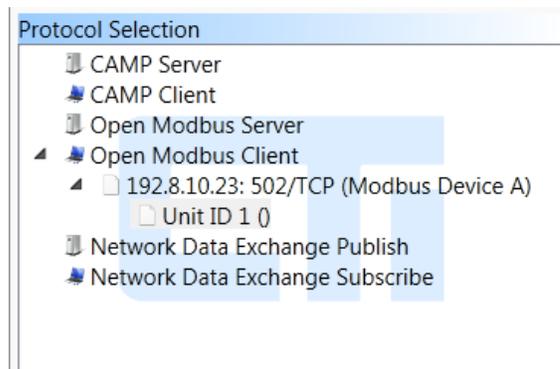


Configuring a Modbus Unit Identifier

The Modbus Unit Identifier (Unit ID) is equivalent to the Modbus Slave Address used by Modbus RTU. The primary purpose of the Unit ID is to provide a means to route requests to a Modbus slave device attached to a Modbus Ethernet to Serial gateway. If you are communicating via a gateway, you will need to create and configure one Modbus Unit ID object for every slave you will access.

If you are communicating with a Modbus TCP/IP device that is not a gateway, you will need to create and configure one Unit ID object. Some of these devices require a specific Unit ID in order to communicate with the device. Other devices may ignore the Unit ID value completely, in which case you can use the default value of 1. Before designating a Unit ID, you should consult the user documentation for the Modbus device you are using.

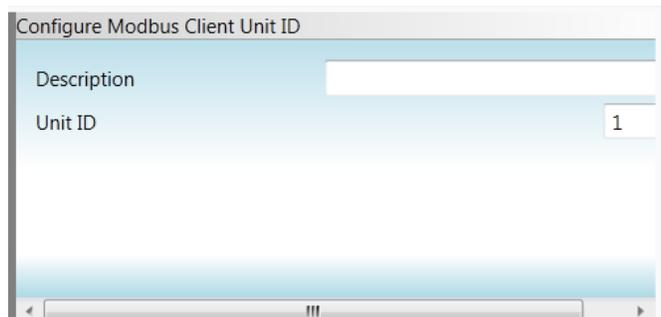
To create an Open Modbus Unit ID object, right click on the Modbus Connection object and select the “Add Unit ID” option. The Protocol Selection panel should appear as shown below.



Clicking on the Unit ID object will allow you to enter the applicable parameters in the configuration panel.

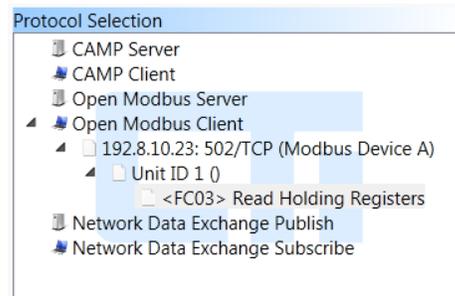
Description: You can enter up to 40 characters of descriptive text. This description will appear beside the Unit ID object in the Protocol Selection panel.

Unit ID: The primary use of the Unit ID is to address Modbus slave devices that are connected using a Modbus Ethernet to Serial Gateway. If you are using a gateway, enter the Modbus address of the target slave device. If you are communicating with a Modbus TCP/IP device that requires a Unit ID, enter the appropriate Unit ID value. Otherwise, use the default value of 1.



Configuring a Modbus Client Request

For each Unit ID object, you must create at least one Open Modbus Client request object. To accomplish this, right click on the Unit ID object and click on the “Add Request” item. The Protocol Selection panel should look like the one below. All requests that are children of this Unit ID object will include the parent Unit ID in the Modbus request. Although, there is no limit on the number of requests per Unit ID, the maximum number of requests per connection is 96.

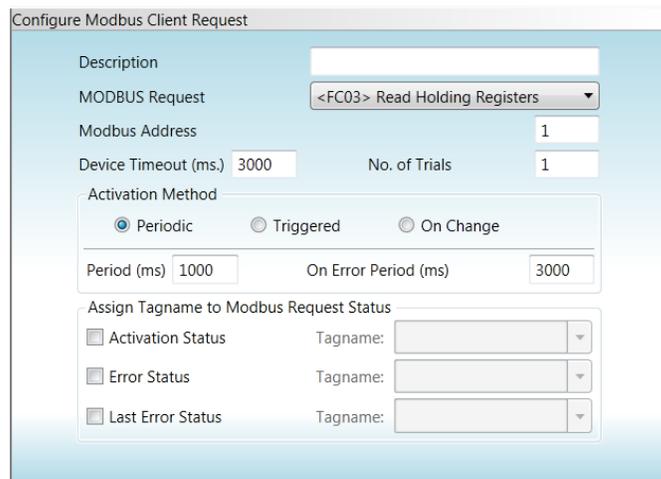


Clicking on the request object will display the following parameters in the configuration panel.

Description: You may enter up to 40 characters describing this request.

MODBUS Request: Select the Modbus request from the list of function codes displayed in the selection box.

Modbus Address: Enter the starting address of the memory type specified in the request. The minimum address is 1; the maximum address is 65536. See your device user documentation to determine its address limitations. *Note: For requests that access multiple Modbus addresses, the data mapped to the request will determine the number of Modbus items to be written or read.*



Device Timeout: The Open Modbus Client protocol expects the Modbus slave/server device to reply to every request. The **DEVICE TIMEOUT** value is the number of milliseconds to wait for a reply from the device (minimum 100ms, maximum= 10,000ms). If the device replies within this time interval, processing continues normally. If the remote device fails to respond within this interval the current attempt to communicate with the device is terminated. Additional attempts to communicate with this device during the same activation period are determined by the “Number of Trials” parameter described in the next section.

Setting this value too small will result in timing out before the device has had a chance to respond, which will have a major effect on device communications. Setting this value too large will result in some unnecessary delays when the device is offline or having problems communicating.

The timeout value should be set to a value slightly larger than the maximum response time of the device. Since the maximum response time is often difficult to determine, you may start with a value that is 2 - 3 times the normal response time of the device. If it is impractical to determine the device response time, start with the default value of 3000ms. This value should allow sufficient time for devices on the local area network to respond while introducing moderate delay when the device is offline or faulty that is acceptable for many applications.

NOTE:

Many Modbus Ethernet to Serial Gateways allow the user to set the timeout period for serial devices connected to it. If a device fails to respond within the timeout period, the gateway will respond with a Modbus error method. If you are using a gateway which supports this capability, make sure that the Modbus Client Device Timeout Value is greater than the gateway timeout value

No. of Trials: This is the maximum number of times that the CAMP Client protocol will attempt to send a request to the device during the current activation of the request. Setting the number or trials greater than 1, allows you to retry sending the request during the activation period. If the request is a WRTE request, additional requests will attempt to send the **same** data each trial. A maximum of 5 trials is allowed.

Additional trials should be used only if the application demands it. Note that each additional trial in which the device fails to respond delays sending other requests on the connection. The amount of delay is approximately equal to the value of the Device Timeout parameter. If you are using the Open Modbus TCP protocol, you should set the number of trials to 1 and rely on TCP to retry sending data, if necessary.

Activation Method: The activation method specifies the event that will initiate the request. The methods include Periodic, Triggered, and On Change.

Periodic: This method initiates the request on a specified time period. When the Periodic method is selected, you can specify the Period and the On-Error period as shown below:

The screenshot shows a configuration window for the 'Periodic' activation method. At the top, there are three radio buttons: 'Periodic' (selected), 'Triggered', and 'On Change'. Below these are two input fields: 'Period (ms)' with the value '1000' and 'On Error Period (ms)' with the value '3000'. At the bottom, there is a checkbox labeled 'Consistency' which is currently unchecked.

Period: This parameter specifies the time interval used when everything is operating normally.

On Error Period: This parameter specifies the time interval used after the device fails to reply to a request within the timeout period. Specifying an **ON ERROR PERIOD** value that is greater than the **PERIOD** value reduces the overhead incurred when polling a device that is off-line or not present.

Consistency: The **CONSISTENCY** checkbox is displayed only for requests that write data. Checking the **CONSISTENCY** checkbox indicates that you want all the data cache members mapped to the request to be updated as a group in the same host controller scan. Since data consistency is usually not required for applications using periodic activation, the default is unchecked (no consistency). See APPENDIX D: DATA CONSISTENCY for guidelines for using data consistency.

Triggered: This method allows user logic in the Host Controller to initiate the request. When the triggered method is displayed, the following parameters are displayed.

The screenshot shows a configuration window for the 'Triggered' activation method. At the top, there are three radio buttons: 'Periodic', 'Triggered' (selected), and 'On Change'. Below these are three options: 'One Shot' (selected), 'Enable', and 'Tagname:' followed by a dropdown menu. At the bottom, there is a checkbox labeled 'Consistency' which is checked.

One Shot: This trigger type initiates a request only when the trigger value transitions from **OFF** to **ON**. If you are using the triggered method, this is the trigger you will most likely use.

Enable: This trigger type initiates requests continually as long as the trigger value is on. After the previous activation is completed, another transaction is initiated. While this method is applicable in certain situations, it is easier to achieve a similar result using the periodic method.

Tagname: This field allows you to select the tagname of a host controller memory location that will be used to set the trigger value.

NOTE:

*When a **one-shot** trigger is used, the Tagname Database item associated with this tagname will automatically be assigned a Cache Refresh Interval (CRI) of 100ms.*

Consistency: The Consistency checkbox is displayed only for requests that write data. Checking the **CONSISTENCY** box indicates that you want all the data cache members associated with request to be updated in the same host controller scan. When used with a one shot trigger it has special meaning: *when consistency is selected, the data mapped to a write request will be obtained from the Host Controller immediately after the trigger transitions from low to high, rather than being read from the cache.* Since this behavior is desirable for most one shot trigger applications, consistency is selected by default. If you are using an enable trigger, you will likely want to disable consistency. See APPENDIX D: DATA CONSISTENCY for additional information regarding consistency.

On Change: The **ON CHANGE** method initiates a request only when the Host Controller data mapped to the request changes in value. *This method can only be used for write requests.* Using the **ON CHANGE** method provides a more timely initiation while reducing unnecessary requests.

Assign Tagname to Modbus Request Status: These fields allow you to associate a Host Controller memory address with the request status. Although this is optional, monitoring the status (especially the error status) is highly recommended.

Activation Status: The **ACTIVATION STATUS** changes to true when the request begins execution. It remains true until the request is completed successfully or terminates with an error, when it is set to False. Although the **ACTIVATION STATUS** can be used with any activation method, it is most useful when used with the **TRIGGERED ONE SHOT** activation method.

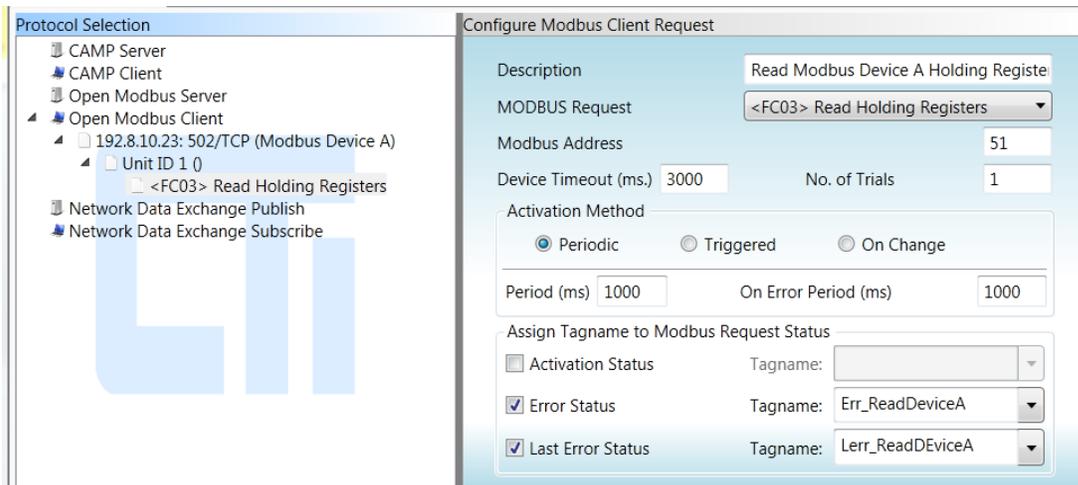
Error Status: Each time a request terminates because of an error, an error code is written to the **ERROR STATUS**. Subsequent errors will overwrite previous error codes. When the request completes successfully, the **ERROR STATUS** is set to 0, clearing any previous error code. *You should assign the Error Status to a tagname when PLC logic is used to process the error condition.*

Last Error Status: The **LAST ERROR STATUS** contains the error code for the last error that occurred when this request was executed, which may be on a previous activation cycle. Unlike the Error Status, the Last Error Status is not set to 0 when the request completes successfully. *If you are not using PLC logic to handle an error condition, you should **always** assign a tagname to the Last Error Status. The last error status is extremely valuable in diagnosing networking problems that may occur.*

You should use a unique tagname for **ACTIVATION STATUS**, **ERROR STATUS**, and **LAST ERROR STATUS**.

Following is an example illustrates the Protocol Selection Panel and the Modbus Client Request configuration panel after configuring a request to read Modbus Holding Registers starting at Modbus Address 51. *Note: number of Modbus Holding Registers to be read will be determined by the cumulative count of the Tagname Database items mapped to the request.*

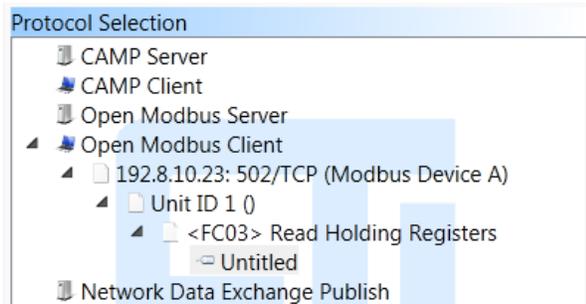
After the request is transmitted, the Modbus client will wait 3000 milliseconds for the device to reply. If it fails to reply within this time, a timeout error will be returned. Only one trial is allowed, since TCP is being used. In this case, the **ON ERROR PERIOD** is set to the same value as the **PERIOD**, 1000ms. The **ERROR STATUS** and **LAST ERROR STATUS** of the request will be monitored using new data items added to the Tagname Database.



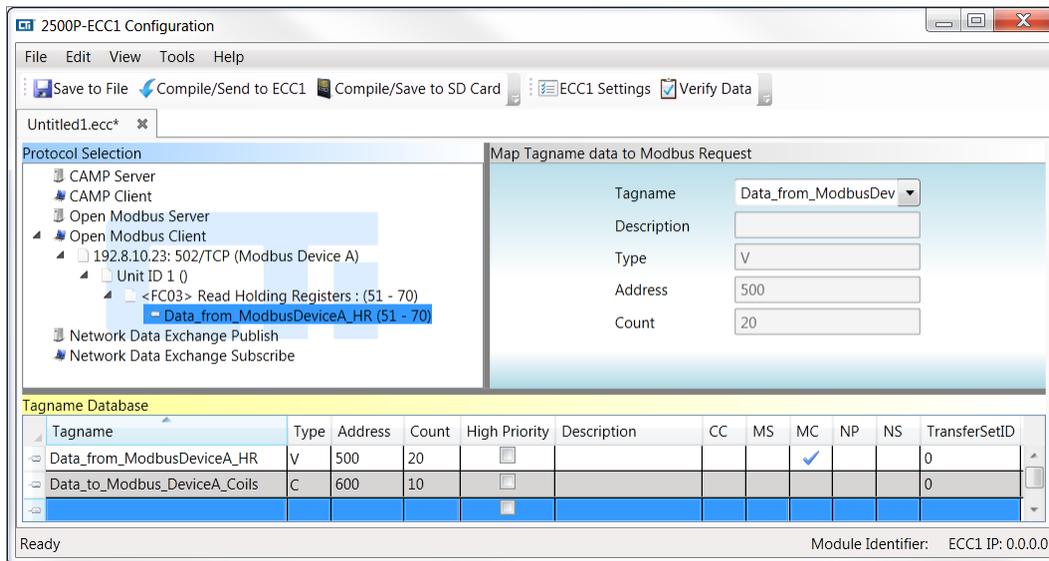
Mapping a Tagname to a Modbus Client Request

After configuring the Modbus request, the next step is to map Host Controller data to the request. To accomplish this task, right click on the request object and select the “Map Tagname” option. The Protocol Selection panel will appear as shown.

When you click on the “Untitled” Data Object, a data mapping display will appear in the Configuration panel.



Selecting the tagname “Data_from_ModbusDeviceA_HR” will result in the display illustrated below.



NOTE:

When mapping Tagname Database items to the request, ensure that the cumulative item count does not cause the request to exceed the address limits of the Modbus device. In addition, ensure that the cumulative item count does not exceed the maximum number of data items per request. See the following table.

Function Code	Description	Maximum Items
FC01	Read Coils	2,000
FC02	Read Discrete Inputs	2,000
FC03	Read Holding Registers	125
FC04	Read Input Registers	125
FC05	Write Single Coil	1
FC06	Write Single Holding Register	1
FC15	Write Multiple Coils	1,968
FC16	Write Multiple Holding Registers	120

5.3.9 Configuring the Network Data Exchange Publisher

You can configure one instance of a Network Data Exchange Publisher, which supply data items to other Network Data Exchange participants that subscribe to them. Each data item is published only when its value changes. The Network Data Exchange publisher uses IP port 9000.

To configure the Network Data Exchange publisher, you must:

- Create entries in the Tagname Database for the Host Controller data you wish to publish,
- Map this data to the Network Data Exchange Publisher.

Creating Tagname Database Items

You will need to specify the Host Controller memory locations that will be used as the source of the data to be published. This is accomplished by adding one or more entries to the Tagname Database, as shown below.

Tagname Database												
Tagname	Type	Address	Count	High Priority	Description	CC	MS	MC	NP	NS	TransferSetID	
Published_Data1	V	2000	20	<input type="checkbox"/>	Data Block 1						0	
Published_Data2	V	2020	10	<input type="checkbox"/>	Data Block 2						0	
				<input type="checkbox"/>								

Ready Module Identifier: ECC1 IP: 0.0.0.0

In this example, two blocks of V memory were created:

- A block of 20 V memory locations, starting with address 2000 was assigned a tagname of "Published_Data1",
- A block of 10 V memory locations starting with address 2020 was assigned a tagname of "Published_Data2".

Configuring a Published Data Item

To configure a published data item, right click on the Network Data Publish object on the Protocol Selection panel and select the "Add Tagname" option. The Protocol Selection panel and Configure Published Data panel should appear as illustrated below.

Protocol Selection

- 📁 CAMP Server
- 📁 CAMP Client
- 📁 Open Modbus Server
- 📁 Open Modbus Client
- 📁 Network Data Exchange Publish
 - 📁 Untitled
- 📁 Network Data Exchange Subscribe

Configure Published Data

Tagname:

Description:

Type:

Address:

Count:

Published Data ID:

To select a data item to be published, click on the Tagname field and select the tagname of an item in the Tagname Database. In this example, the “Published_Data1” tagname has been selected. After selecting the tagname, the configuration window should appear as shown in the accompanying illustration.

Configure Published Data	
Tagname	Published_Data1
Description	Data Block 1
Type	V
Address	2000
Count	20
Published Data ID	0

Next, you will need to enter a Published Data ID. The Data ID is like a subscription number; it uniquely identifies a data item published by this publisher. When you are publishing a block of data, the number you enter will be assigned to the first data address. The remaining data addresses will be automatically assigned consecutive Data IDs. For example, if you published a data item with a count of 3 and entered a Data ID of 10, the first element would be assigned a Data ID of 10, the second a Data ID of 11, and the third a data ID of 12.

In this example we will enter a Data ID of 1 for the block, which would assign a data ID of 1 to the first item and consecutive Data IDs from 2 – 20 to the remainin data elements. After entering the Data ID, the Protocol Selection Panel and Configure Published Data Panel appear as shown below.

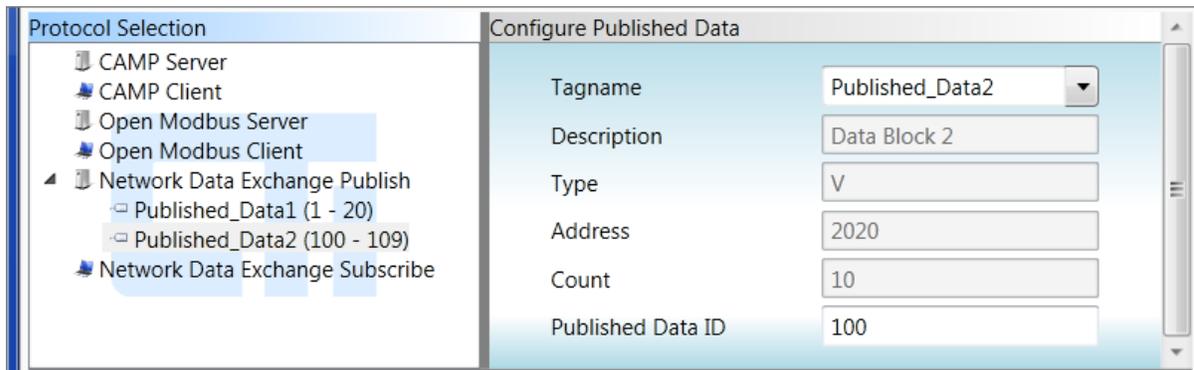
Configure Published Data	
Tagname	Published_Data1
Description	Data Block 1
Type	V
Address	2000
Count	20
Published Data ID	1

To map additional tagnames to the publisher, right click on the Network Data Exchange Publish object and select the “Add Tagname” option. Next select another tagname to be mapped and enter the Data ID. You can enter any number other than one already used for this publisher.

NOTE:
Data IDs are required to be unique only among data items published by a particular publisher. Different publishers can use the same Data ID. For example, publisher A and Publisher B can both use Data IDs 1 -100.

In this example the Tagname “Published_Data2” tagname is selected and a Data ID of 100 is entered as shown in the following illustration.

In this example, Published Data IDs 1 – 20 have been mapped to the Host Controller data represented by the “Published_Data1” tagname and Data IDs 100-109 have been mapped to the Host Controller data represented by the “Published_Data2” tagname.



Note:
Even though Tagnames representing blocks of data have been mapped to the publisher, each Host Controller data item within the blocks are published separately when it changes in value. For example, if V2021 (which is the second item in the block of items represented by the tagname Published_Data2) changes in value, the value will be published along with its Data ID (101).

5.3.10 Configuring the Network Data Exchange Subscriber

To obtain data from a Network Data Exchange publisher, you must establish a connection to the publisher and subscribe to the data items that you want to receive from the publisher. When you subscribe to a data item, the publisher will send the data to you when the value of the data item changes. You can subscribe to multiple data items per publisher and establish connections to up to 20 publishers.

To configure the Network Data Exchange Subscriber, you must:

- Create entries in the Tagname Database for the Host Controller memory addresses that will receive the published data,
- Configure a connection to each Publisher to which you wish to subscribe,
- Map the Tagnames to the published Data ID

Creating Tagname Database Items

You will need to specify the Host Controller memory locations that will be used to receive the published data published. This is accomplished by adding one or more entries to the Tagname Database as shown below. Host Controller data represented by tagnames “PLCA_ProcessData1” and “PLCA_ProcessData2” will be used to receive the data published by PLCA. Tagname “PLCA-ConnStatus” represents a control relay (C50) which will be used to monitor the connection with PLCA.

Tagname Database												
Tagname	Type	Address	Count	High Priority	Description	CC	MS	MC	NP	NS	TransferSetID	
PLCA_ProcessData1	V	500	10	<input type="checkbox"/>	Process Data Set 1 from PLCA						0	
PLCA_ProcessData2	V	5010	2	<input type="checkbox"/>	Process Data Set 2 from PLCA						0	
PLCA-ConnStatus	C	50	1	<input type="checkbox"/>	Status of Connection to PLCA						0	
				<input type="checkbox"/>								

Ready Module Identifier: ECC1 IP: 0.0.0.0

Configuring a Subscriber Connection to a Publisher

To configure a connection to a publisher, right click on the Network Data Exchange Subscribe object in the protocol selection panel and select the “Add Connection” object. After doing so, the “Configure Subscriber Connection to a Publisher” parameters will be displayed.

Configure Subscriber Connection to a Publisher

Connection Name

Publisher IP Address

Use Default Port

Assign Connection Status to Tagname

Connection Status Tagname:

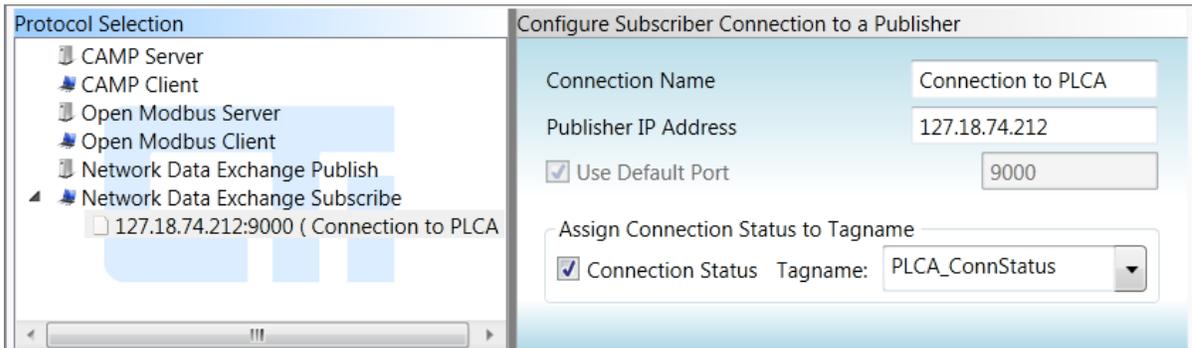
Connection Name: You may enter up to 40 characters to describe the publisher connection.

Publisher IP Address: Enter the IP address of the publisher.

Assign Tagname to Connection Status: By assigning a tagname representing a Host Controller memory location to the connection status, host controller logic can monitor the connection status. See Network Data Exchange Subscriber Error Codes in Appendix A for a list of error codes.

A status code of 0 indicates that the connection status is good. Any non-0 status code should be treated as an error. Mapping the connection status to a control relay, as shown in this example, will achieve this result, since any non-0 value will cause the control relay to turn on.

The following illustrates the appearance of the Protocol Selection and Configuration panels after entering a connection name and Publisher IP address and assigning a Tagname to monitor the connection status.



Mapping Tagname Database Items to Subscribed Data

For each published data item to which you want to subscribe, you must select one or more tagnames representing the Host Controller memory that will receive the published data identify the corresponding published data by specifying the publisher Data ID. Right click on the Publisher Connection object and select the "Add Tagname" option. After selecting this option, the data mapping parameters should be displayed in the Configuration Window as illustrated below.

Tagname: To map the data, select the tagname representing the controller memory that will receive the published data. When you select the tagname, the attributes of the Tagname Database item will be displayed.

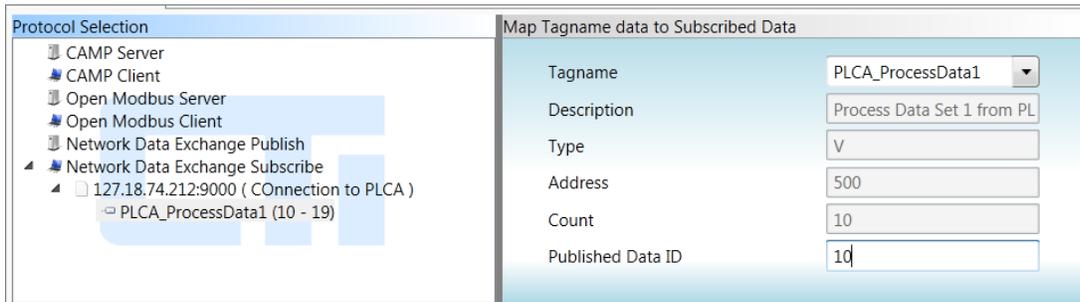
Published Data ID: Enter the Data ID of assigned by the publisher to which you wish to subscribe. When the count is greater than 1, enter the Data

ID of the first data element in the data block. The ECC1 will automatically subscribe to the next consecutive data IDs based on the count. For example, if the Count is 3 and a Data ID value of 5 is entered, the first PLC data address will subscribe to Data ID 5, the second to Data ID 6, and the third to Data ID 7.

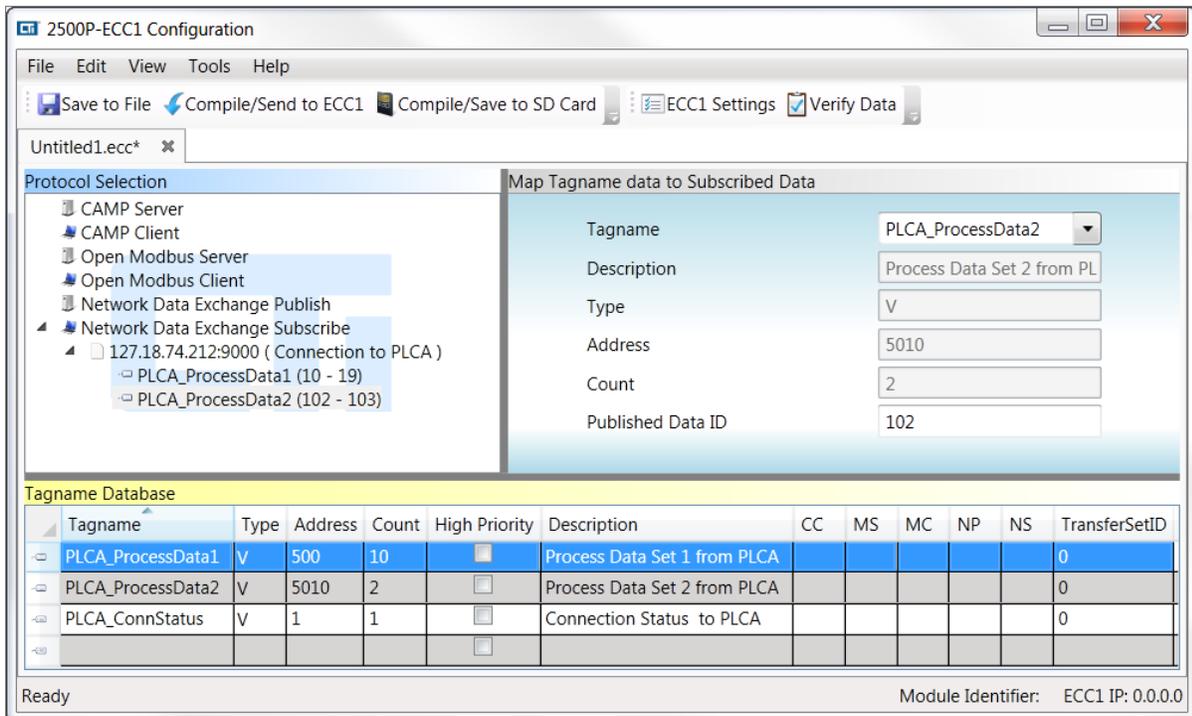


Even though a publisher may have published a large block of data, you are not required to subscribe to all the data. You may subscribe to smaller blocks of data or single non-contiguous data IDs.

The following example illustrates the selection of a tagname representing a block of 10 V memory locations and a subscription to publisher data IDs 10 - 19.



You can subscribe to additional data by selecting another tagname and assigning a different Publisher ID. See the illustration below.



CHAPTER 6 UPDATING FIRMWARE

6.1 Overview

The CTI 2500P-ECC1 module stores the operating firmware in non-volatile flash memory. You can replace the current operating firmware with a different version to correct problems or add new features. During this procedure, the new firmware will be copied to controller RAM, verified, and then written to flash memory.

The normal method for updating the firmware is to use the firmware update function in the CTI 2500P-ECC1 Configuration program, which transfers the firmware update file to the 2500P-ECC1 module over an Ethernet link. An alternate method, which allows you to update the firmware using an SD card, is available for situations where the Ethernet transfer is not practical or where the SD card method can be more efficient, such as updating multiple modules.

During the firmware update process, various status codes and error codes may be displayed on the 2500P-ECC1 Multi-Segment Display (MSD).

- Firmware update status codes, which start with “**U**” followed by two numeric digits, indicate what is happening in the update process. See *Section 6.4* for a list of the firmware update status codes.
- Firmware update error codes, which start with “**E**” followed by two numeric digits, are used to indicate the specific error that occurred. When an error occurs during firmware update, the process will stop. You must cycle power to the base to restart the firmware update procedure. See APPENDIX A: ERROR CODES for a list of error codes and corresponding corrective actions.

Prior to updating firmware, you will need to obtain a firmware update file for the 2500P-ECC1 module. This file can be downloaded from the CTI website <http://www.controltechnology.com/downloads/>. After obtaining this file, you should save it to a file on your PC or on an accessible network drive.

6.2 Ethernet Firmware Update Method

The Ethernet firmware update method provides a fast efficient, means for updating firmware on the product. Prior to attempting to update firmware using this method, you should ensure that the PC used to perform the update can communicate with the IP address of the 2500P-ECC1 module you want to update. The steps in the firmware update process are described in the following paragraphs.

1. **Prepare the ECC1 Module for Firmware Update over Ethernet**
 - a. Disconnect power and remove the ECC1 module from the base, if installed,
 - b. Set SW1 to the closed (Firmware Update) position,
 - c. Ensure that SW2 (Firmware Update Method) is in the Open (Ethernet) position,
 - d. Install the Product into the base and apply power.

2. **Monitor the ECC1 Module Startup:**

When the ECC1 module starts up, it will attempt to execute the firmware file loader. If an error is encountered, an error code will be displayed on the Multi Segment Display (MSD) and the update process will be terminated. After correcting the problem, you must restart the module.

If there is no error, status code **U01** will be displayed on the MSD, indicating that the Product is ready to accept the firmware update file.

3. **Transfer the firmware update file to the ECC1 module:**

- a. Start the 2500P-ECC1 Configuration Program
- b. Under the **Tools** Menu item select Firmware Update option. You will be prompted to select a firmware file to be transferred.
- c. After selecting the file, you will be prompted to enter the IP address of the 2500P-ECC1 module you want to update.
- d. After you enter the IP address, the 2500P-ECC1 Configuration Program will attempt to connect to a 2500P-ECC1 module using the IP address you entered. If the connection can be established, the program will validate that the module is prepared for firmware update over Ethernet and initiate the file transfer. If an error is encountered, the program will display an error message and wait for corrective action.
- e. During the file transfer, the ECC1 module Multi-Segment Display (MSD) will display **U02**, indicating that file transfer is taking place. If a file transfer error occurs, an error code will be displayed and the firmware update process will be terminated (see APPENDIX A: ERROR CODES). After correcting the problem, you must restart the module, and re-initiate the firmware update procedure.
- f. When the file transfer is completed successfully, the 2500P-ECC1 Configuration Program will display a message indicating the update file was successfully transferred.
- g. After the file is transferred, the ECC1 module will validate the file and, if the validation is successful, begin replacing the firmware stored in flash memory. While this process is progressing, various status messages will be displayed. If an error occurs, an error code will be displayed on the module MSD and the firmware update process will be terminated. After correcting the problem, you must restart the module, and re-initiate the firmware update procedure.

NOTE:

Do not remove power from the module while the firmware is being replaced.

- h. When the firmware update is complete, status Code **U00** will be displayed on the Module MSD.

4. **Return the ECC1 module to normal operating mode.**

- a. Disconnect power and remove the ECC1 module from the base,
- b. Set SW1 to the Open (Normal Operation) position,
- c. Re-insert the module into the base.
- d. Re-apply Power.

6.3 SD Card Firmware Update Method

The SD card firmware update method provides an alternate method of updating the ECC1 module firmware. Prior to updating the firmware, you must copy the firmware update file to the root directory of an SD card.

NOTE:

The firmware update process will not work if there is more than one 2500P-ECC1 firmware update file in the root directory of the SD card.

1. Prepare the 2500P-ECC1 Module for Firmware Update using an SD Card

- a. Disconnect power and remove the Product from the base, if installed,
- b. Remove any SD card installed in the product and install the SD card containing the firmware update file,
- c. Set SW1 to the closed (Firmware Update) position,
- d. Set SW2 (Firmware Update Method) to the closed (SD Card) position.
- e. Install the Product into the base and apply power.

2. Monitor the 2500P-ECC1 Firmware Update Process:

When the ECC1 module starts up in firmware update mode with SW2 set to the SD card position, the module will attempt to copy the firmware update file contents from the SD card to ECC1 module RAM. Status code **U02** will be displayed on the Multi-Segment Display (MSD) while this is taking place. If an error is encountered, such as a missing firmware update file, an error code will be displayed on the Multi-Segment Display (MSD) and the firmware update process will be terminated (see APPENDIX A: ERROR CODES). You must correct the problem and restart the firmware update process.

Once the file has been copied to RAM and verified, the ECC1 module will proceed with replacing the current firmware in flash. While the firmware in flash is being replaced, various status codes will be displayed on the MSD. If an error is encountered while updating flash, an error code will be displayed on the MSD and the firmware update process terminated. You must correct the error and restart the firmware update process. After the firmware update process has completed successfully, status code **U00** will be displayed on the MSD.

NOTE:

Do not remove power from the module while firmware is being updated.

3. Return the ECC1 Module to the Normal Operating Mode:

- a. Disconnect power and remove the ECC1 module from the base,
- b. Replace the firmware update SD card with the operating SD card,
- c. Set SW1 to the open (Normal Operation) position,
- d. Set SW2 to the open position (recommended) ,
- e. Re-install the ECC1 module into the base and apply power.

6.4 Firmware Update Status Codes

The following status codes indicate progress in updating firmware. Under normal circumstances, many actions happen so fast that a status code will not be displayed.

Status Code	Status
U00	Module firmware successfully updated
U01	Waiting for firmware file transfer to begin
U02	Firmware file transfer in progress
U03	Validating downloaded Image File header in DRAM
U04	Validating downloaded Image File in DRAM
U05	Building the CTI Image File in DRAM.
U06	Erasing the application data communications area
U07	Erasing the FLASH area to hold the CTI Image File
U08	Writing the CTI Image File to FLASH
U09	Verifying the CTI Image File was written to FLASH properly
U10	Searching for an Image Trailer in FLASH
U11	Searching for a Boot Loader Image in the image file
U12	Validating the Boot Loader Image found in FLASH
U13	Copying the Boot Loader Image from FLASH to DRAM and verifying copy
U14	Erasing the area in FLASH to hold the downloaded Boot Loader
U15	Writing the downloaded Boot Loader into FLASH

APPENDIX A: ERROR CODES

Error Code Sections
Initial Startup Error Codes
Operational Error Codes
Protocol Error Codes
Firmware Update Error Codes
Configuration Error Codes

Initial Startup Error Codes

The following error codes may be displayed on the ECC1 Multi-Segment Display (MSD) during initial startup.

Error Code	Description	Comments
E01	Invalid Product Model	The firmware stored in flash is not compatible with the product model. Contact CTI support.
E81 – E99	Invalid Main Application	An error was encountered while loading the application firmware. This is likely due to a hardware problem. Contact CTI support.

Operational Error Codes

Operational Error Codes are returned by the ECC1 Module application firmware. Operational Error Codes are presented on the ECC1 Multi-Segment Display (MSD) as a 3 numeric digits, preceded by the characters “Err”. In situations where multiple errors are present, the error code for the highest priority error will be displayed. Once the ECC1 Module web server is running, operational errors can be viewed in the **ERROR CODES AND DESCRIPTIONS** web page.

[Back to Chapter Top](#)

Error Code	Description	Comments	Recovery
010	No Stored IP Address Using Auto-Assigned IP Address	Not displayed on MSD	Create a configuration containing an IP address for the ECC1 and transfer it to the Module. See Section 0
020	Ethernet Port Switch Error	Unable to configure switch	Restart the ECC1 module. If error persists contact CTI support.
030	Manufacturing Data Error	Checksum error detected	Restart the ECC1 module. If error persists contact CTI.

Error Code	Description	Comments	Recovery
040	OS Component Start Error	Components include the Ethernet Stack, TCP/IP Stack, GH file system, RAM disk, FTP Server, HTTP Server, and SD card stack.	Restart the ECC1 module. If error persists contact CTI support.
120	Configuration File Processing Error	An error was encountered while copying the configuration file to RAM	This may indicate a defective SD card. Try reformatting the SD card. If the error persists after transferring the configuration files to the reformatted card, replace the card.
130	SD Card Not Found	SD Card cannot be accessed, missing or unsupported type.	Insert a compatible SD card See Section 3.1.1
140	Configuration File Missing	The SD card does not contain a configuration (.ini) file.	See recovery procedures for Errors 160 – 180 below.
150	Incompatible Configuration File	The configuration file version cannot be used with the current ECC1 firmware version.	Update firmware to the required version. See CHAPTER 6.
160	Invalid Configuration File	The contents of the configuration file are not valid.	Using the 2500P-ECC1 Configuration program: <ul style="list-style-type: none"> • Select the "Compile/Send" tool bar item to create a new set of configuration files and transfer them to the ECC1 module OR • Select the "Compile/Save to SD Card" tool bar item to create and save a new set of configuration files to an SD card. Install the SD card in the ECC1 module and re-power the module
170	Missing Application Logic Program	Application logic program file cannot be found on the SD card	
180	Logic Engine Start Error	Unable to load and execute application logic.	
205	No Host Controller IP Address	Configured Host Controller IP address was 0.0.0.0	Modify configuration to include address of host controller and transfer configuration to the ECC1 module.

Error Code	Description	Comments	Recovery
210	Host TCP Connection Failure	Unable to connect to Host Computer.	<ul style="list-style-type: none"> • Ensure that the Host controller is online, • Ensure that there is a network path between the ECC1 module and the Host controller. • Confirm that the configured Host Controller IP address matches the one being used • Confirm that the ECC1 module IP Address and the Subnet mask is compatible with the Host Controller IP address and subnet mask (see Appendix B) • Ensure that the Host Controller firmware is the correct version.
220	Host/ECC1 Firmware Incompatible	The host controller firmware version and ECC1 firmware version are not compatible.	Ensure that the ECC1 and Host Controller are updated to compatible firmware versions. See CTI web site: http://www.controltechnology.com/support/software_revision/
230	Host Registration Failed	The ECC1 module cannot register because the maximum number of ECC1 modules is already communicating with the Host Controller.	If you are replacing an ECC1 module, ensure that the module to be replaced is disconnected before connecting the replacement module.
240	Host Link Inactive	The ECC1 module is not communicating with the Host Controller.	<ul style="list-style-type: none"> • Ensure that the Host controller is online, • Ensure that the ECC1 module and the Host Controller are connected to the same local area network, • If a VLAN is used, ensure that the Host Controller and the ECC1 module are on the same VLAN.

Error Code	Description	Comments	Recovery
310	Host Controller Fatal Error	The Host Controller is in fatal error state. All protocols are suspended except CAMP server.	Clear the Host Controller Fatal Error.
320	Host in Program Mode	Occurs only when The “Disable Protocols when PLC in Program Mode” option is enabled in the configuration and Host Controller is in Program mode. All protocols are suspended except the CAMP server.	If this option has been enabled in error, change the configuration.
330	Configured Data Point Inaccessible	Unable to access one or more data addresses specified in the Tagname Database configuration.	Compare the Host Controller Memory configuration with the Tagname Database entries to determine where the problem exists. Change Host Controller memory configuration or revise the Tagname Database entry.
410	CAMP Server Error	CAMP Server encountered a Startup error	Restart the ECC1 module. If the problem persists reload the ECC1 firmware. If the problem continues contact CTI support.
415	CAMP Client Error	CAMP client encountered a startup error or is unable to connect to one or more remote network devices	Ensure all remote devices are online Confirm that there is a network path between the ECC1 module and the remote device. Ensure that the configured IP address for the remote device matches the remote device.
420	Modbus Server Error	Modbus Server encountered a Startup Error	Restart the ECC1 module. If the problem persists, ensure that the Modbus Server data blocks are mapped to accessible memory addresses. If the problem persists, reload the ECC1 firmware. If the problem continues contact CTI support.

Error Code	Description	Comments	Recovery
425	Modbus Client Error	Modbus client encountered a startup error or is unable to connect to one or more Modbus server devices.	Ensure all Modbus devices are online Confirm that there is a network path between the ECC1 module and the remote device. Ensure that the configured IP address for the remote device matches the remote device.
430	Data Exchange Publisher Error	Data Exchange Publisher encountered a startup error	Restart the ECC1 module. If the problem persists reload the ECC1 firmware. If the problem continues contact CTI support.
435	Data Exchange Subscriber Error	Data Exchange Subscriber encountered a startup error or is unable to connect to one or more publishers.	Ensure publishers are online Ensure that the correct IP address has been configured for each subscriber to publisher session.

[Back to Chapter Top](#)

Host Controller Status Error Handling

Protocol managers react to certain Host Controller conditions as specified below:

Protocol Manager Type	Host Controller Connection Loss	Host Controller Fatal Error	Host Controller In Program Mode
CTI CAMP Server	<p>Respond to CAMP data read, write, or Packed Task Code requests for Host Controller data with error code 0xBC.</p> <p>Respond to raw task code data requests to access controller data with task code error 17 (no response from PLC).</p>	<p>Continue to respond to CAMP and raw task code requests to read PLC status words.</p> <p>Respond to CAMP data read, write, or packed task code requests to access other Host Controller data with error code 0xBB.</p> <p>Respond to raw task code requests for other Host controller data with task code error 07 (fatal error detected).</p>	<p>The Host Controller Program/Run mode shall not affect the operation of the CTI CAMP Server.</p>
Other Protocols	<p>When either condition is detected, if the protocol managers are not already in the disabled (non-operational state), they will be placed in the disabled state after closing all connected TCP connections. The protocols shall remain in the disabled state as long as either condition exists. When the Product is successfully re-connected to the Host controller and the Host Controller is not in fatal error mode, the Product shall attempt to start up all configured protocols and re-establish all specified TCP connections.</p>		<p>While the Host Controller is in Program mode and the "DISABLE PROTOCOLS WHEN HOST CONTROLLER IS IN PROGRAM MODE" option is enabled, the Straton protocols will be placed in the disabled state. When the Host Controller transitions from Program to Run mode, if the Straton protocols are in the disabled state, the Product shall attempt to start all configured protocols and to re-establish all specified TCP connections.</p>

[Back to Chapter Top](#)

Protocol Error Codes

Protocol Error Codes are returned by the protocol tasks. They indicate problems in sending or receiving messages using the protocol. Server protocol tasks return error codes to their clients within the protocol messages. Client protocol tasks allow protocol error codes, including those returned by servers, to be mapped to a Host Controller memory address. Similarly, publisher protocol tasks may return error codes to their subscribers. Subscriber protocol tasks allow error codes to be mapped to a Host controller memory address.

[Back to Chapter Top](#)

[CAMP Server Error Codes](#)

[CAMP Client Error Codes](#)

[Modbus Server Error Codes](#)

[Modbus Client Error Codes](#)

[NDE Subscriber Error Codes](#)

CAMP Server Error Codes

The CAMP Server may return the following CAMP error codes to the client.

[Back to Protocol Errors Top](#)

Error Code		Description	Corrective Action
Decimal	Hex		
116	74	Server detected a checksum error	This usually results from a transmission error. Retry request. If the error persists, it could result from a client checksum generation error
117	75	Server Received an unsupported or invalid command	The command is not supported by the server or is not valid for a request. Correct the client request.
118	76	Invalid Character Received	The protocol accepts only characters 0-9, A-F, ? ,[,] Correct the client request.
119	77	Odd Number Of Characters In Packet	The protocol requires that the message contain an even number of characters Correct the client request
129	81	Write Request contains no data to write	A message requesting to write data contained no data Correct the client request.
131	83	Attempted to Write to Address 0	A message requesting to write data contained an address of 0, which is invalid. Correct the client request.
143	8F	Attempted to read 0 Words	A message requesting to read data specified the number of words as 0. Correct the client request.
144	90	Unsupported Data Element type	The data element type specified in the request message is not supported by the CAMP server. Correct the client request. See APPENDIX C: CAMP SERVER SUPPORT for a list of supported data element types.
145	91	Tried To Read More Than 256 Words	A CAMP message cannot contain more than 256 words Modify the client to request less data.

Error Code		Description	Corrective Action
Decimal	Hex		
147	93	Maximum CAMP Response Size Exceeded	The resulting size of a reply to a Packed Task Code request would exceed the maximum size of a CAMP message. Correct the client request.
148	94	Maximum Packed Task Code Requests Exceeded	The number of Packed Task Code requests contained in the CAMP message exceeded the maximum allowed.
149	95	Maximum Packed Task Code Length Exceeded	The length field of a task code request contained in a CAMP Packed Task code message exceeded the maximum value. Correct the client request
172	AC	Cannot Read From Memory Address	The Host Controller was unable to read from the requested memory address. This usually occurs when the request exceeds the maximum address available in the host controller. Consider the following corrective actions: <ul style="list-style-type: none"> • Change the client request to change the address to be read • If the client is reading a block of data addresses, ensure that the number of addresses requested does not cause the maximum address to be exceeded. • Change the Host Controller memory configuration so that the requested address(es) are available
173	AD	Cannot Write To Memory Address	The Host Controller was unable to write to the requested memory address. This usually occurs when the request exceeds the maximum address available in the host controller. Consider the following corrective actions: <ul style="list-style-type: none"> • Change the client request to change the address to be read • If the client is reading a block of data addresses, ensure that the number of addresses requested does not cause the maximum address to be exceeded. Change the Host Controller memory configuration so that the requested address(es) are available
187	BB	Host Controller In Fatal Error	All protocols except the CAMP server are suspended. If the CAMP server is enabled, data access will be limited. After the fatal error is cleared, all configured protocols will resume. If the CAMP server is enabled, full data access will be restored.
188	BC	No Communications With Host PLC	The ECC1 Module is unable to communicate with the designated Host Controller. Check for the following situations: <ul style="list-style-type: none"> • The Ethernet connection between the ECC1 module and the Host Controller has been disrupted. • The Host Controller is offline/powered down. • The IP address of the Host Controller has changed
189	BD	Camp Server Is Not Enabled	If CAMP accessed is desired, enable the CAMP server using the 2500P-ECC1 CAMP Server.
190	BE	CAMP Write Request for 32 bit data (long or float) does not contain enough data words	2 words per long or floating point number is required
191	BF	CAMP Request is attempting to write to read-only memory location	See APPENDIX C: CAMP SERVER SUPPORT to determine read-only data types.

Error Code		Description	Corrective Action
Decimal	Hex		
198	C6	Unable to obtain data from Host Controller	The Host Controller did not respond to a request to read this data. If this error persists, contact CTI.
199	C7	Request Queue Limit Exceeded	This indicates that the maximum number of data access requests are waiting to be serviced. If this error persists, reduce the rate of requests.
200	C8	ECC1 Host Controller Data Cache Full	This indicates that no more points can be added to the cache at this time. If this error persists, limit the request load or divide the load between two ECC1 modules.

CAMP Client Error Codes

The following table describes the errors that CAMP client may detect in the reply request or in attempting to communicate with another device that implements a CAMP server. Also see the [CAMP Server Error Codes](#) section above for a list of error codes that could be returned to the CAMP client by a CAMP server.

[Back to Protocol Errors Top](#)

Error Code		Description	Comments
Decimal	Hex		
107	6B	Read Word Count Error	The number of words returned in the reply does not equal the number of words requested. This indicates a server error. Contact CTI.
123	7B	Missing Delimiter Detected	The reply is missing a message delimiter. This Indicates a server error. Contact CTI.
124	7C	The reply contains a bad checksum	Usually results from transmission error. Retry the request. This may result from a server checksum generation error. Contact CTI.
125	7D	Invalid command type in Reply	This indicates a server error. Contact CTI.
126	7E	Invalid Character in Reply	The protocol accepts only characters 0-9, A-F, ?, [, ,] This indicates a server error. Contact CTI.
127	7F	Odd Number Of Characters In Reply	The protocol requires that the message contain an even number of characters. This indicates a server error. Contact CTI.
128	80	Invalid Error Type in Reply	The reply contains an invalid error type character. This indicates a server error. Contact CTI.
130	82	Word Count Error	The number of data words in the reply does not match the number of words requested. This indicates a server error. Contact CTI.
132	84	Word Write Count Error	The number of data words written does not match the number of words in the reply. This indicates a server error. Contact CTI.
146	92	Message ID Mismatch	The message ID of the reply does not match the Message ID of the response. This could occur if the reply arrives after a timeout and is interpreted as a reply to the next request. If this error persists, increase the timeout period.

Error Code		Description	Comments
Decimal	Hex		
152	98	Unable to Connect to remote device	This could be caused by one of the following: <ul style="list-style-type: none"> • The remote device is offline or powered down. • The maximum number of TCP connections to the remote device has been reached, • The IP address specified in the configuration does not match the IP address of the remote device, • There is no network path between the ECC1 module and the remote device.
155	9B	Error Reading Ethernet	The CAMP client was unable to read data from the Ethernet interface. If this error persists, contact CTI
156	9C	Error Writing Ethernet	The CAMP client was unable to write data to the Ethernet Interface. If this error persists, contact CTI.
157	9D	Timeout Error	A reply from the remote device was not received within the timeout period specified in the configuration. This could be caused by one of the following: <ul style="list-style-type: none"> • The remote device is offline (when using UDP) • The IP address specified in the configuration does not match the IP address of the remote device (when using UDP), • There is no network path between the ECC1 module and the remote device (when using UDP), • The configured timeout value is too small. It does not allow enough time for the remote device to respond.

Task Code Error Codes

To aid in diagnosing error, the following presents a list of all task code error codes returned in a task code 00 error reply.

Error Code (Hexadecimal)	Description
01	Reset Current Transaction
02	Address out of Range (other than ladder logic)
03	Requested data not found
04	Illegal Task Code Request
05	Request exceeds available memory
06	Diagnostic fail on power up
07	Fatal error detected
08	Keylock/password protection error
09	Incorrect amount of data sent with request
0A	Illegal request in current operational mode
0B	Network was not deleted
0C	Attempted write operation did not verify
0D	Illegal number of ASCII characters received
0E	Illegal request when running from EEPROM or flash
0F	Data not inserted
10	Data not written

Error Code (Hexadecimal)	Description
11	Illegal data sent with command
12	Invalid operation with NIM local/remote mode (obsolete)
13	The store and forward buffer is busy
14	No response from special function module
15	Illegal instruction found in program memory (may include memory address)
16	Attempted to write to protected variable (e.g. TCC, TCP)
17	No response from PLC (e.g single scan not performed)
18	Requested memory size exceeds total available memory
19	Requested memory size is not multiple of block allocation size
1A	Requested memory size is less than minimum defined value
1B	Requested memory size is larger than maximum defined value
1C	PLC busy – cannot complete requested operation
1D	Communications error in HOLD mode – Transition to Run not allowed
1E	Port Lockout is active
1F	Attempting to delete active program via reconfiguration
20	Program load in progress or invalidated
21	I/O configuration error –too many points
22	I/O configuration error – attempt to assign Output point to multiple applications
23-3E	Unused
3F	Bus error detected
40	Operating system error detected
41	Invalid control block type
42	Control block number out of range
43	Control block does not exist
44	Control Block already exists
46	Offset out of range
47	Arithmetic error detected while writing Loop or Loop Alarm parameters
48	Invalid SF program type
49	Instruction number or RAMP/SOAK step number out of range
4A	Attempt to access an integer variable as a real
4B	Attempt to access a real variable as an integer
4C	Task code buffer overflow – too much data requested
4D	Control block size error (cannot exceed 32767 bytes)
4E	Attempt to write to a read-only variable
4F	Invalid data type for this operation
50	Task code request buffer too large
51	Invalid SF statement size
52	Invalid return value
53	Attempt to execute a cyclic statement in a non-cyclic SF program
54	Control block is disabled
55	Control block is not disabled
56	Attempt to perform a FSTR_OUT SF statement on an empty FIFO
57	Attempt to perform a FSTR_INT SF statement on a full FIFO
58	Stack overflow while evaluating a MATH, IF-THEN, or IMATH statement

Error Code (Hexadecimal)	Description
59	Maximum SF subroutine nesting level exceeded (maximum = 4)
5A	Arithmetic Overflow
5B	Invalid operator in and IF, MATH, or IMATH expression
5C	S memory overflow
5D	Attempt to divide by 0
5E	FIFO is incompatible with FSTR statement
5F	FIFO is invalid
60	Invalid data type code
61	RAMP/SOAK step type mismatch
62	Invalid code
63-67	Unused
68	Data cache full (ECC1 module)
69	CAMP server not enabled (ECC1 module)
70	NITP message size error too small, too big, odd number of characters)
71	NITP message contains an invalid character
72	NITP message length does not match the actual message length
73	NITP Checksum error
74-8F	Reserved
90	Invalid IP parameter
91	IP parameter set failed
92-FD	Unused
FE	Normal I/O Error
FF	DAM error

Modbus Server Error Codes

The following error codes may be returned by the Modbus Server.

[Back to Protocol Errors Top](#)

Error Code	Description	Comments
1	Unsupported MODBUS function	<p>The function code received in the query is not supported by the server. This can result from the following conditions:</p> <ul style="list-style-type: none"> The client is requesting a function code not supported by the ECC1 Modbus server. See Section 2.3 for a list of supported function codes. The configured data blocks do not support the requested function code. For example, the client is attempting to read input registers but no data block of containing input registers exists.
2	Invalid MODBUS address.	<p>The Modbus query attempts to access an address that is not within the range of MODBUS addresses that have been configured for the requested Modbus data type. This can result from the following conditions:</p> <ul style="list-style-type: none"> The client is erroneously requesting an unsupported address, The server data block is misconfigured. The starting address of the data block is in error or the size is too small.
3	Invalid MODBUS value.	<p>A value contained in the query data field is not an allowable value for the slave.</p> <p>This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It does not mean that a data item to be written has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value in any particular register.</p> <p>This indicates a problem with the Modbus client.</p>
4	MODBUS server failure.	<p>An unrecoverable error occurred while the server was attempting to perform the requested action.</p> <p>If this error persists , contact CTI</p>

Modbus Client Error Codes

[Back to Protocol Errors Top](#)

Error Code	Description	Comments
0	Request Successful	No Error
1	Unsupported MODBUS function	The function code in the query is not supported by the Modbus slave device. Some Modbus devices may return this error if the function is not allowed for the data you are attempting to access, for example attempting to write data to which you have only read access. You should review user documentation for the device and make corrections as necessary.
2	Invalid MODBUS address.	The Modbus query attempted to access an address that is not a valid MODBUS addresses for the device. Some Modbus slave devices may also return this code to indicate that that access is inhibited (for example, write protected).
3	Invalid MODBUS value.	A value contained in the query data field is not an allowable value for the Modbus slave device.
4	MODBUS server failure.	An unrecoverable error occurred while the Modbus device was attempting to perform the requested action.
6	Modbus Server Busy	The Modbus slave device was unable to service the request because it was busy with other tasks.
8	Data Parity Error	This error is not expected, but is included for completeness. This error is returned in response to Function Codes 20 and 21, indicate that the extended memory area failed to pass a consistency check. The ECC1 Modbus client does not use FC20 or FC21.
10	Invalid Gateway Path	This response is returned by a Modbus Ethernet to serial gateway. It indicates that the gateway was unable to allocate a path to the device. This typically that the Unit ID (used to select the path) does not match the address of a slave on the serial network.
11	Gateway Target Failed	This response is returned by a Modbus Ethernet to Serial gateway. It indicates that no response was obtained from the target device. The device may not be connected, may be powered down, or the serial parameters may be misconfigured. NOTE: some devices may return this error even when they don't have an Ethernet/Serial gateway. This can sometimes be resolved by using a different UnitID like 255. Consult your device documentation or contact the Device manufacturer.

Error Code	Description	Comments
128	Communication Timeout	This indicates that the target Modbus Device failed to respond within the designated timeout period.
129	Bad CRC16	Indicates a data transmission error. Reported only by Modbus slave devices attached to serial links. Modbus TCP/IP protocol does not contain a Modbus CRC.
130	Lost Connection	The TCP connection to the device has been lost.

Network Data Exchange Subscriber Error Codes

The following error codes may be returned by the Network Data Exchange Subscriber.

[Back to Protocol Errors Top](#)

Error Code	Description	Comments
0	No Error	No Error
1	Re-establishing lost connection	The subscriber is attempting to reconnect to the publisher
2	TCP Error Occurred	The subscriber experienced a non-recoverable TCP error, causing the connection to be closed.
3	Timeout Error	Nothing has been received from the Publisher within the expected time interval.
4	Other Error Detected	General Connection Error

[Back to Protocol Errors Top](#)

NOTE:

If a connection is interrupted, the Network Data Exchange subscriber will automatically attempt to reconnect. Consequently, the error codes will typically continue to cycle until the connection is good. If your Host Controller logic is performing an action based on error, you should condition the action when the controller memory address associated with the error status is non-zero rather than the monitoring individual error codes.

Firmware Update Error Codes

If an error occurs during the firmware update procedure, the procedure will stop and wait for corrective action. An error code will be displayed on the MSD. The following table describes the error codes and the corrective action.

[Back to Chapter Top](#)

Error Code	Description	Corrective Action
E16	Firmware update failed	Rerun the firmware update procedure. If the error persists, contact CTI.
E17	Unable to write the firmware update status flag.	Rerun the firmware update procedure. If the error persists, contact CTI.
E18 – E25	Unable to locate or execute the firmware file loader	Contact CTI.
E26	An error was encountered when attempting to open the firmware update file.	Reformat the SD card, copy the firmware file to the SD card, and rerun the firmware update procedure.
E27	The firmware update file could not be read.	Delete the firmware update file, re-copy the firmware update to the SD card, and rerun the firmware update procedure. If the error persists, try using a new SD card for the firmware update.
E28	The SD card directory could not be read	Reformat the SD card, copy the firmware update file to the SD card, and rerun the firmware update procedure.
E29	No firmware update (.ffl) file could be found on the SD card or more than one firmware update file was found.	One and only one firmware update file is allowed on the SD card. This file must be in the root directory. Take the necessary action to ensure that the desired firmware update file (and only the file), is located in the root directory. Once this has been done, rerun the firmware update procedure.
E30	Unsupported SD card inserted	The ECC1 module supports an SDSC (SD Standard Capacity) or an SDHC (SD High capacity SD card).
E31	Unused in ECC1 product	.
E32	SD Card memory region is not available	Rerun the firmware update procedure. If this error persists contact CTI.
E33 – E73	Error erasing, writing, or verifying, flash.	Retry the firmware download procedure. If the error persists, contact CTI Product Support.
E74 - E79	The file transferred to the ECC1 Module does not appear to be a valid ECC1 Firmware Update File.	Ensure that the file you transferred is a firmware update file. If so, retry the firmware update procedure. If not, obtain a valid firmware update file and repeat the firmware update procedure.
E80	An error occurred while the firmware update file was being transferred to the controller.	Ethernet Transfer Method The network path between the PC and the ECC1 module may have been disrupted. Retry the firmware update procedure.
		SD Card Transfer Method The SD card may have failed. Retry the firmware update procedure. If the error persists, retry using a different SD card.

Configuration Error Codes

[Back to Chapter Top](#)

Error Code	Error Message	Comments
V1	Address/Count combination for tagname exceeds maximum C memory address	This error can occur if the address value exceeds the maximum or, if the count is greater than 1, the count value causes one or more addresses in the block to exceed the maximum. NOTE: The actual C (control relay) addresses available in the Host controller may be less than the configuration maximum, since the number of control relay addresses varies by controller model
V2	User defined tagname cannot have 2 consecutive underscores	You must change the tagname to comply with tagname requirements. If the tagname was assigned to a protocol, you must modify the protocol data mapping.
V3	Invalid characters in tagname. User defined tagnames are case-insensitive and can contain only letters, numbers and underscores.	You must change the tagname to comply with tagname requirements. If the tagname was assigned to a protocol, you must modify the protocol data mapping.
V4	User defined tagname cannot start with a number	You must change the tagname to comply with tagname requirements. If the tagname was assigned to a protocol, you must modify the protocol data mapping.
V5	User defined tagname cannot start with an underscore	You must change the tagname to comply with tagname requirements. If the tagname was assigned to a protocol, you must modify the protocol data mapping.
V6	User defined tagname '{0}' cannot contain "_Pt" in its name	You must change the tagname to comply with tagname requirements. If the tagname was assigned to a protocol, you must modify the protocol data mapping.
V7	Count for tagname exceeds maximum size.	You must reduce the count value to clear the error. If you need more data, you should create another Tagname Database item.
V8	Count must be greater than zero for tagname	A count of 0 is invalid.

Error Code	Error Message	Comments
V9	Tagnames are referencing the same Host Controller memory addresses(es).	<p>Only one tagname can reference a particular memory address in the host controller. You will need to change the address or the count. This error usually occurs because the value of the count causes one or more of the addresses in the block to overlap addresses referenced in another tagname. Also note:</p> <ul style="list-style-type: none"> • Type CP addresses cannot overlap Type C addresses, since both reference control relays, • Type XYP addresses cannot overlap Type XY addresses, since both reference discrete I/O registers, • Type V32 and VF cannot overlap Type V addresses, since they all reference V memory. • Type V32 and VF consume two V memory addresses of each count.
V10	Duplicate Tagname	Tagnames must be unique. Click on the Tagname heading in the Tagname Database panel to sort by tagnames.
V11	Invalid data type for tagname	The configured data type is not supported
V12	Missing tagname in Tagname Database	One or more entries do not have a tagname. You will need to add a tagname or delete the entry.
V13	Address must be greater than zero for tagname.	An address of 0 is invalid.
V14	Count for tagname must be 16 or less if using a packed data word (CP or XYP)	If you selected one of these data types, the maximum is 16, since the count represents the number of control relays or discrete I/O register points to pack into a 16 bit word. You must reduce the count or change data types.
V15	Total Cached Data exceeds maximum number of points allowed	You will need to delete some Tagname Database items or reduce the count for Tagname Database items.
V16	Transfer set has a total item count of n. This exceeds the maximum set size.	Transfer sets are created when you select data consistency. See APPENDIX D: DATA CONSISTENCY.
V17	Address/Count combination for tagname exceeds maximum V memory address.	<p>This error can occur if the address value exceeds the maximum or, if the count is greater than 1, the count value causes one or more addresses in the block to exceed the maximum.</p> <p>NOTE: The actual V memory address available in the Host controller will likely be less than the configuration maximum, since V memory is user configurable.</p>

Error Code	Error Message	Comments
V18	Address/Count combination for tagname exceeds maximum WXWY memory address	This error can occur if the address value exceeds the maximum or, if the count is greater than 1, the count value causes one or more addresses in the block to exceed the maximum. NOTE: The actual WXWY (word I/O memory) addresses available in the Host controller may be less than the configuration maximum, since the number of Word I/O memory addresses varies by controller model.
V19	Address/Count combination for tagname exceeds maximum XY memory.	This error can occur if the address value exceeds the maximum or, if the count is greater than 1, the count value causes one or more addresses in the block to exceed the maximum. NOTE: The actual XY (discrete I/O memory) addresses available in the Host controller may be less than the configuration maximum, since the number of Discrete I/O memory addresses varies by controller model
P1	CAMP requests are limited to 256 words. (Note that VF and V32 data types consume 2 words each).	Reduce the size of this request by eliminating mapping of some tagnames or reduce the Count for mapped tagnames. Use another request to access additional data.
P2	CAMP request exceeds maximum addressable memory limit.	Reduce the maximum V memory address by changing the starting address in the request. NOTE: The actual V memory address available in the Host controller will likely be less than the configuration maximum, since V memory is user configurable.
P3	Maximum connections exceeded. Only n connections of this are allowed	Reduce the number of connections.
P4	Cannot subscribe to the same IP Address more than once	Eliminate the duplicate connection.
P5	Please correct invalid data entry	Invalid entry is highlighted by a red border around the entry box.
P6	Missing or Empty Tagname	No tagname was selected to be mapped. You must select a tagname to associate the Host Controller data with this protocol.
P7	Invalid Modbus Block	The Modbus Server block data type is not supported. This error will occur only if the configuration has been hand edited.

Error Code	Error Message	Comments
P8	Modbus Server Block is overlapping Modbus address space with another block of this type	The Modbus block has been configured such that the same Modbus addresses are contained in more than one block. Change the Starting Modbus address of the block or change the size of the block. The size of the block is determined by the cumulative count of all Tagname Database items mapped to it.
P9	The address and number of items specified exceeds the Modbus maximum address of 65536	The starting Modbus address and the cumulative count of the Host controller Tagname Database items mapped to it determine the maximum address requested.
P10	Invalid Modbus Request	The Modbus request is not supported. See Modbus Function Code Table .
P11	The selected Modbus request is limited to n items	The maximum number of coils depends on the function code used. See Modbus Function Code Table .
P12	The selected Modbus request is limited to n registers. (Note that V32 and VF count as 2 Registers each)	The maximum number of registers depends on the function code used. See Modbus Function Code Table .
P13	Modbus request exceeds maximum addressable memory limit	Change the Modbus address and/or the Host Controller Data items mapped to the request.
P14	READ Requests are not valid when protocol is UDP-Multicast	Change request to a Write request
P15	Tagnames are using the same published Data ID.	Change the published Data ID to be unique.
P16	Published ID must be greater than zero.	Change the published Data ID to a unique non-zero number
P17	Subscribed ID must be greater than zero.	Change the published Data ID to a unique non-zero number
P18	No variables are mapped.	You must map at least one tagname to the object
P19	On Change activation is valid only for a WRITE request	Change the request to a Write request or use the Periodic or Triggered activation method.
P20	Only n requests are allowed per client connection	Delete requests until the number is less than or equal to the maximum allowed. Some devices allow you to make multiple connections to the same device. If so, you could make another connection and add requests to it.
P21	Mapping of _ECC_STATUS_WORD is currently disabled in settings	Enable mapping of this status word in the advanced tab of the Settings window or delete the mapping to this status word.
P22	Status and Trigger tagnames must reference a Tagname Database Item whose Count is 1.	Change the count of the mapped tagname(s) or create another tagname with a count of 1 for this purpose.

Error Code	Error Message	Comments
P23	Tagname was not found in the Tagname Database	This can happen when you map a tagname to a request or other object and subsequently delete or change the name of the tagname. You will need to re-map a tagname currently in the Tagname Database or add a new Tagname Database item and map it.

Configuration Program Error Messages

The following table describes the error messages that may be displayed when downloading a new configuration to the ECC1 module or when updating the module firmware. The error displayed depends on the operating system you are using.

Error Message	Cause of Error	Comments
System Error	This is a general error message indicating that the attempted activity failed for an unspecified reason.	See the comments for specific error messages.
Unable to connect to the remote server. The operation has timed out.	Incorrect IP address, no path exists between the PC and the module, or the file server is not available	Ensure the IP address you are using to connect matches that of the ECC1 module. Ensure that a connection between the module and the PC exists. Restart the module by cycling power, after ensuring that another PC is not logged into the module.
The remote server returned and error (421). Service not available, closing control connection.	Another user is already logged into the module. The ECC1 module is in the process of updating the flash memory or SD card with data from a previous download.	Wait until the other user disconnects and the ECC1 module completes updating flash or the SD card.
Server returned: 421 Max connections reached; cannot accept connection.		
The SD card in the ECC1's slot is write protected. Please check the switch on the SD Card.	The Write Protect switch on the SD card is in the Lock position. Dipswitch position 2 (firmware update method) is in the SD Card (closed) position.	Remove the SD card from the ECC1 SD card slot. If the switch is in the Lock position, set the switch to the non-lock position. Examine Dipswitch position 2. If it is in the Closed position, set it to the Open position.

APPENDIX B: IP ADDRESS INFORMATION

IP Address Nomenclature

IP Address

Every host interface on a TCP/IP network is identified by a unique IP address. This address is used to uniquely identify the host device, such as a workstation or communications module, and the network to which the host belongs.

Each IP address consists of 32 bits, divided into four 8 bit entities called *octets*. An IP address is expressed in *dotted notation*, with each octet expressed as its decimal equivalent. See the example below.

Notation	Octet 1	Octet 2	Octet 3	Octet 4
Binary	11000000	11011111	10110001	00000001
Decimal	192	223	177	1

Although an IP address is a single value, it contains two types of information: the *Network ID* and the *Host ID*. The Network ID identifies the IP network to which the host belongs. The Host ID identifies a specific IP host on that IP network. All IP hosts on a particular local area network must have the same network ID. Each IP host on a particular local area network must use a unique Host ID.

Address Classes

The Internet community originally defined network classes to accommodate networks of varying sizes. The network class can be discerned from the first octet of its IP address.

The following table summarizes the relationship between the first octet of a given address and its Network ID and Host ID fields. It also identifies the total number of Network IDs and Host IDs for each address class that participates in the Internet addressing scheme.

Class	First Octet Value*	Network ID	Host ID	Number of networks	Number of hosts per net
A	1-126	First Octet	Last 3 Octets	126	16,777,214
B	128-191	First 2 Octets	Last 2 Octets	16,384	65,534
C	192-223	First 3 Octets	Last Octet	2,097,151	254

* Address 127 is reserved for loopback testing and inter-process communication on the local computer; it is not a valid network address. Addresses 224 – 239 are used for Class D (IP multicast).

Subnet Mask

Used alone, the designation of network classes is very inflexible. For example, a Class A network assigns a large number of host devices to the same IP network; potentially reducing performance, limiting topology, and compromising network security. An additional entity, the Subnet Mask, provides means of dividing a large IP network into a collection of smaller networks called subnets.

The Subnet Mask is a collection of 32 bits that distinguish the network ID portion of the IP address from the host ID. Subnet Masks set bits that belong to the network ID to 1 bits that belong to the host ID to 0. Like the IP Address, the resulting 32-bit value is expressed in dotted decimal notation. See the example below.

Bits for Network Mask				Network Mask in Dotted Decimal
11111111	00000000	00000000	00000000	255.0.0.0 (default class A subnet mask)
11111111	11111111	00000000	00000000	255.255.0 (default class B subnet mask)
11111111	11111111	11110000	00000000	255.255.240.0 (subnetted class B network)
11111111	11111111	11111111	00000000	255.255.255.0 (default class C subnet mask)

For example: when the IP address is 172.54.177.97 and the subnet mask is 255.255.255.0, the Network ID is 172.54.177 and the Host ID is 97.

NOTE

The binary representation of a Network Mask must be a single continuous block 1's followed by a contiguous block of zeroes. When entering the Network Mask in dotted decimal notation, you must ensure that this requirement is maintained. For example, a network mask of 255.247.0.0 is not valid because the binary equivalent (11111111111101110000000000000000) violates this rule.

The Network Mask must allow at least two bits of host address. In addition, a network mask which causes the derived host ID to be 0 or a broadcast address (all Host ID bits set to 1) should not be used.

Using the Subnet Mask

For Class A, B, and C IP addresses, the IP Host uses the Subnet Mask to determine where to send an IP message. After deriving the Network ID and Host ID portion of the IP Address using the Subnet Mask, the IP Host compares the Network ID of the destination IP address with the Network ID of the Host IP address. If the Network IDs are the same, the message is sent to another Host on the local network. If the Network IDs are different, the message is sent to an IP Gateway, for routing to another network, if possible.

When you are configuring the IP address of devices that must communicate on a local network, you must ensure that:

- The Subnet Mask of all devices are the same,
- The Network ID of all hosts are the same,
- The Host ID of each host is different.

If you are using subnet masks that are aligned with the IP address octets, this can easily be done by examining the dotted decimal values. The octets of the IP address where the corresponding octet of the subnet mask is 255 belong to the Network ID and the octets of the IP address where the corresponding octet of the subnet mask is 0 belong to the Host ID.

For example, where the IP address is 127.18.40.3 with a subnet mask of 255.255.0.0, the Network ID is 127.18 and the Host ID is 40.3.

IP Address	127	18	40	3
Subnet Mask	255	255	0	0
Network ID	127	18		
Host ID			40	3

However, if you are using a subnet mask that does not align with the octet boundaries, this is more difficult. You will need to perform a bitwise “and” calculation to arrive at the Network address. See the following illustration.

Assuming an IP address of 127.18.40.3 and a Subnet Mask of 255.255.240.0, the following table illustrates the bitwise “and” operation. In essence, wherever the subnet mask bit is one, the corresponding IP address bit is part of the Network ID.

Item	Dotted Decimal	Binary Equivalent			
		1 st Octet	2 nd Octet	3 rd Octet	4 th Octet
IP Address	127.18.40.3	01111111	00010010	00101000	00000011
Subnet Mask	255.255.240.0	11111111	11111111	11110000	00000000
Derived Network Address	127.18.32.0	01111111	00010010	00100000	00000000

An easier way to determine this is to compare only non-aligned subnet mask octet with the corresponding octet of the IP address. For example, since the subnet mask of the first two octets is 255.255, the first two octets of the Network ID are the same as the dotted decimal values (127.18) of the IP address. However, since the third octet of the subnet mask is not 255 or 0, you must perform a bitwise calculation using the third octet of the IP address and network mask.

This can be accomplished by using the windows calculator (Programmer view). Using this example, you would enter the value of the third octet (40), click on the “and” button, enter the subnet mask (240), and then click on the “=” button. The result, in this case, is 32. Thus the Network address is 127.18.32.0.

Selecting an IP Address

In order to perform its functions, the 2500P-ECC1 requires a fixed IP address. If you are connecting to an existing network, you should obtain an unused static IP address and the network subnet mask from the network administrator.

If you are establishing your own IP addresses, you should select IP addresses from a block of ‘private’ addresses established by the Internet Assigned Numbers Authority (IANA). The private address blocks are:

- 10.0.0.0 through 10.255.255.255 (Class A)
- 172.16.0.0 through 172.31.255.255 (Class B)
- 192.168.0.0 through 192.168.255.255 (Class C)

These addresses will not be forwarded by the Internet backbone routers; therefore, you are free to use any address in this group as long as it does not conflict with the usage by your local organization.

Selecting a Multicast Address

The address range of 239.0.0.0 – 239.255.255.255 has been designated as an administratively scoped Multicast address space (RFC 2365). Addresses in this range are designated for use by private multicast domains. They do not conflict with other multicast address spaces that are explicitly assigned by Internet Assigned Numbers Authority (IANA). Within this range, addresses 239.255.0.0 – 239.255.255.255 is designated for the IPV4 multicast local scope.

If you are choosing a multicast address for a new factory floor application, you should choose a multicast address in the IPV4 local scope range (239.255.0.0 – 239.255.255.255) unless you have a specific reason to do otherwise. When choosing a multicast address, you should verify there is no conflict with other multicast addresses being used locally.

NOTE:

The 2500P-ECC1 implementation of UDP Multicast requires that all participating hosts be on the same Ethernet local area network. Routing of multicast packets is not supported.

In case you are using the 2500P-ECC1 module in an existing multicast application that uses a multicast address outside of the administratively scoped address space, the configuration program allows you to enter the complete range of assignable multicast addresses (224.0.0.1 – 239.255.255.255).

For a current list of IANA assigned multicast addresses, see www.iana.org/assignments/multicast-addresses/.

APPENDIX C: CAMP SERVER SUPPORT

Overview

This appendix specifies the CAMP Messages, task code, and controller data types supported by the CAMP server. The use of ECC1 Status Word (STW2048) is also described.

Task Code Support

The CAMP Server supports the following task codes.

Task Code	Description	Comments
01	Read User Word Random	Reads most controller data types using word codes
02	Write User Word Random	Writes most controller data types using word codes
05	Read Event Drum Preset Count	Requires firmware version 2.07 or greater. Reads the drum preset count value for a specified drum and drum step.
07	Read Drum Timer Current Count and Preset/ Current Step	Requires firmware version 2.07 or greater. Reads the Preset step, the current step and the count to the current step for a specified drum. See note for TC09.
09	Write Drum Timer Preset Step	Requires firmware version 2.07 or greater. Writes the preset step for a specified drum. NOTE: The Preset Step value will be immediately updated in the cache. The cached current count and current step will be updated based on their Cache Refresh Interval. As a result, sending TC07 immediately after writing the Preset Step may return inconsistent values.
0C	Read Timer /Counter Preset/Current	Requires firmware version 2.07 or greater. Reads the preset and current counter value of a specified timer/counter. See note for TC0E.
0E	Write Timer/Counter Preset	Requires firmware version 2.07 or greater. Writes the Preset value of a specified Timer/Counter. NOTE: The Preset value will be immediately updated in the cache. The Timer/Counter current value will be updated in the cache based on its Cache Refresh Interval. As a result, sending TC0C immediately after writing the Preset may return inconsistent values.
11	Read Discrete/Force Status	Requires firmware version 2.05 or greater. Allows the status and force state of a discrete I/O point or control relay to be read. NOTE: The Module Present and Input/Output bits are not updated.
12	Read Discrete Status (Fast)	Requires firmware version 2.05 or greater. Allows the status of a discrete I/O point or control relay to be read.

Task Code	Description	Comments
14	Write Discrete Force/Status	Allows the status of X, Y, C to be changed (OFF or ON). Does not allow forcing a point.
19	Read Word Force	Reads the force status for a specified word (WX/WY). Note: Only the force status bit of the response is updated by the module. Attempts to access a discrete data type (X, Y, C) will return task code error 02 – (Invalid Address).
50	Read User Word Area Block	Reads a contiguous set of controller addresses
51	Write User Word Area Block	Writes a contiguous set of controller addresses
52	Fill User Word Area Block	Fills a contiguous set of controller addresses with a designated value
59	Write Discrete I/O Status or Force via Data Element Type	Does not support writing forces (TT Type 0B, 0C, 0D). Attempts to access these data types will return task code error 11 (Invalid Data sent with the command).
5A	Write Block	Writes a block of loop or alarm data
60	Write Loop	Writes specified loop parameters to a designated loop
64	Write Alarm	Writes specified alarm parameters to a designated analog alarm.
6B	Read Discrete I/O Status by TT type	Reads discrete inputs and outputs, control relays, discrete forces, control relay forces, and word I/O forces
76	Read Loop	Reads specified loop parameters from a designated loop. NOTE: LOOP SEARCHES (AL=00) ARE NOT SUPPORTED.
79	Read Alarm	Reads specified alarm parameters from a designated analog alarm. NOTE: ALARM SEARCHES (AL=00) ARE NOT SUPPORTED.
7E	Read Random Word	Reads random data memory addresses
7F	Read Block Word	Writes a block of data to contiguous addresses of a specified data type.
9D	Read Random Block via TT type	Reads multiple blocks of controller data
9E	Write Random Block via TT type	Writes multiple blocks of controller data

Because the 2500P-ECC1 is not designed as a programming interface, the ECC1 CAMP Server does not support many of the task codes used by programming software. Programming software should be connected to the 2500 Series local port (Ethernet or Serial).

CAMP Message Support

The CAMP protocol is used by many CTI products to enable reading and writing large blocks of PLC data as well as to provide an efficient method for transporting and processing task codes. The CAMP Server supports the following CAMP message types:

CAMP Message Type	Description
Read Data	Reads up to 256 words of a given controller data element type
Write Data	Writes up to 256 words of a given controller data element type

Unacknowledged Write Data	Used by UDP Multicast
Memory Exchange	Writes up to 254 words of a given controller data type and reads up to 256 words of the same data element type in one transaction
Packed Task Code	Sends up to 14 task codes (NITP format) in a single message

Data Element Support

The CAMP server provides access to common data element types, loop data element types, and alarm data element types using data element type (TT) codes as indicated in the tables below.

Common Data Element Types

TT	Mnemonic	Description	Comments
01	V	Variable Memory	Integer
02	K	Constant Memory	
03	X	Discrete Input IR	Byte (FF=True, 00=False)
04	Y	Discrete Output IR	
05	C	Control Relay	
06	XP	Discrete IR Packed	
07	YP	Discrete IR Packed	Byte <i>See Note 1</i>
08	CP	Control Relay Packed	
09	WX	Word Input	Integer
0A	WY	Word Output	
0B	DF	Discrete Force	Read-Only Not accessible using CAMP Read Data Command but can be accessed via task code. <i>See Note 4 for additional information.</i>
0C	CF	Control Relay Force	
0D	WF	Word Force	
0E	TCP	Timer Counter Preset	
0F	TCC	Timer Counter Current	
10	DSP	Drum Step Preset	
11	DSC	Drum Step Current	
12	DCP	Drum Count Preset	Word. <i>See Note 2</i>
1A	STW	System Status Words	16 bit Word
1B	DCC	Drum Count Current	Read Only. 32 bit structure
1C	V.	Variable memory	Real. <i>See note 3</i>
1D	K.	Constant Memory	

Note 1: For packed Boolean type, (X, C, or Y) requested values are returned packed 8 bits to a byte. The first Boolean shall be returned in the LSB of the first byte with each successive Boolean occupying the next highest position. Additional Boolean values beyond the first byte will be returned in successive bytes following the same pattern as the first byte. Unused bits of the last byte shall be set to 0.

Note 2: The offset field for DCP is a 24 bit field coded as: Bit 0 - 3: Step -1, Bit 4 – 23: Drum-1 where bit 0 is the least significant bit.

Note 3: When addressing V or K as real, the value occupies two V memory positions. The offset must indicate the actual V memory starting location.

Note 4:

This data type can be read using task codes only. The task codes can be included in a CAMP Packed task code message.

When accessing Data Type 0B or 0C using task code 6B, the response from the ECC1 will be the same as the response when reading directly from the 2500 Series® controller or Series 505®. The force status for the designated discrete or control relay addresses are returned packed into one or more bytes, starting at the least significant bit. A force is indicated when the corresponding bit is set to 1. Unused bits are set to 0. See the task code specification for additional information.

Using task code 9D, you can retrieve the force state and the discrete status or word value at the same time. When accessing Data Type 0B, 0C, or 0D using task code 9D, the response from the ECC1 will be **different** from the response returned when reading directly from the 2500 Series® controller or Series 505 in the following way: the bits for Module Present and Input/Output are not returned.

Data Type 0B and 0C, when accessed directly from a Series 505® or 2500 Series® controller using task code 9D, return a byte for each designated discrete or control relay address. The binary format for the byte is shown below.



Where:

0 = Unused bits (set to 0)

A = Module Present (1 = Present)

B = Forced State (1 = Forced)

C = Input/Output (1 = Output)

D = Discrete/Word (Always 0 = Discrete)

E = Status (1 = On)

When Data Type 0B or 0C is accessed via the ECC1, the Bit A (Module Present) and Bit C (Input/Output) are not set. The returned byte is shown below, where B and E can be either 1 or 0 and D is always 0.



Data Type 0D, when accessed directly from a Series 505® or 2500 Series® controller using task code 9D, returns a status byte shown below plus two bytes of data, representing the word value.



Where:

A = Forced State (1 = Forced)

B = Module Present (1 = Present)

C = Discrete/Word (Always 1 = Word)

D = Input/Output (1 = Output)

When Data Type 0D is accessed via the ECC1, a status byte along with two bytes of data representing the word value is returned. However, bits B and D are not set by the ECC1. The returned status byte is shown below, where bit A can be either 0 or 1 and bit C is always 1.



Loop Data Element Types

TT	Mnemonic	Description	Comments
1E	LS	Loop Status	16 bits , Read Only
1F	LM	Loop Mode	16 bits
20	LKC	Loop Gain	Real
21	LTI	Reset Time (min)	Real
22	LTD	Rate Time (min)	Real
23	LHA	High Alarm Limit	Real
24	LLA	Low Alarm Limit	Real
25	LPV	Loop Process Variable (PV)	Real
26	LPVH	PV High Limit	Real
27	LPVL	PV Low Limit	Real
28	LODA	Orange Deviation Alarm Limit	Real
29	LYDA	Yellow Deviation Alarm Limit	Real
2A	LTS	Sample Rate (seconds)	Real
2B	LSP	Setpoint	Real
2C	LMN	Output (percent)	Real
2D	LVF	Loop V Flags	16 bits
2E	LCF	Loop C Flags	32 bits
2F	LRSF	Loop Ramp Soak Flags	16 bit
30	LERR	Loop Error	Real , Read Only
31	LMX	Loop Bias	Real
32	LHHA	Alarm High High Limit	Real
33	LLLA	Alarm Low Low Limit	Real
34	LRCA	Rate of Change Alarm Limit (Eng Units / Min)	Real
35	LSPH	Setpoint High Limit	Real
36	LSPL	Setpoint Low Limit	Real
37	LADB	Alarm Deadband	Real
38	LHA	Raw High Alarm Limit	Integer
39	LLA	Raw Low Alarm Limit	Integer
3A	LPV	Raw Process Variable	Integer
3B	LODA	Raw Orange Deviation Limit	Integer
3C	LYDA	Raw Yellow Deviation Limit	Integer
3D	LMN	Raw Output	Integer
3E	LSP	Raw Setpoint	Integer
3F	LERR	Raw Error	Integer, Read Only
40	LHHA	Raw High-High Alarm Limit	Integer
41	LLLA	Raw Low-Low Alarm Limit	Integer
42	LADB	Raw Alarm Deadband	Integer
48	LMX	Raw Bias	Integer
49	LSPL	Raw Setpoint Low Limit	Integer
4A	LSPH	Raw Setpoint High Limit	Integer
4B	LCFH	C Flag Most Significant Word	Integer

TT	Mnemonic	Description	Comments
4C	LCFL	C Flag Least Significant Word	integer
4D	LKD	Loop Derivative Gain Coefficient	Real
4E	LRSN	Ramp Soak Step Number	Integer
4F	LACK	Alarm Acknowledge Flags	Integer

Alarm Data Element Types

TT	Mnemonic	Description	Comments
50	AHA	High Alarm Limit	Real
51	ALA	Low Alarm Limit	Real
52	APV	Alarm Process Variable (PV)	Real
53	APVH	Alarm PV High Limit	Real
54	APVL	Alarm PV Low Limit	Real
55	AODA	Orange Deviation Alarm Limit	Real
56	AYDA	Yellow Deviation Alarm Limit	Real
57	ATS	Sample Rate in seconds	Real
58	ASP	Setpoint	Real
59	AVF	Alarm V Flag	16 bits
5A	ACF	Alarm C Flag	32 bits
5B	AERR	Alarm Error	Real , Read Only
5C	AHHA	Alarm High High Limit	Real
5D	ALLA	Alarm Low Low Limit	Real
5E	ARCA	Rate of Change Alarm Limit (Eng Units / Minute)	Real
5F	ASPH	Setpoint High Limit	Real
60	ASPL	Setpoint Low Limit	Real
61	AADB	Alarm Deadband	Real
62	AHA	Raw High Alarm Limit	Integer
63	ALA	Raw Low Alarm Limit	Integer
64	APV	Raw Process Variable	Integer
65	AODA	Raw Orange Dev. Alarm Limit	Integer
66	AYDA	Raw Yellow Dev. Alarm Limit	Integer
67	ASP	Raw Setpoint	Integer
68	AADB	Raw Alarm Deadband	Integer
69	AERR	Raw Error	Integer, Read Only
6A	AHHA	Raw High-High Alarm Limit	Integer
6B	ALLA	Raw Low-Low Alarm Limit	Integer
6F	ASPL	Raw Setpoint Low Limit	Integer
70	ASPH	Raw Setpoint High Limit	Integer
71	ACFH	Alarm C Flag Most Significant Word	Integer
72	ACFL	Alarm C Flag Least Significant Word	Integer
73	AACK	Alarm Acknowledge Flags	Integer

Status Word 2048

Status Word 2048 (STW2048) enables the CAMP Server to access the ECC1 Status Word. The ECC1 Status Word provides contains bits reflecting the operational state of the ECC1 module and its Host Controller.

Unlike other Status Word addresses, STW2048 is read from the ECC1 module, not the Host Controller. The format of STW2048 is illustrated below. To comply with the 2500 Series conventions, bit number 1 is shown as the most significant bit. Unused bits, shown as 0 in the table below, are reserved for future use.

Bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	0	0	0	0	0	0	0	0	0	0	A	B	C	D	0	0

A	Invalid Host Controller Memory Address	0 = All items in the Tag Database reference valid Host Controller memory addresses. 1 = Tag Database contains one or more items that reference memory addresses exceed the address range of the Host Controller.
B	Priority Cache Status	0 = Current – All members of the cache are being updated within the designated cache refresh interval. 1 = Stale – some members of the cache are not being updated within the designated cache refresh interval
C	Normal Cache Status	0 = Current – All members of the cache are being updated within the designated cache refresh interval. 1 = Stale – some members of the cache are not being updated within the designated cache refresh interval
D	Host Controller Mode	0 = Host Controller is in Run Mode 1 = Host Controller is in Program Mode or status is unknown

When STW2048 is evaluated as a 16 bit unsigned integer, a value of 0 indicates that all status attributes are normal.

NOTE:

Status Word 2048 cannot be accessed when the Host Controller is not connected or is in fatal error mode, since both states block servicing of CAMP requests.. CTI is planning on addressing this issue in a future release.

APPENDIX D: DATA CONSISTENCY

Overview

When caching data items, the ECC1 module updates the cache based at the user specified Cache Refresh Interval (CRI). If the ECC1 module attempted update as many cache items as possible in a single scan, it would result in some scans consuming much more time than others processing cache update requests. Thus the Host Controller peak scan time would be larger than necessary.

To minimize the peak scan time of the Host Controller, cache updates are distributed over time. As a result, various data cache members may be updated from data obtained in different Host Controller scans. Since most data communications applications do not require that data items in the cache be obtained in the same scan, this distribution technique works well in the majority of cases. However, there are some applications that do require that a selected set of data be updated in the same Host Controller scan. For these applications, you can select data consistency.

How Data Consistency Works

Client applications read data from devices on the network and write the returned data to the Host Controller. When writing data to devices on the network, they read data from the ECC1 data cache. Consequently, for client applications, data consistency applies to write requests only. Data consistency is selected by checking the Consistency box in the client request. When the box is checked, all of the Tagname Database items that are mapped to the request are assigned a transfer set number. The ECC1 attempts to update all items in a particular transfer set in the same scan.

For transfer sets whose members are mapped to write requests using the periodic, enable trigger, or on-change activation methods, the data cache update frequency is determined by shortest cache refresh interval (CRI) of all the Tagname Database items that are members of the set. For example, if one Tagname Database item in a transfer set is using a Priority update whose CRI is 500ms and the rest of the transfer set members are using a Normal update with a CRI of 1000ms, all items will be updated every 500ms.

For transfer sets whose members are mapped to write requests using the one-shot trigger activation method, the data cache is updated immediately after the trigger transitions from low to high.

If any of the members of a transfer set is mapped to another write request using consistency along with one or more data items that were not in the original transfer set, the new items are added to the transfer set. For example, if Tagnames A and B were mapped to one write request using consistency and Tagnames A and C were used in another request using consistency, the transfer set would include All Host Controller data associated with Tagnames A, B, and C.

The maximum number of Host Controller memory addresses that can be in a transfer set is 256. If your configuration exceeds the maximum, the configuration program will report an error when you attempt to compile and send or save the configuration.

Guidelines for Using Data Consistency

For best results, the following guidelines should be considered when using data consistency.

- Select consistency only when needed. Client write requests using the one-shot trigger method will likely benefit from consistency. For other activation methods, the need will depend on the application. Excessive use of consistency may increase the Host Controller peak scan time and could prevent timely cache updates in some circumstances.
- Minimize the number of Host Controller data items that are mapped to a write request using consistency. Attempt to include only those items that must be updated from data obtained in the same scan.
- When possible, map a Tagname Database item representing a block of contiguous Host Controller memory addresses rather than mapping many Tagname Database items with non-contiguous Host Controller memory addresses. It is more efficient to read a block of controller memory than to read scattered memory locations.
- If possible, avoid combining Tagnames used in a write request using consistency with other Tagnames used in another request. This increases the size of the transfer set, as explained in the section above.

APPENDIX E: DATA CACHE OPERATION

Overview

The 2500P-ECC1 module maintains a cache of Host Controller data items from which data is supplied to the ECC1 protocols. The data caching method used by the ECC1 module substantially reduces the time to respond to protocol events that read Host Controller data while providing consistent data quality.

Cache Access

Protocol events that result in reading data from the Host Controller are serviced by supplying the corresponding data from the cache. The events include:

- Receipt of a client request to read Host Controller data by an ECC1 server protocol,
- Activation of a request to write Host Controller data to a network device by an ECC1 client,
- Publishing of a Host Controller data item by the ECC1 Network Data Exchange Publisher.

Protocol events that write data to the Host Controller do not use the data cache. Instead, the write request is *immediately* transferred to the Host Controller to ensure timely update. After the data is successfully written to the Host Controller, the corresponding data cache item is updated. Events that write data to the Host Controller include:

- Receipt of a client request to write Host Controller data by an ECC1 server protocol,
- Receipt of a reply to a previous request to read data from a network device by an ECC1 client,
- Receipt of a subscribed item from a published by a Network Data Exchange client.

Cache Membership

There are two methods used to manage cache membership. The method used depends on the ECC1 protocol being used.

Cache membership for Host Controller data items accessed by the CAMP Server but not contained in the Tagname Database is dynamically managed. A data item is added to the data cache after it is first accessed. It remains in the cache so long as a request to read the data item arrives within the item Time-to-Live period (approximately 60 seconds). If no request is received within the item Time-to-Live period, the item is removed from the cache.

Cache membership for Host Controller data items accessed by the other ECC1 protocols is statically determined based on the ECC1 configuration. Data items contained in the Tagname Database are added to the cache under either of the following conditions:

- The ECC1 configuration designates that they will be accessed by one of the ECC1 protocols,
- The Tagname Database item referencing the data item is assigned to the high priority Cache Refresh Interval (CRI)

NOTE:

In order to be added to the cache, the data element memory address in the Host Controller corresponding to the data item must be accessible. The controller memory configuration must contain the address.

Cache Update

Each data cache member that is flagged for read access is periodically updated with new data from the Host Controller based on the Cache Refresh Interval (CRI) assigned to it. A cache member is flagged for read access only if the ECC1 configuration designates that the data item will be read by one of the ECC1 protocols or the CAMP server processes a client request to read the data item. Data cache members that are not flagged for read access are not periodically updated.

There are two user configurable Cache Refresh Interval categories: Normal and High Priority. The refresh interval values for the Normal and High Priority CRI categories are set in the configuration program (see Section 5.3.2). Cache members flagged for read access are automatically updated using the Normal CRI unless they are explicitly assigned to the High Priority CRI category by checking the High Priority box for corresponding Tagname Database item.

A data item whose tagname is assigned to a one-shot trigger will automatically be updated as fast as possible, regardless of the user specified CRI. If the consistency option is selected for client write request activated by the one-shot trigger, the data associated with the write request will be read from the Host Controller immediately before the request is transmitted to the network device. See APPENDIX D: DATA CONSISTENCY.

You should note that you can create configurations where it is impossible to achieve the specified CRI. See Section 4.3.3 for more information.

APPENDIX F: ALIAS IP FEATURE

Overview

The term “Alias IP Address” refers to associating multiple IP addresses to an Ethernet network interface, represented by a MAC (Media Access Control) address. This capability enables the ECC1 Ethernet network interface to communicate on two TCP/IP subnets.

The Alias IP Address feature allows you to assign two IP addresses (including associated subnet masks) to the Ethernet interface – a primary IP address and an Alias IP Address. Once this has been accomplished, server applications, such as the CAMP server and Open Modbus server, will accept requests from a local IP host with an IP address on either the IP subnet associated with the primary IP address or the IP subnet associated with the Alias IP address. In addition, Client applications, such as CAMP client and Open Modbus client, will be able to send requests to a local IP host with an IP address on either the IP subnet represented by the Primary IP address or the IP subnet represented by the Alias IP address.

The 2500P-ECC1 module contains an embedded three port switch that provides two external Ethernet ports. (Port 3 is connected to the ECC1 processor). In a typical configuration, one port connects directly to the host PLC and the other to the Ethernet network. Alternatively, each external port can connect to a different Ethernet LAN (or VLAN). The Port Isolation feature, which prevents frames from being forwarded between the external ports, can be used to block broadcast traffic from propagating from one network to the other.

It is important to understand that the Alias IP Address feature is independent of the embedded switch functionality. IP hosts that are members of different IP subnets can exist on the same Ethernet network as well as different Ethernet networks. It is also important to understand that all enabled ECC1 servers are accessible by clients on either the primary IP subnet or the Alias IP subnet, regardless of the port to which they are connected. Requests are directed to a particular server based on the IP port number. Similarly, any enabled ECC1 client can access IP hosts on either IP subnet, regardless of the port to which they are connected. For some applications, port isolation is neither required nor desired.

Use Cases

The following use cases illustrate how the Alias IP Feature can be used.

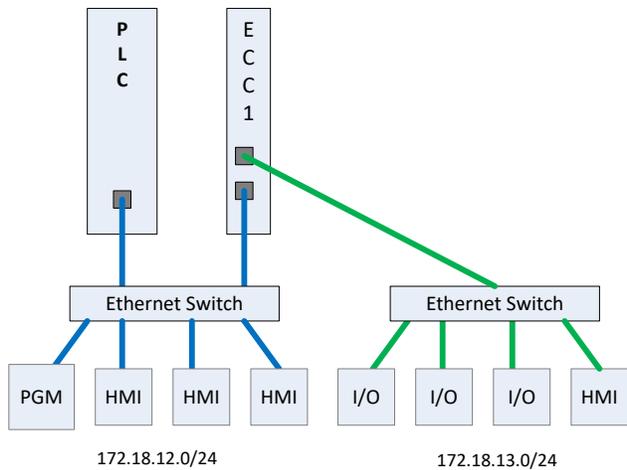
Communicating on Two Ethernet Networks.

In this case, HMI clients on the Ethernet data network must communicate with the CAMP server server on the 2500P-ECC1 module (hereafter referenced as ECC1) to access data on a CTI 2500 Series PLC. This network is also being used to program the PLC. In addition, a MODBUS client application must update MODBUS I/O modules on a separate Ethernet network. Devices on this network are required to use IP addresses that are not on the same IP subnet as the devices on the data network. To prevent broadcast traffic from the data

network from degrading the I/O network, forwarding of packets between the port connected to the data network and the port connected to the I/O network must be prevented.

The Ethernet controller has been assigned two IP addresses:

1. The primary IP address is a member of the 172.18.12.0/24 subnet (for example, an IP address of 172.18.12.10 with a subnet mask of 255.255.255.0).
2. The Alias IP address is a member of the IP Subnet 172.18.13.0/24, (for example, an IP address of 172.18.13.10 with a subnet mask of 255.255.255.0).
3. Port 2 of the ECC1 Module is connected to the data communications network containing the PLC, HMI, and programming (PGM) devices. Each IP host on this network is assigned an IP address that is member of the 172.18.12.0/24 subnet.
4. Port 1 of the ECC1 module is connected to the network containing the I/O devices. Each IP host on this network is assigned an IP address that is member of the 172.18.13.0/24 subnet.



Traffic Flow

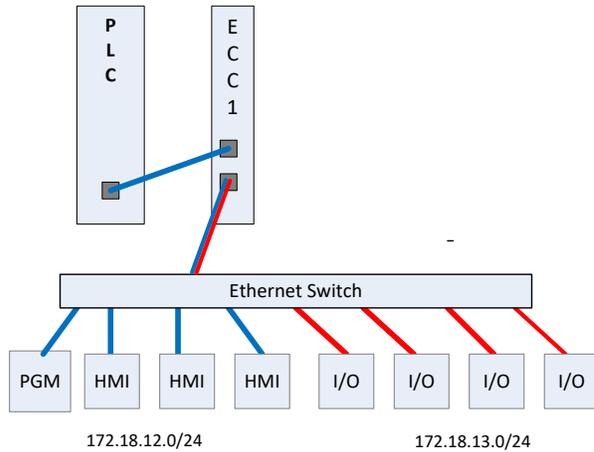
HMI stations, whose IP addresses are a member of IP subnet 172.18.12.0/24, communicate with the ECC1 CAMP server via Ethernet Port 2 using the ECC1 primary IP Address. The programming workstation (PGM) and the ECC1 communicate with the PLC using the PLC IP address, which is also a member of this IP subnet.

The ECC1 Open Modbus client communicates via Ethernet Port 1 with I/O devices whose IP addresses are members of IP subnet 172.18.13.0/24 using the alias IP address. HMI clients attached to this network can also communicate with the ECC1 CAMP server. The ECC1 TCP/IP stack will route the request to the correct server based on the IP port number.

Once the embedded Ethernet switch learns which port is a path to a device with a particular IP address, unicast packets are transmitted only via that port. Broadcast packets from the ECC1 module will be forwarded to both port 1 and port 2. Broadcast packets from devices on either port will be forwarded to the ECC1 module. Since the requirements state that broadcast packets cannot be forwarded between the Ethernet networks connected to Ethernet port 1 and port 2, the PORT ISOLATION feature must be enabled.

Communicating on an Ethernet Network with Multiple IP Subnets

Some applications require the capability to communicate with devices that are on a different subnet IP subnet than the PLC with which the module is communicating, but have no need for separate, isolated Ethernet networks. The inability to assign IP addresses that are on the same IP as the PLC subnet could be due to a lack of available IP addresses or network standards imposed by the IT department. In this case, traffic for both IP subnets is transmitted on the same Ethernet local area network. As an added benefit, this configuration allows the module to provide broadcast storm protection for the PLC, while allowing programming software to access the PLC. See diagram below.



Traffic Flow

All packets between the Programming Workstation (PGM), HMI, and I/O are transmitted and received by the ECC1 via port 2. HMI and I/O traffic is serviced by the ECC1. Requests from the programming workstation and replies from the PLC are forwarded between port 1 and port 2 of the ECC1 by the embedded switch. Port 1 also transfers data requests and replies between the ECC1 and the PLC. Because forwarding between Ethernet ports on the ECC1 module is required, the PORT ISOLATION feature is not enabled.

APPENDIX G: PRODUCT SPECIFICATIONS

Hardware Specifications

Module Size: Single Wide I/O

Backplane Power Consumption: 5 watts @ 5 VDC

Operating Temperature: 0° to 60° C (32° to 185° F)

Storage Temperature: -40° to 85° C (-40° to 185° F)

Humidity: 0% to 95%, non-condensing

LIMITED PRODUCT WARRANTY

Warranty. Control Technology Inc. ("CTI") warrants that this CTI Industrial Product (the "Product") shall be free from defects in material and workmanship for a period of one (1) year from the date of purchase from CTI or from an authorized CTI Industrial Distributor, as the case may be. Repaired or replacement CTI products provided under this warranty are similarly warranted for a period of 6 months from the date of shipment to the customer or the remainder of the original warranty term, whichever is longer. This Product and any repaired or replacement products will be manufactured from new and/or serviceable used parts which are equal to new in the Product. This warranty is limited to the initial purchaser of the Product from CTI or from an authorized CTI Industrial Distributor and may not be transferred or assigned.

2. Remedies. Remedies under this warranty shall be limited, at CTI's option, to the replacement or repair of this Product, or the parts thereof, only after shipment by the customer at the customer's expense to a designated CTI service location along with proof of purchase date and an associated serial number. Repair parts and replacement products furnished under this warranty will be on an exchange basis and all exchanged parts or products become the property of CTI. Should any product or part returned to CTI hereunder be found by CTI to be without defect, CTI will return such product or part to the customer. The foregoing will be the exclusive remedies for any breach of warranty or breach of contract arising therefrom.

3. General. This warranty is only available if (a) the customer provides CTI with written notice of a warranty claim within the warranty period set forth above in Section 1 and (b) CTI's examination of the Product or the parts thereof discloses that any alleged defect has not been caused by a failure to provide a suitable environment as specified in the CTI Standard Environmental Specification and applicable Product specifications, or damage caused by accident, disaster, acts of God, neglect, abuse, misuse, transportation, alterations, attachments, accessories, supplies, non-CTI parts, non-CTI repairs or activities, or to any damage whose proximate cause was utilities or utility-like services, or faulty installation or maintenance done by someone other than CTI.

4. Product Improvement. CTI reserves the right to make changes to the Product in order to improve reliability, function or design in the pursuit of providing the best possible products.

5. Exclusive Warranty. THE WARRANTIES SET FORTH HEREIN ARE CUSTOMER'S EXCLUSIVE WARRANTIES. CTI HEREBY DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED. WITHOUT LIMITING THE FOREGOING, CTI SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, COURSE OF DEALING, AND USAGE OF TRADE.

6. Disclaimer and Limitation of Liability. TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAW, CTI WILL NOT BE LIABLE FOR ANY BUSINESS INTERRUPTION OR LOSS OF PROFIT, REVENUE, MATERIALS, ANTICIPATED SAVINGS, DATA, CONTRACT, GOODWILL OR THE LIKE (WHETHER DIRECT OR INDIRECT IN NATURE) OR FOR ANY OTHER FORM OF INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND. CTI'S MAXIMUM CUMULATIVE LIABILITY RELATIVE TO ALL OTHER CLAIMS AND LIABILITIES, INCLUDING OBLIGATIONS UNDER ANY INDEMNITY, WHETHER OR NOT INSURED, WILL NOT EXCEED THE COST OF THE PRODUCT(S) GIVING RISE TO THE CLAIM OR LIABILITY. CTI DISCLAIMS ALL LIABILITY RELATIVE TO GRATUITOUS INFORMATION OR ASSISTANCE PROVIDED BY, BUT NOT REQUIRED OF CTI HEREUNDER. ANY ACTION AGAINST CTI MUST BE

BROUGHT WITHIN EIGHTEEN (18) MONTHS AFTER THE CAUSE OF ACTION ACCRUES. THESE DISCLAIMERS AND LIMITATIONS OF LIABILITY WILL APPLY REGARDLESS OF ANY OTHER CONTRARY PROVISION HEREOF AND REGARDLESS OF THE FORM OF ACTION, WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY) OR OTHERWISE, AND FURTHER WILL EXTEND TO THE BENEFIT OF CTI'S VENDORS, APPOINTED DISTRIBUTORS AND OTHER AUTHORIZED RESELLERS AS THIRD-PARTY BENEFICIARIES. EACH PROVISION HEREOF WHICH PROVIDES FOR A LIMITATION OF LIABILITY, DISCLAIMER OF WARRANTY OR CONDITION OR EXCLUSION OF DAMAGES IS SEVERABLE AND INDEPENDENT OF ANY OTHER PROVISION AND IS TO BE ENFORCED AS SUCH.

7. Adequate Remedy. The customer is limited to the remedies specified herein and shall have no others for a nonconformity in the Product. The customer agrees that these remedies provide the customer with a minimum adequate remedy and are its exclusive remedies, whether based on contract, warranty, tort (including negligence), strict liability, indemnity, or any other legal theory, and whether arising out of warranties, representations, instructions, installations, or non-conformities from any cause. The customer further acknowledges that the purchase price of the Product reflects these warranty terms and remedies.

8. Force Majeure. CTI will not be liable for any loss, damage or delay arising out of its failure (or that of its subcontractors) to perform hereunder due to causes beyond its reasonable control, including without limitation, acts of God, acts or omissions of the customer, acts of civil or military authority, fires, strikes, floods, epidemics, quarantine restrictions, war, riots, acts of terrorism, delays in transportation, or transportation embargoes. In the event of such delay, CTI's performance date(s) will be extended for such length of time as may be reasonably necessary to compensate for the delay.

9. Governing Law. The laws of the State of Tennessee shall govern the validity, interpretation and enforcement of this warranty, without regard to its conflicts of law principles. The application of the United Nations Convention on Contracts for the International Sale of Goods shall be excluded.

REPAIR POLICY

In the event that the Product should fail during or after the warranty period, a Return Material Authorization (RMA) number can be requested orally or in writing from CTI main offices. Whether this equipment is in or out of warranty, providing a Purchase Order number to CTI when requesting the RMA number will aid in expediting the repair process. The RMA number that is issued and your Purchase Order number should be referenced on the returning equipment's shipping documentation. Additionally, if the product is under warranty, proof of purchase date and serial number must accompany the returned equipment. The current repair and/or exchange rates can be obtained by contacting CTI's main office at 1-800-537-8398 or go to www.controltechnology.com/support/repairs/.

When returning any module to CTI, follow proper static control precautions. Keep the module away from polyethylene products, polystyrene products and all other static producing materials. Packing the module in its original conductive bag is the preferred way to control static problems during shipment. Failure to observe static control precautions may void the warranty.