

# PACSystems™ RX3i and RSTi-EP TCP/IP Ethernet Communications User Manual



## Warnings and Caution Notes as Used in this Publication

### WARNING

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

---

### CAUTION

Caution notices are used where equipment might be damaged if care is not taken.

---

**Note:** *Notes merely call attention to information that is especially significant to understanding and operating the equipment.*

These instructions do not purport to cover all details or variations in equipment, nor to provide for every possible contingency to be met during installation, operation, and maintenance. The information is supplied for informational purposes only, and Emerson makes no warranty as to the accuracy of the information included herein. Changes, modifications, and/or improvements to equipment and specifications are made periodically and these changes may or may not be reflected herein. It is understood that Emerson may make changes, modifications, or improvements to the equipment referenced herein or to the document itself at any time. This document is intended for trained personnel familiar with the Emerson products referenced herein.

Emerson may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not provide any license whatsoever to any of these patents.

Emerson provides the following document and the information included therein as-is and without warranty of any kind, expressed or implied, including but not limited to any implied statutory warranty of merchantability or fitness for particular purpose.

# Contents

Table of Figures .....	ix
<b>Section 1 Introduction .....</b>	<b>1</b>
<b>1.1 Revisions in this Manual .....</b>	<b>2</b>
<b>1.2 PACSystems Documentation .....</b>	<b>4</b>
1.2.1 PACSystems Manuals .....	4
1.2.2 RX3i Manuals .....	4
1.2.3 RSTi-EP Manuals .....	4
<b>1.3 Ethernet Interfaces for PACSystems Controllers .....</b>	<b>5</b>
1.3.1 RX3i Rack-Based Ethernet Interfaces – Features .....	6
1.3.2 RX3i & RSTi-EP Embedded Ethernet Interface - Features .....	7
1.3.3 Ethernet Interface Specifications .....	8
1.3.4 Ethernet interface Ports .....	11
1.3.5 Station Manager .....	11
1.3.6 Firmware Upgrades .....	12
1.3.7 SRTP Client (Channels) .....	12
1.3.8 Modbus TCP Client (Channels) .....	12
1.3.9 Ethernet Global Data (EGD) .....	12
1.3.10 SRTP Inactivity Timeout .....	14
<b>1.4 Ethernet Redundancy Operation .....</b>	<b>15</b>
1.4.1 Hot Standby (HSB) CPU Redundancy .....	16
1.4.2 Non-HSB Redundancy .....	16
1.4.3 Effect of Redundancy Role Switching on Ethernet Communications .....	17
1.4.4 SRTP Server Operation in a Redundancy System .....	19
1.4.5 SRTP Client Operation in a Redundancy System .....	19
1.4.6 Modbus TCP Server Operation in a Redundancy System .....	20
1.4.7 Modbus TCP Client Operation in a Redundancy System .....	20
1.4.8 EGD Class 1 (Production & Consumption) in a Redundancy System .....	20
1.4.9 EGD Class 2 Commands in a Redundancy System .....	21
1.4.10 Web Server Operation in a Redundancy System .....	21
1.4.11 FTP Operation in a Redundancy System .....	21
1.4.12 SNMP Operation in a Redundancy System .....	21
1.4.13 Remote Station Manager Operation in a Redundancy System .....	22
1.4.14 IP Address Configuration in a Redundancy System .....	22

<b>Section 2</b>	<b>Installation and Start-up: RX3i/RSTi-EP Embedded Interface</b>	<b>24</b>
2.1	<b>RX3i/RSTi-EP Embedded Ethernet Interface Indicators</b>	<b>25</b>
2.1.1	Ethernet Port LEDs Operation	25
2.1.2	Module Installation	28
2.2	<b>Ethernet Port Connector</b>	<b>28</b>
2.2.1	Connection to a 10Base-T/100Base-TX Network	28
2.2.2	10Base-T/100Base-TX Port Pinouts	28
2.3	<b>Pinging TCP/IP Ethernet interfaces on the Network</b>	<b>29</b>
2.3.1	Determining if an IP Address is Already Being Used	29
<b>Section 3</b>	<b>Installation and Start-up: Ethernet Module Interfaces</b>	<b>31</b>
3.1	<b>Ethernet Module Interface Characteristics</b>	<b>32</b>
3.1.1	Front Panel Port	33
3.1.2	Ethernet Port Connections	33
3.1.3	LEDs on the RX3i Ethernet Interface Module	34
3.1.4	Ethernet LEDs	34
3.1.5	Restart/Reset Pushbutton Operation	36
3.2	<b>Ethernet Module Installation</b>	<b>37</b>
3.2.1	Module Installation	37
3.2.2	Module Removal	37
3.3	<b>Ethernet Port Connectors</b>	<b>38</b>
3.3.1	Embedded Switch	38
3.3.2	Connection to a 10Base-T/100Base-TX/1000Base-T Network	40
3.4	<b>Station Manager Port</b>	<b>42</b>
3.4.1	Port Settings	42
3.5	<b>Verifying Proper Power-Up of the Ethernet Interface After Configuration</b>	<b>43</b>
3.6	<b>Pinging TCP/IP Ethernet interfaces on the Network</b>	<b>44</b>
3.6.1	Determining if an IP Address is Already Being Used	44
3.7	<b>Ethernet Plug-in Applications</b>	<b>44</b>
<b>Section 4</b>	<b>Configuration</b>	<b>46</b>
4.1	<b>RX3i/RSTi-EP Embedded Ethernet Interfaces</b>	<b>47</b>
4.1.1	Ethernet Configuration Data	47
4.1.2	Initial IP Address Assignment	48
4.1.3	Configuring the Ethernet Interface Parameters	49

- 4.2 **RX3i Rack-Based Ethernet Interface Modules..... 63**
  - 4.2.1 Ethernet Configuration Data ..... 64
  - 4.2.2 Initial IP Address Assignment ..... 65
  - 4.2.3 Configuring Ethernet Interface Parameters ..... 70
  - 4.2.4 Configuring Ethernet Global Data ..... 77
- Section 5 Ethernet Global Data ..... 94**
  - 5.1 **Comparison Model for Ethernet Global Data Support..... 95**
  - 5.2 **Ethernet Global Data Operation..... 95**
    - 5.2.1 EGD Producer ..... 96
    - 5.2.2 EGD Consumers..... 96
  - 5.3 **EGD Exchanges..... 96**
    - 5.3.1 Content of an Ethernet Global Data Exchange ..... 97
    - 5.3.2 Data Ranges (Variables) in an Ethernet Global Data Exchange ..... 97
    - 5.3.3 Valid Memory Types for Ethernet Global Data ..... 97
    - 5.3.4 Planning Exchanges ..... 99
    - 5.3.5 Using Ethernet Global Data in a Redundancy System ..... 99
  - 5.4 **Sending an Ethernet Global Data Exchange to Multiple Consumers..... 99**
    - 5.4.1 Multicasting Ethernet Global Data ..... 100
    - 5.4.2 Broadcasting Ethernet Global Data ..... 101
    - 5.4.3 Changing Group ID in Run Mode ..... 101
  - 5.5 **Ethernet Global Data Timing ..... 102**
    - 5.5.1 EGD Synchronization ..... 102
    - 5.5.2 Configurable Producer Period for an EGD Exchange ..... 103
    - 5.5.3 Consumer Update Timeout Period..... 103
  - 5.6 **Effect of PLC Modes and Actions on EGD Operations ..... 105**
    - 5.6.1 Run Mode Store of EGD..... 106
  - 5.7 **Monitoring Ethernet Global Data Exchange Status ..... 111**
    - 5.7.1 Exchange Status Word ..... 111
    - 5.7.2 Exchange Status Errors..... 111
    - 5.7.3 Exchange Status Word Error Codes ..... 111

<b>Section 6</b>	<b>Programming EGD Commands.....</b>	<b>113</b>
6.1	General Use of EGD Commands.....	114
6.2	Using EGD Commands in a Redundancy System .....	114
6.3	COMMREQ Format for Programming EGD Commands.....	114
6.4	COMMREQ Status for the EGD Commands .....	116
6.4.1	COMMREQ Status Values .....	116
6.5	Read PLC Memory (4000).....	117
6.5.1	Read PLC Memory Command Block .....	117
6.6	Write PLC Memory (4001).....	120
6.6.1	Write PLC Memory Command Block .....	120
6.7	Read EGD Exchange (4002) .....	122
6.7.1	Read EGD Exchange Command Block .....	122
6.8	Write EGD Exchange (4003) .....	125
6.8.1	Write EGD Exchange Command Block.....	125
6.9	Masked Write to EGD Exchange (4004) .....	127
6.9.1	Masked Write EGD Exchange Command Block .....	127
<b>Section 7</b>	<b>SNTP Operation.....</b>	<b>131</b>
7.1	Normal SNTP Operation .....	132
7.1.1	SNTP Broadcast and Multicast Operation Mode .....	132
7.1.2	SNTP Unicast Operation Mode .....	132
7.2	Multiple SNTP Servers (Applies only to SNTP Broadcast and Multicast Mode) .....	133
7.3	Loss or Absence of SNTP Timing Signals .....	133
7.4	Time-Stamping of Ethernet Global Data Exchanges .....	134
7.4.1	Obtaining Timestamps from the Ethernet Interface Clock.....	135
7.4.2	Obtaining Timestamps from the CPU TOD Clock.....	137
<b>Section 8</b>	<b>Programming SRTP Channel Commands.....</b>	<b>148</b>
8.1	Model Comparison for SRTP Server Capabilities .....	149
8.2	SRTP Channel Commands .....	149
8.2.1	Channel Operations.....	149
8.2.2	Aborting and Re-tasking a Channel .....	150
8.2.3	Monitoring the Channel Status .....	150

8.2.4	SRTP Channel Commands in a Redundant System.....	150
8.2.5	Executing a Channel Command .....	151
<b>8.3</b>	<b>COMMREQ Format for Programming Channel Commands .....</b>	<b>152</b>
8.3.1	The COMMREQ Command Block: General Description .....	153
8.3.2	Establish Read Channel (2003).....	155
8.3.3	Establish Write Channel (2004).....	159
8.3.4	Send Information Report (2010) .....	162
8.3.5	Abort Channel (2001) .....	165
8.3.6	Retrieve Detailed Channel Status (2002).....	166
<b>8.4</b>	<b>Programming for Channel Commands .....</b>	<b>168</b>
8.4.1	COMMREQ Sample Logic .....	168
8.4.2	Sequencing Communications Requests .....	170
8.4.3	Managing Channels and TCP Connections .....	171
8.4.4	Use “Channel Re-Tasking” To Avoid Using Up TCP Connections .....	171
8.4.5	Client Channels TCP Resource Management .....	172
8.4.6	SRTP Application Timeouts .....	172
<b>8.5</b>	<b>Monitoring Channel Status.....</b>	<b>173</b>
8.5.1	Format of the COMMREQ Status Word.....	173
<b>Section 9</b>	<b>Modbus/TCP Server .....</b>	<b>176</b>
<b>9.1</b>	<b>Model Comparison for Modbus/TCP Server Capabilities .....</b>	<b>177</b>
<b>9.2</b>	<b>Modbus Conformance Classes.....</b>	<b>178</b>
9.2.1	Server Protocol Services.....	178
9.2.2	Station Manager Support.....	178
<b>9.3</b>	<b>Reference Mapping.....</b>	<b>178</b>
9.3.1	Modbus Reference Tables .....	178
9.3.2	Address Configuration .....	180
<b>9.4</b>	<b>Modbus Function Codes.....</b>	<b>181</b>
<b>Section 10</b>	<b>Modbus/TCP Client .....</b>	<b>183</b>
<b>10.1</b>	<b>Model Comparison for Modbus/TCP Client Capabilities .....</b>	<b>184</b>
<b>10.2</b>	<b>Communications Request .....</b>	<b>184</b>
10.2.1	Structure of the Communications Request .....	184
10.2.2	COMMREQ Function Block .....	185
10.2.3	COMMREQ Command Block .....	185
10.2.4	Modbus/TCP Channel Commands.....	185
10.2.5	Status Data .....	186

10.2.6	Operation of the Communications Request .....	187
<b>10.3</b>	<b>COMMREQ Function Block and Command Block.....</b>	<b>188</b>
10.3.1	The COMMREQ Function Block .....	188
10.3.2	The COMMREQ Command Block.....	189
<b>10.4</b>	<b>Modbus/TCP Channel Commands .....</b>	<b>191</b>
10.4.1	Open a Modbus/TCP Client Connection (3000) .....	191
10.4.2	Close a Modbus/TCP Client Connection (3001) .....	193
10.4.3	Read Data from a Modbus/TCP Device (3003).....	194
10.4.4	Write Data to a Modbus/TCP Device (3004) .....	200
10.4.5	Mask Write Register Request to a Modbus Server Device (3009) .....	204
10.4.6	Read/Write Multiple Registers to/from a Modbus Server Device (3005) .....	205
<b>10.5</b>	<b>Status Data .....</b>	<b>207</b>
10.5.1	Types of Status Data .....	207
<b>10.6</b>	<b>Controlling Communications in the Ladder Program.....</b>	<b>208</b>
10.6.1	Essential Elements of the Ladder Program .....	208
10.6.2	Managing Channels and TCP Connections .....	209
10.6.3	Client Channels TCP Resource Management .....	209
10.6.4	COMMREQ Ladder Logic Example .....	210
10.6.5	Troubleshooting a Ladder Program.....	216
10.6.6	Monitoring the Communications Channel .....	217



Section 11	OPC UA Server .....	220
11.1	Model Comparison for OPC UA Server Capabilities .....	221
11.2	OPC UA Certificate Security .....	221
11.3	Controlling the OPC UA Server with PAC Machine Edition .....	222
11.4	Log Level Settings.....	225
11.5	Application Logic to Control the OPC UA Server .....	228
11.6	Connect OPC UA Client to OPC UA Server .....	242
11.7	OPC UA Service Performance and Estimation.....	260
Section 12	Diagnostics.....	268
12.1	What to do if You Cannot Solve the Problem .....	269
12.2	Diagnostic Tools Available for Troubleshooting .....	270
12.3	Initialization Example of the RX3i Ethernet ETM001-jx Module Interface .....	271
12.4	ETHERNET OK/OK LED Blink Codes for Hardware Failures (ETM001-jx) .....	273
12.5	Controller Fault Table .....	274
12.5.1	Controller Fault Table Descriptions .....	274
12.6	Monitoring the Ethernet Interface Status Bits .....	277
12.6.1	LAN Interface Status (LIS) Bits .....	280
12.6.2	Channel Status Bits .....	282
12.7	Monitoring the FT Output of the COMMREQ Function Block .....	283
12.8	Monitoring the COMMREQ Status Word.....	284
12.8.1	Format of the COMMREQ Status Word.....	284
12.8.2	Major Error Codes in the COMMREQ Status Word.....	285
12.8.3	Minor Error Codes for Major Error Codes 05H (at Remote Server PLC) and 85H (at Client PLC) .....	286
12.8.4	Minor Error Codes for Major Error Code 11H (at Remote Server PLC) .....	287
12.8.5	Minor Error Codes for Major Error Code 90H (at Client PLC) .....	290
12.8.6	Minor Error Codes for Major Error Code 91H (at Remote Modbus/TCP Server) .....	292
12.8.7	Minor Error Codes for Major Error Code A0H (at Client PLC).....	293
12.9	Using the EGD Management Tool (RX3i Ethernet Module).....	294
12.9.1	Installing the EGD Management Tool .....	294
12.9.2	Launching the EGD Management Tool .....	294
12.9.3	Monitoring EGD Devices .....	295

12.9.4	Monitoring Status of Ethernet Global Data for a Device .....	296
<b>12.10</b>	<b>Troubleshooting Common Ethernet Difficulties.....</b>	<b>299</b>
12.10.1	COMMREQ Fault Errors .....	299
12.10.2	PLC Timeout Errors .....	299
12.10.3	Application Timeout Errors .....	300
12.10.4	EGD Configuration Mismatch Errors.....	301
12.10.5	Station Manager Lockout under Heavy Load .....	301
12.10.6	PING Restrictions .....	301
12.10.7	SRTP and Modbus/TCP Connection Timeout.....	302
12.10.8	Sluggish Programmer Response after Network Disruption .....	302
12.10.9	EGD Command Session Conflicts .....	303
12.10.10	SRTP Request Incompatibility with Existing Host Communications Toolkit Devices or Other SRTP Clients .....	303
12.10.11	COMMREQ Flooding Can Interrupt Normal Operation.....	303
12.10.12	Accelerated EGD Consumption Can Interfere with EGD Production.....	303
12.10.13	Channels Operation Depends Upon PLC Input Scanning.....	304
<b>Section 13</b>	<b>Network Administration .....</b>	<b>306</b>
<b>13.1</b>	<b>IP Addressing .....</b>	<b>307</b>
13.1.1	IP Address Format for Network Classes A, B, C.....	307
13.1.2	IP Addresses Reserved for Private Networks .....	308
13.1.3	Multicast IP Addresses .....	308
13.1.4	Loopback IP Addresses.....	308
13.1.5	Overlapping Subnets .....	308
<b>13.2</b>	<b>Gateways.....</b>	<b>312</b>
13.2.1	Networks Connected by a Gateway.....	312
<b>13.3</b>	<b>Subnets and Supernets .....</b>	<b>312</b>
13.3.1	Subnet Addressing and Subnet Masks .....	313
13.3.2	Example: Network Divided into Two Subnets .....	314
13.3.3	Example: Two Networks Combined into a Supernet .....	315
	<b>SNTP Time Transfer to CPU Parameters (task n).....</b>	<b>322</b>

# Table of Figures

---

Figure 1: Ethernet Connection System Diagram .....	5
Figure 2: Ethernet Operation in Redundancy Mode .....	16
Figure 3: Basic Non-HSB System with Redundant IP .....	17
Figure 4: RJ45 Connector .....	28
Figure 5: Ethernet Cable Routing .....	29
Figure 6: ETM001-Jx Faceplate .....	32
Figure 7: ETM001-Kxxx Faceplate .....	32
Figure 8: Ethernet Port Connectors on IC695ETM001-Kxxx .....	33
Figure 9: Install Module into RX3i Backplane .....	37
Figure 10: Remove Module from RX3i Backplane .....	37
Figure 11: Diagram of Embedded Ethernet Switch .....	38
Figure 12: System Diagram: Ethernet Routing Using Embedded Switch .....	39
Figure 13: Connection Using Hub/Switch/Repeater .....	41
Figure 14: Direct Connection to the Embedded Ethernet Ports .....	42
Figure 15: Expand CPU Slot to Display Ethernet Node .....	48
Figure 16: Expand RX3i CPU Node to Configure Embedded Ethernet interface .....	51
Figure 17: CPE330/CPE400/CPL410/CPE100/CPE115/ETM001-Kxxx Settings tab .....	54
Figure 18: CPE330 Advanced Ethernet Configuration LAN1 & LAN2 .....	55
Figure 19: CPE100/CPE115/CPE400/CPL410 Advanced Ethernet Configuration LAN1 & LAN2 .....	56
Figure 20: SNTP PME configuration for the CPE302/CPE305/CPE310/CPE330/CPE400/CPL410 CPU settings .....	57
Figure 21: SNTP Multicast/Broadcast or Unicast Mode Settings .....	57
Figure 22: UTC Time Zone Settings .....	58
Figure 23: Terminals Tab Settings in PAC Machine Edition .....	59
Figure 24: Adding Ethernet Global Data (EGD) to the Configuration .....	60
Figure 25: Defining EGD Produced Data Exchange .....	61
Figure 26: Defining EGD Consumed Data Exchange .....	61
Figure 27: Configuring Multicast & Broadcast EGD on LAN1 .....	62
Figure 28: Configuring Multicast & Broadcast EGD on LAN2 .....	63
Figure 29: Setting Temporary IP Address .....	66
Figure 30: Install ETM001-Jx Module in Rack/Slot & Expand to Configure .....	70
Figure 31: Expand Node to View Ethernet Global Data .....	77
Figure 32: Local Producer ID .....	78
Figure 33: Configuring the EGD Configuration Server .....	79
Figure 34: Exchange ID Offset in an Ethernet Redundancy System .....	92
Figure 35: Configuring Redundancy for Ethernet Global Data .....	92
Figure 36: Configuring Produce in Backup Mode Parameter .....	93
Figure 37: Producing & Consuming Ethernet Global Data .....	95
Figure 38: Adding Symbolic Reference to Ethernet Global Data Exchange .....	98

Figure 39: Grouping of Devices for Ethernet Global Data Multicasting ..... 100

Figure 40: Memory Sharing between PLC and Ethernet interface ..... 102

Figure 41: EGB Timing Example #1 ..... 104

Figure 42: EGB Timing Example #2 ..... 104

Figure 43: Sample COMMREQ Ladder Diagram ..... 114

Figure 44: Example: Masked Write to EGD Exchange Bitmask and Data Bits ..... 129

Figure 45: Obtaining Timestamps from the Ethernet interface Clock ..... 135

Figure 46: Obtaining Timestamps from the PLC Time Clock ..... 136

Figure 47: Obtaining Timestamps from the SNTP Server’s Time Clock ..... 137

Figure 48: Synchronizing CPU Time-of-Day Clock to an SNTP Server ..... 138

Figure 49: Operating Sequence for CPU Clock Synchronization ..... 139

Figure 50 Hardware Configuration Module in PME ..... 141

Figure 51: COMMREQ to Control the CPU Time-of-Day Clock ..... 142

Figure 52: COMMREQ Sequence for Establish Read Channel ..... 151

Figure 53: COMMREQ for Programming Channel Commands ..... 152

Figure 54: Interpreting Detailed Channel Status Words ..... 167

Figure 55: Sample Ladder Logic for COMMREQ ..... 169

Figure 56: Interpreting COMMREQ Status Word ..... 173

Figure 57: Calculations for Modbus File and Record %W Memory Address ..... 179

Figure 58: Phases of a COMMREQ Execution ..... 185

Figure 59: Illustration of Phased Operation of a COMMREQ ..... 187

Figure 60: The COMMREQ Function Block ..... 188

Figure 61: Interpreting the COMMREQ Status Word ..... 208

Figure 62: COMMREQ Ladder Logic Segment ..... 210

Figure 63: COMMREQ Ladder Logic Segment (continued) ..... 211

Figure 64: COMMREQ Ladder Logic Segment (continued) ..... 212

Figure 65: COMMREQ Ladder Logic Segment (continued) ..... 213

Figure 66: COMMREQ Ladder Logic Segment (continued) ..... 213

Figure 67: COMMREQ Ladder Logic Segment (continued) ..... 214

Figure 68: COMMREQ Ladder Logic Segment (continued) ..... 215

Figure 69: OPC UA Parameters in Hardware Configuration ..... 226

Figure 70: Example of Start OPC UA Server Service Request ..... 229

Figure 71: Example of Stop OPC UA Server Service Request ..... 230

Figure 72: Example of Clear OPC UA Server Service Request ..... 231

Figure 73: Example of Restart OPC UA Server Request ..... 232

Figure 74: SERVER\_STATUS Word bit definitions ..... 233

Figure 75: Example of Get OPC UA Server Status Service Request ..... 234

Figure 76: CONFIG\_STATUS Word bit definitions ..... 235

Figure 77: Server Status Definition ..... 236

Figure 78: Config Status Definition ..... 236

Figure 79: Example of Get Provisioning Mode Status Service Request ..... 237

Figure 80: OPC UA Binary Connection String .....	242
Figure 81: Project Inspector/Ethernet Config Window .....	243
Figure 82: OPC UA Server Client Connection String.....	243
Figure 83: OPC UA Client Connection Dialog.....	244
Figure 84: PME Controller Hardware Configuration – Passwords Disabled.....	245
Figure 85: PME Controller Hardware Configuration – Passwords Enabled.....	246
Figure 86: PME Online Command to Set Passwords .....	247
Figure 87: Example OPC UA Address Space .....	248
Figure 88: PME Variable Inspector.....	249
Figure 89: Application Variable Address Space .....	250
Figure 90: OPC UA Address Space - Server Node .....	251
Figure 91: Server Specific Address Space .....	251
Figure 92: BuildInfo Subscription .....	252
Figure 93: OPC UA Address Space - Application Information.....	252
Figure 94: OPC UA Address Space – Device Information .....	253
Figure 95: OPC UA Device Redundancy Configuration in Cimplicity 10 SIM 17 .....	256
Figure 96: Sweep Time Location .....	262
Figure 97: Add Performance View Document.....	264
Figure 98: Nodes in List .....	265
Figure 99: Performance View Configuration .....	265
Figure 100: Performance View.....	267
Figure 101: States of the Ethernet interface .....	271
Figure 102: Fault Extra Data Example.....	274
Figure 104: Monitoring FT Output in COMMREQ Function Block.....	283
Figure 105: Decoding the COMMREQ Status Word .....	284
Figure 106: EGD Management Tool Screenshot .....	294
Figure 107: EGD Monitoring Tool Monitoring EGD Network .....	295
Figure 108: EGD Management Tool Displaying EGD Exchange Information .....	296
Figure 109: EGD Management Tool Displaying EGD Statistics .....	297
Figure 110: EGD Management Tool Displaying List of Variables for an Exchange.....	298
Figure 111: IP Address Format for Network Classes A, B, C.....	307
Figure 112: CPE330 Overlapping Local IP Subnet Example .....	309
Figure 113: Expected Response Path.....	310
Figure 114: Actual Response Path.....	311
Figure 115: Gateway Connected to Two Networks.....	312
Figure 116: Class B Network netid and hostid Bit Formats .....	313
Figure 117: Use of Subnet Mask.....	313
Figure 118: Network 2 Divided into Subnets 2.1 and 2.2.....	314
Figure 119: Subnet Mask Used to Affect a Supernet .....	315
Figure 120: Resulting Supernet.....	315

# Section 1 Introduction

This chapter includes basic information about Ethernet interfaces for the PACSystems family of controllers. It describes features of the Ethernet interfaces in both conventional and redundancy systems. The rest of this manual provides instructions for installing and applying the PACSystems Ethernet interfaces:

- **Section 2, Installation and Start-up: RX3i/RSTi-EP Embedded Interface** describes user features and basic installation procedures.
- **Section 3, Installation and Start-up: Ethernet Module Interfaces** describes user features and basic installation procedures.
- **Section 4, Configuration** describes assigning a temporary IP address and configuring the Ethernet interface parameters. For the RX3i rack-based and embedded interfaces, describes how to configure Ethernet Global Data (EGD) and set up the RS-232 port for Local Station Manager operation.
- **Section 5, Ethernet Global Data** describes basic EGD operation for rack-based and embedded interfaces.
- **Section 6, Programming EGD Commands** describes a set of commands that can be used in the application program to read and write PLC data or use Ethernet Global Data exchange data over the network.
- **Section 7, SNTP Operation** describes the benefit of synchronizing SNTP-capable interfaces with an SNTP server to keep internal clocks up-to-date for accurate timestamp communications.
- **Section 8, Programming SRTP Channel Commands** explains how to implement PLC to PLC communications over the Ethernet network using Service Request Transfer Protocol (SRTP) Channel commands.
- **Section 9, Modbus/TCP Server** describes the implementation of the Modbus TCP Server feature for the PACSystems family of products.
- **Section 10, Modbus/TCP Client** explains how to program communications over the Ethernet network using
  - Modbus TCP Channel commands.
- **Section 11, OPC UA Server** explains how to program communications for this protocol using the embedded Ethernet port.
- **Section 12, Diagnostics** describes diagnostic techniques for a PACSystems Ethernet interface. This chapter also lists COMMREQ Status codes.
- **Section 13, Network Administration** discusses how devices are identified on the network and how data is routed among devices.
- **Appendix A, Configuring Advanced User Parameters** describes an optional configuration of internal operating parameters used by the Ethernet interface. For most applications, the default Advanced User Parameters (AUPs) should not be changed.

## 1.1 Revisions in this Manual

A given feature may not be implemented on all PACSystems Ethernet interfaces. To determine whether a feature is available on a given model and firmware version, please refer to the *Important Product Information (IPI)* document provided with the product.

This revision of TCP/IP Ethernet Communications for PACSystems RX3i and RSTi-EP includes the following changes:

Rev	Date	Description
AB	Jan-2021	<ul style="list-style-type: none"> <li>Introduces OPC UA Non-transparent Server Redundancy on the CPE330 R10.10, CPE400/CPL410 R10.10</li> <li>Caution added with regard to redundant PLCs that perform a role switch.</li> </ul>
AA	Jun-2020	<ul style="list-style-type: none"> <li>Added content for the Produce EGD on Redundant IP feature</li> <li>Section 12.8.5. Error Code 0290H was updated to include behavior when sending TCP comm to a non-existent server through a gateway.</li> </ul>
Z	Feb-2020	<ul style="list-style-type: none"> <li>Content updated in support of CPE100/115.</li> </ul>
Y	Aug-2019	<ul style="list-style-type: none"> <li>RX3i IC695ETM001-Kxxx Available                             <ul style="list-style-type: none"> <li>Backward compatible with IC695ETM001</li> <li>Station Manager serial port replaced with Ethernet port</li> <li>Two Ethernet connectors are relocated to the bottom of the module.</li> </ul> </li> <li>Achilles Level 2 Security cert-tested.</li> <li>New option to select user-based parameters into menu systems. AUP functionalist is partially deprecated.</li> <li>Diagnostics information for the RX3i embedded Ethernet interface has been moved from Chapter 12 to Chapter 11.</li> </ul>
W	Jul-2018	<ul style="list-style-type: none"> <li>Added IC695CPL410 (new CPU w/Linux)</li> </ul>
V	Apr-2018	<ul style="list-style-type: none"> <li>Extended the document to EPSCPE115</li> </ul>
U	Feb-2018	<ul style="list-style-type: none"> <li>Addition of CPE302 throughout.</li> <li>Clarification as to which products support 1000Base-T IEEE 802.3.</li> </ul>
T	Oct-2017	<ul style="list-style-type: none"> <li>Added CPE400 LAN3 (Redundancy-only LAN)</li> <li>Clarified support for Redundant IP Addressing in various CPU configurations.</li> </ul>
S	Aug-2017	<ul style="list-style-type: none"> <li>Content added to Ethernet interface Status Bits for RSTi-EP CPE100.</li> </ul>
R	May-2017	<ul style="list-style-type: none"> <li>Content added in support of RSTi-EP CPE100.</li> </ul>
Q	Mar-2017	<ul style="list-style-type: none"> <li>Content added in support of CPE400 and embedded SNTP.</li> </ul>
P	Sept-2015	<ul style="list-style-type: none"> <li>Added section</li> <li><i>Sessions and Subscriptions for OPC UA.</i></li> <li>Content added in support of CPE330 (new product).</li> </ul>

M	Oct-2014	<ul style="list-style-type: none"> <li>• Effective with RX3i CPE305/CPE310 firmware version 8.20, OPC UA Server is supported using the embedded Ethernet port.</li> <li>• Effective with RX3i CPE305/CPE310 firmware version 8.30, EGD Class 1 is supported on the embedded Ethernet interface. Earlier CPU versions do not directly support EGD. However, EGD was supported on the Ethernet interface Module ETM001.</li> </ul>
L	Jun-2013	<p>Newly available features:</p> <ul style="list-style-type: none"> <li>• TCP/IP communication services using SRTP</li> <li>• SRTP Client (Channels)</li> <li>• Modbus/TCP Server, supporting Modbus Conformance classes 0, 1, and 2.</li> <li>• Modbus/TCP Client, supporting Modbus Conformance classes 0, 1, and Function Codes 15, 22, 23, and 24 for Conformance class 2.</li> <li>• Support for Unicast mode, and Daylight Saving and Local Time corrections for SNTP operation.</li> </ul> <p>Diagnostics information for the RX3i embedded Ethernet interface has been moved from 2 0to To 11. Configuration information has been moved to Section 4.</p> <p>Information about Channel Status bits has been removed from sections 2, 7 and 9, and consolidated in Section 11.</p>



## 1.2 PACSystems Documentation

### 1.2.1 PACSystems Manuals

<i>PACSystems RX3i and RSTi-EP CPU Reference Manual</i>	GFK-2222
<i>PACSystems RX3i and RSTi-EP CPU Programmer's Reference Manual</i>	GFK-2950
<i>PACSystems TCP/IP Ethernet Communications Station Manager User Manual</i>	GFK-2225
<i>PACSystems Hot Standby CPU Redundancy User's Guide</i>	GFK-2308
<i>PAC Machine Edition Logic Developer Getting Started</i>	GFK-1918
<i>PACSystems RXi, RX3i and RSTi-EP Controller Secure Deployment Guide</i>	GFK-2830
<i>PACSystems RX7i Installation Manual</i>	GFK-2223

### 1.2.2 RX3i Manuals

<i>PACSystems RX3i System Manual</i>	GFK-2314
<i>PACSystems RX3i Ethernet Network Interface Unit (NIU) User's Manual</i>	GFK-2439
<i>PACSystems RX3i IEC 61850 Ethernet Communication Module User Manual</i>	GFK-2849
<i>PACSystems RX3i Serial Communications Modules User Manual</i>	GFK-2460
<i>PACSystems RX3i IEC 104 Server Module IC695EIS001 User's Manual</i>	GFK-2949
<i>PACSystems RX3i IC695CPE400 1.2GHz 64MB Rackless CPU w/Field Agent QSG</i>	GFK-3002
<i>PACSystems RX3i IC695CPL410 1.2GHz 64MB Rackless CPU w/Linux QSG</i>	GFK-3053
<i>PACSystems RX3i Sequence of Events User Manual</i>	GFK-3050

### 1.2.3 RSTi-EP Manuals

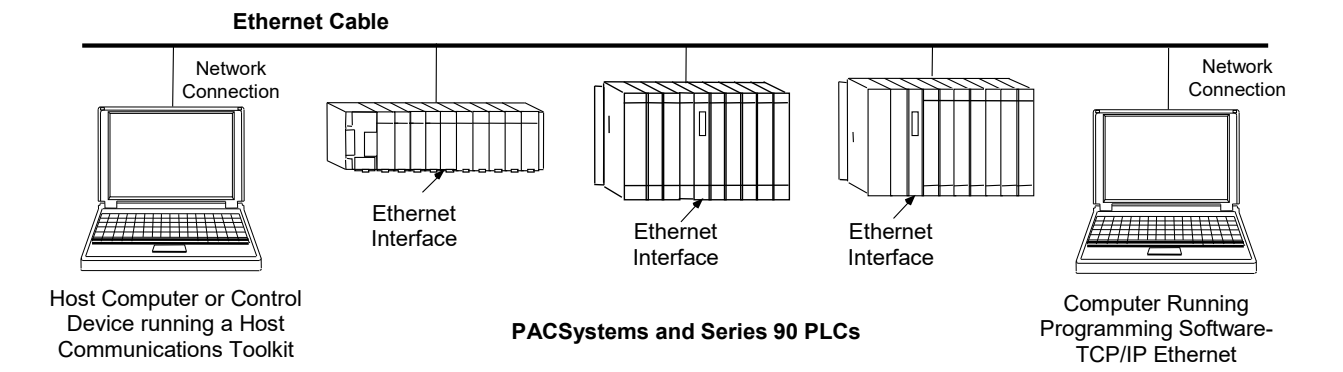
<i>PACSystems RSTi-EP System Manual</i>	GFK-2958
---	----------

## 1.3 Ethernet Interfaces for PACSystems Controllers

A PACSystems Ethernet interface enables a PACSystems controller to communicate with other PACSystems equipment and with Series 90 and VersaMax controllers. The Ethernet interface provides TCP/IP communications with other PLCs, host computers running the Host Communications Toolkit or CIMPLICITY software, and computers running the TCP/IP version of the programming software. These communications use the proprietary SRTP and Ethernet Global Data (EGD) protocols over a four-layer TCP/IP (Internet) stack.

The Ethernet interface has SRTP client/server capability. As a client, the interface can initiate communications with other PLCs that contain Ethernet interfaces. This is done from the PLC ladder program using the COMMREQ function. As a *server*, the Ethernet interface responds to requests from devices such as PLC programming software, a Host computer running an SRTP application, or another PLC acting as a client.

**Figure 1: Ethernet Connection System Diagram**



## 1.3.1 RX3i Rack-Based Ethernet Interfaces – Features

- Full RX3i Controller programming and configuration services with an inactivity timeout
- Periodic data exchange using Ethernet Global Data (EGD)
- EGD Commands to read and write PLC and EGD exchange memory over the network
- TCP/IP communication services using SRTP
- SRTP Client (Channels)
- Modbus TCP Server, supporting Modbus Conformance classes 0, 1, and 2
- Modbus TCP Client, supporting Modbus Conformance classes 0, 1, and Function Codes 15, 22, 23, and 24 for Conformance class 2
- Redundant IP Addressing capability
- Comprehensive station management and diagnostic tools
- Extended controller connectivity via IEEE 802.3 CSMA/CD 10 Mbps, 100M bps, and 1000 Mbps Ethernet LAN port connectors
- A network switch that has Auto negotiate, Sense, Speed, and crossover detection
- The protocol is stored in flash memory in the Ethernet interface and is easily upgraded through the CPU serial port.
- Communications with remote PLCs and other nodes reachable through routers. The gateway IP address must be configured.

## 1.3.2 RX3i & RSTi-EP Embedded Ethernet Interface - Features

- Periodic data exchange using Ethernet Global Data (EGD).
- Full RX3i controller programming and configuration services with an inactivity timeout
- TCP/IP communication services using SRTP.
- SRTP Client (Channels)
- Modbus TCP Server, supporting Modbus Conformance classes 0, 1, and 2.
- Modbus TCP Client, supporting Modbus Conformance classes 0, 1, and Function Codes 15, 22, 23, and 24 for Conformance class 2.
- Communications with remote PLCs and other nodes reachable through routers. The Gateway IP address must be configured.
- Comprehensive station management and diagnostic tools. For supported commands, refer to the *PACSystems TCP/IP Ethernet Communications Station Manager User Manual*, GFK-2225J, or later.

### CPE302/CPE305/CPE310

- Extended controller connectivity via IEEE 802.3 CSMA/CD 10 Mbps and 100 Mbps Ethernet LAN port connectors.
- A network switch that has Auto negotiate, Sense, Speed, and crossover detection.
- Direct connection to Base-T (twisted pair) network switch, hub, or repeater without an external transceiver.

### CPE330/CPE400/CPL410

- Two independent 10/100/1000 Ethernet LANs under the control of the embedded RX3i PLC. Port 1 attaches to LAN1 through a dedicated RJ45 connector. Port 2 attaches to LAN2 through a pair of internally-switched RJ45 connectors. Space is provided to mark the two corresponding IP addresses.
- The embedded Ethernet interface permits the CPU to support two LANs.
- CPE400 has a third Ethernet port (located on the underside) which is under the control of the Field Agent.
- CPL410 also has a third Ethernet port (located on the underside) which is under the control of the Linux OS.

## RSTi-EP CPE100/CPE115

- Two independent 10/100 Ethernet LANs. Port 1 attaches to LAN1 through a dedicated RJ45 connector. Port 2 attaches to LAN2 through three internally-switched RJ45 connectors.
- The embedded Ethernet interface permits the CPU to support two LANs.

Refer to the PACSystems RX7i, RX3i, and RSTi-EP CPU Reference Manual, GFK-2222, specifically to the section, RX3i CPU Features and Specifications for RX3i CPUs & RSTi-EP CPU Features and Specifications for RSTi-EP CPU, for a detailed list of features and specifications.

### 1.3.3 Ethernet Interface Specifications

RX3i Rack-Based Ethernet Interface Modules Connectors	<b>IC695ETM001-jx or earlier</b> - Two RJ45 connectors - One 9-pin d-sub male serial connector (Station Manager port)
	<b>IC695ETM001-Kxxx</b> Three autosensing RJ45 ports
LAN	IC695ETM001-jx or earlier: IEEE 802.3 CSMA/CD Medium Access Control 10/100 Mbps IC695ETM001-Kxxx: IEEE 802.3 CSMA/CD Medium Access Control 10/100/1000 Mbps
Number of IP addresses	One
Maximum number of simultaneous connections	<ul style="list-style-type: none"> <li>▪ A maximum of 48 SRTP Server total connections</li> <li>▪ A maximum of 16 Modbus/TCP Server connections</li> <li>▪ A maximum of 32 communication channels. (Each channel may be an SRTP Client or a Modbus/TCP Client. Any given channel can be assigned to only one protocol at a time.)</li> </ul>
Embedded Ethernet Switch	Yes – Allows daisy-chaining of Ethernet nodes.
Serial Port	<b>IC695ETM001-jx</b> Station Mgr Port: RS-232 DCE, 1200 - 115200 bps.
	<b>IC695ETM001-Kxxx</b> Not applicable.
Station Manager	<b>IC695ETM001-jx</b> Access via local serial port or remote UDP. Refer to the PACSystems TCP/IP Ethernet Communications Station Manager User Manual, GFK-2225J or later, for supported commands.
	<b>IC695ETM001-Kxxx</b> Station Manager serial port has been replaced by the front panel Ethernet port
Maximum ETM001 Modules per CPU rack	Eight positions

## RX3i Embedded Interface

Connector	<p>CPE302/CPE305 &amp; CPE310: One RJ45 connector</p> <p>CPE330: Three RJ45 connectors</p> <p>CPE400: Six RJ45 connectors: five on the front for LAN1, LAN2 &amp; LAN3; one EFA on the underside. (There is also a serial RJ45 on the underside, marked COM1.)</p> <p>CPL410: Six RJ45 connectors: five on the front for LAN1, LAN2 &amp; LAN3; one ETH on the underside. (There is also a serial RJ45 on the underside, marked COM1.)</p> <p>CPE100/CPE115: Four RJ45 connectors</p>
LAN	<p>IEEE 802.3 CSMA/CD Medium Access Control 10/100/1000 Mbps</p> <p>CPE302/CPE305 &amp; CPE310 has one 10Base-T/100Base-TX Port (LAN1)</p> <p>CPE330 has two independent 10/100 Mbps Ethernet LANs:</p> <ul style="list-style-type: none"> <li>▪ The top Ethernet port attaches to LAN1 using a dedicated RJ45 connector</li> <li>▪ The bottom two Ethernet ports attach to LAN2 using a pair of internally-switched RJ45 connectors</li> </ul> <p>CPE400 supports four independent 10/100/1000 Ethernet LANs which are under the control of the embedded RX3i PLC.</p> <ul style="list-style-type: none"> <li>▪ LAN1 attaches via the upper, dedicated RJ45 front-panel connector.</li> <li>▪ LAN2 and LAN3<sup>1</sup> each attach via a pair of internally-switched RJ45 front-panel connectors.</li> <li>▪ The fourth LAN, labeled EFA (Embedded Field Agent), is located on the underside and is specifically used for Field Agent connectivity.</li> </ul> <p>CPL410 supports four independent 10/100/1000 Ethernet LANs which are under the control of the embedded RX3i PLC.</p> <ul style="list-style-type: none"> <li>▪ LAN1 attaches via the upper, dedicated RJ45 front-panel connector.</li> <li>▪ LAN2 and LAN3 each attach via a pair of internally-switched RJ45 front-panel connectors.</li> </ul> <p>The fourth LAN, labeled ETH (Ethernet), is located on the underside and is under the control of the embedded Linux Operating System.</p> <p>CPE100/CPE115 supports two independent 10/100 Ethernet LANs located on the front panel.</p> <ul style="list-style-type: none"> <li>▪ LAN1 attaches via the upper, dedicated RJ45 connector.</li> <li>▪ LAN2 attaches via three internally-switched RJ45 connectors.</li> </ul>
Number of IP addresses	<p>CPE302/CPE305 &amp; CPE310: One IP address</p> <p>CPE330 has two IP addresses</p> <p>CPE400 has four IP addresses (one for EFA, three for Ethernet LANs)</p> <p>CPL410 has four IP addresses (one for ETH, three for Ethernet LANs)</p> <p>CPE100/CPE115 has two IP addresses</p>

<sup>1</sup> CPE400 firmware version 9.30 supports Redundancy via LAN3. No LAN components other than the two Redundant CPUs are permitted on LAN3. All firmware versions of CPL410 support the same feature.

<p>Maximum number of connections</p>	<p>For CPE302/CPE305 &amp; CPE310 LAN1:</p> <ul style="list-style-type: none"> <li>▪ Up to 32 SRTP Server connections, includes:</li> <li>▪ Up to 16 simultaneous Modbus/TCP Server connections.</li> <li>▪ Up to 16 Client channels. (Each channel may be an SRTP Client or a Modbus/TCP Client. Any given channel can be assigned to only one protocol at a time.)</li> <li>▪ OPC UA Server with support for up to 5 concurrent sessions with up to 10 concurrent variable subscriptions and up to 12,500 variables.</li> <li>▪ Up to 255 simultaneous Class 1 Ethernet Global Data (EGD) exchanges.</li> </ul> <p>For CPE330, CPE400<sup>1</sup> and CPL410, the embedded Ethernet permits the CPU to support LAN1 and LAN2 with:</p> <ul style="list-style-type: none"> <li>▪ Up to 48 simultaneous SRTP Server connections, and</li> <li>▪ Up to 16 simultaneous Modbus/TCP Server connections</li> <li>▪ Up to 32 Clients are permitted; each may be SRTP or Modbus/TCP</li> <li>▪ OPC UA Server with support for up to 5 concurrent sessions with up to 10 concurrent variable subscriptions and up to 12,500 variables</li> <li>▪ Up to 255 simultaneous Class 1 Ethernet Global Data (EGD) exchanges.</li> </ul> <p>For CPE100/CPE115, the embedded Ethernet permits the CPU to support LAN1 and LAN2 with:</p> <ul style="list-style-type: none"> <li>▪ Up to 16 simultaneous SRTP Server connections, and</li> <li>▪ Up to 8 simultaneous Modbus/TCP Server connections</li> <li>▪ Up to 8 Clients are permitted; each may be SRTP or Modbus/TCP</li> <li>▪ Up to 8 simultaneous Class 1 Ethernet Global Data (EGD) exchanges.</li> </ul>
<p>Station Manager</p>	<p>Access remote UDP <i>Refer to the PACSystems TCP/IP Ethernet Communications Station Manager User Manual, GFK-2225J</i> or later for supported commands.</p>

## 1.3.4 Ethernet interface Ports

The PACSystems Ethernet interface use auto-sensing 10Base-T/100Base-TX/1000Base-T RJ45 shielded twisted pair Ethernet ports for connection to either a 10BaseT, 100BaseTX, or 1000Base-T IEEE 802.3 network.

The RX3i CPE330, CPE400, and CPL410 embedded Ethernet interface additionally supports 1000Base-T IEEE 802.3 connections.

The RX3i Controllers with embedded Ethernet provide one such port; dedicated Ethernet interface Modules provide two.

The port automatically senses the speed (10 Mbps, 100 Mbps, or 1000Mbps), duplex mode (half-duplex or full-duplex), and cable configuration (straight-through or crossover) attached to it with no intervention required.

### Ethernet Media

The Ethernet interface can operate directly on 10Base-T/100Base-TX/1000Base-T media via its network ports.

**10Base-T:** 10Base-T uses a twisted pair cable of up to 100 meters in length between each node and a switch, hub, or repeater. Typical switches, hubs, or repeaters support connections in a star topology.

**100Base-TX:** 100Base-TX uses a cable of up to 100 meters in length between each node and a switch, hub, or repeater. The cable should be data grade Category 5 unshielded twisted pair (UTP) or shielded twisted pair (STP) cable. Two pairs of wire are used, one for transmission, and the other for collision detection and receive. Typical switches, hubs, or repeaters support 6 to 12 nodes connected in a star wiring topology.

**1000Base-T:** 1000Base-T uses a cable of up to 100 meters in length between each node and a switch, hub, or repeater. The cable should be data grade Category 6 unshielded twisted pair (UTP) or shielded twisted pair (STP) cable or better. Four pairs of wire are used which are designed to operate over 4-pair UTP cable and supports full-duplex data transfer at 1000Mbps. Typical switches, hubs, or repeaters support 6 to 12 nodes connected in a star wiring topology.

## 1.3.5 Station Manager

The built-in Station Manager function of the Ethernet interface provides on-line supervisory access to the Ethernet interface, through the Station Manager port or over the Ethernet cable. Station Manager services include:

- An interactive set of commands for interrogating and controlling the station.
- Unrestricted access to observe internal statistics, an exception log, and configuration parameters.
- Password security for commands that change station parameters or operation.

For remote Station Manager operation over the Ethernet network, the Ethernet interface uses IP addresses. A PACSystems Ethernet interface cannot send or receive remote Station Manager messages sent to a MAC address.

Refer to the *PACSystems TCP/IP Ethernet Communications Station Manager User Manual*, GFK-2225 for complete information on the Station Manager.



## 1.3.6 Firmware Upgrades

PACSystems Ethernet interfaces receive their firmware upgrades indirectly from the RX3i CPU using the WinLoader software utility. WinLoader is supplied with any updates to the Ethernet interface software. The user connects WinLoader to the PLC CPU serial port and specifies the target module by its Rack/Slot location.

For the CPU module, the embedded Ethernet interface firmware is upgraded along with the rest of the CPU firmware. WinLoader seamlessly upgrades first the CPU firmware and then the embedded Ethernet firmware without user intervention. Each Ethernet interface module's firmware must be explicitly upgraded by specifying the rack and slot location of the module to the WinLoader utility.

Firmware upgrades for the CPE330, CPE400, CPL410, and CPE100/CPE115 are performed over Ethernet using a web browser. This method provides enhanced security features. Instructions for the procedure are included in the corresponding upgrade kit documentation. The WinLoader utility will not work with the CPE330, CPE400, CPL410, or CPE100/CPE115 CPUs.

## 1.3.7 SRTP Client (Channels)

SRTP Client allows the PACSystems PLC to initiate data transfer with other SRTP-capable devices on the network. SRTP channels can be set up in the PLC application program. SRTP supports COMMREQ-driven channel commands to establish new channels, abort existing channels, transfer data on an existing channel, and retrieve the status of an existing channel.

Any given channel can be assigned to only one protocol at a time. For the number and combinations of channels supported, refer to Section Modbus TCP Client (Channels)

Modbus TCP Client allows the PACSystems PLC to initiate data transfer with other Modbus TCP server devices on the network. Modbus TCP channels can be set up in the application program. The Modbus TCP Client supports COMMREQ-driven channel commands to open new channels, close existing channels, and transfer data on an existing channel.

Any given channel can be assigned to only one protocol at a time. For the number and combinations of channels supported, refer to Section 1.3.3, *Ethernet Interface Specifications*.

## 1.3.8 Ethernet Global Data (EGD)

- EGD Classes:
  - EGD Class 1 is configured exchanges with no logic control of EGD operation.
    - Supported in CPE302/CPE305/CPE310/CPE330/CPE400/CPL410/CPE100/CPE115
  - EGD Class 2 is EGD Commands which are logic-driven EGD exchanges using COMMREQs.
    - Supported on IC695ETM001
    - Not supported on embedded Ethernet ports of CPE302/CPE305/CPE310/CPE330/CPE400/CPL410/CPE100/CPE115 at time of publication

Each PACSystems RX3i CPU supports up to 255 Class 1 simultaneous EGD exchanges and RSTi-EP CPU CPE100/CPE115 supports up to eight Class 1 simultaneous EGD exchanges. EGD exchanges are configured using the programmer and stored in the PLC. Both Produced and Consumed exchanges can be configured. PACSystems Ethernet interfaces support both selective consumption of EGD exchanges and EGD exchange production and consumption to the broadcast IP address of the local subnet.

**Note:** For Broadcast addressing a Subnet value of 0.0.0.0 is NOT supported.

## Synchronizing EGD Timestamps with SNTP

Both the ETM001-Jx and -Kxxx Ethernet interfaces can be configured to use Simple Network Time Protocol (SNTP) to synchronize the timestamps of produced EGD exchanges.

With an appropriate *PME Hardware Configuration*, the embedded Ethernet interface on the CPE302, CPE305, CPE310, CPE330, CPE400, CPL410, and CPE115 will also support SNTP. For more information on *PME Hardware Configuration*, please refer to Section 4.1.3, *Configuring the Ethernet Interface Parameters*.

**Note:** The RSTi-EP CPE100 does not support SNTP.

Ethernet interface module	SNTP Support
ETM001-Jx	Yes, with PME Hardware Configuration (default configuration without using AUP file)
ETM001-Kxxx	Yes, with PME Hardware Configuration
CPU	SNTP Support
CPL410	Yes, with PME Hardware Configuration
CPE400	Yes, with PME Hardware Configuration
CPE330	Yes, with PME Hardware Configuration
CPE302/CPE305/ CPE310	Yes, with PME Hardware Configuration
CPE100	Not supported
CPE115	Yes, with PME Hardware Configuration

### 1.3.9 SRTP Inactivity Timeout

Starting with Release 6.00, the PACSystems Ethernet interface supports inactivity timeout checking on Secure Real-Time Transport Protocol (SRTP) server connections with any PAC Machine Edition (PME) PLC programmer. Until the server connection is removed, other programmers cannot switch from Monitor to Programmer mode. With inactivity timeout checking, the Ethernet interface removes an abandoned SRTP server connection and all its resources when there is no activity on the connection for the specified timeout interval. Without the SRTP inactivity timeout, an abandoned SRTP server connection persists until the underlying TCP connection times out (typically 7 minutes). All network PME programmer connections initially use an SRTP inactivity timeout value of 30 seconds (as set by the "vconn\_tout" AUP parameter).

PME programmers can override the initial timeout value on a specified server connection. Typically, the PME programmer sets the SRTP inactivity timeout to 20 seconds. An inactivity timeout value of zero disables SRTP inactivity timeout checking.

The SRTP server uses an internal inactivity timeout resolution of 5 seconds. This has two effects. First, any non-zero inactivity timeout value (either set by the AUP parameter or overridden on the programmer connection) is rounded up to the next multiple of 5 seconds. Additionally, the actual SRTP inactivity timeout detection for any individual connection may vary up to an additional 5 seconds. The actual inactivity detection time will never be less than the specified value.

**Note:** The SRTP inactivity timeout applies only to programmer connections over SRTP. It does not affect HMI or SRTP channels.

## 1.4 Ethernet Redundancy Operation

The Redundant IP feature allows a single IP address to be assigned to two Ethernet modules, where the two modules are in two different PLCs configured as a redundant system. This functionality has been integrated into the product line, as follows:

CPU	Embedded Ethernet Redundancy Support	Support via Ethernet Module (ETM001-jx or ETM001-Kxxx)
CPL410	All firmware versions	Not supported
CPE400	Embedded Ethernet requires CPU Firmware Version 9.30	Not supported
CPE330	Embedded Ethernet requires CPU Firmware Version 8.70	Supported
CPE302/CPE305/ CPE310	Not supported	Not supported
CPE100/CPE115	Not supported	Not supported
CRU320	Not supported	Supported

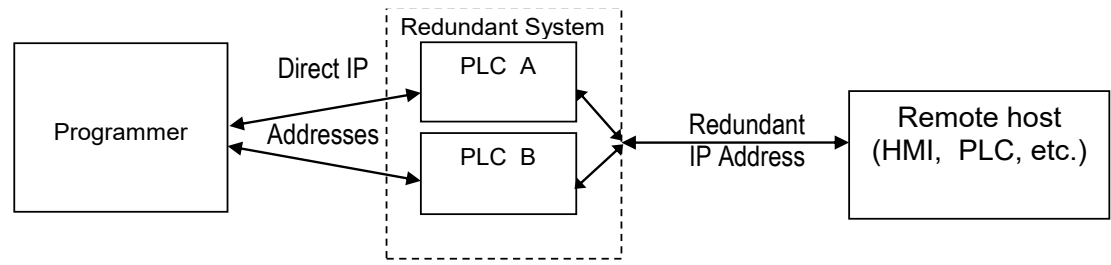
The Redundant IP Address is configured *in addition to* the normal unique (direct) IP address of each interface.

Only one of the two Ethernet interfaces that share the Redundant IP address may use the Redundant IP address at any time; this is the “active” unit. When commanded by its PLC CPU, this Ethernet interface activates the Redundant IP address and starts responding to the Redundant IP address in addition to its direct IP address. The active unit continues responding to the Redundant IP address until it is commanded to deactivate the Redundant IP or until the Ethernet interface determines that it has lost communications with the PLC CPU.

The backup unit does not initiate communications or respond on the network using the Redundant IP address. It can only use the Redundant IP address if it is commanded by its CPU to become the active unit.

Both the active and backup unit may continue to use their direct IP addresses, permitting programmer connection to the active or backup PLC at any time.

Figure 2: Ethernet Operation in Redundancy Mode



**Note:** The Redundant IP feature is supported by Hot Standby (HSB) CPUs and non-HSB CPUs. To use this feature, be sure to toggle *Enable Redundancy* for the target CPU.

### 1.4.1 Hot Standby (HSB) CPU Redundancy

An HSB system uses redundant CPUs to provide the coordination between the PLC units in the system and determine which is the active unit and which is the backup unit. HSB redundancy requires dedicated links to provide communications between the units in a redundancy system. For information about HSB architectures, refer to the *PACSystems Hot Standby CPU Redundancy User's Guide*, GFK-2308.

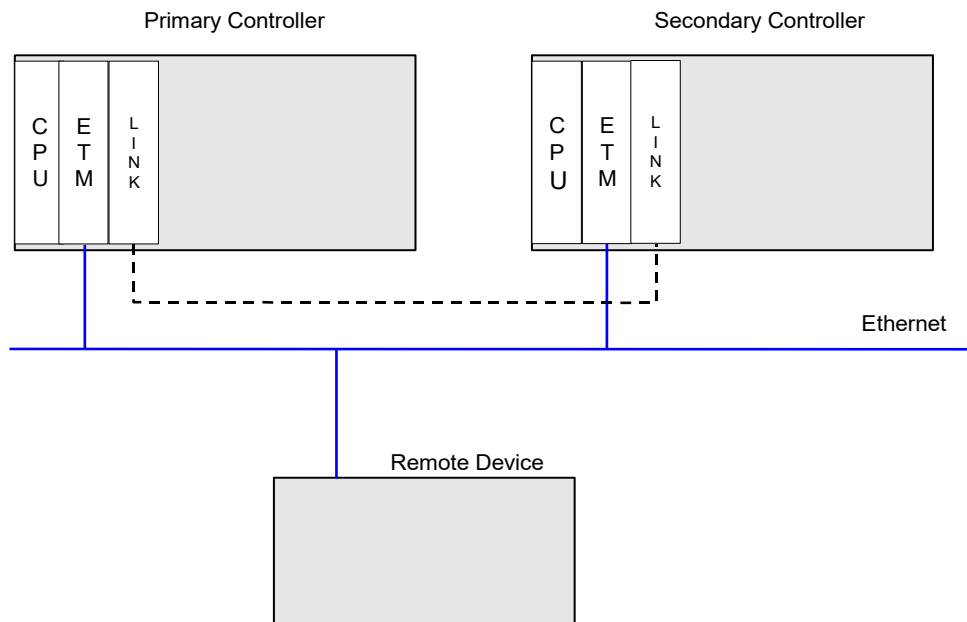
### 1.4.2 Non-HSB Redundancy

Non-HSB redundancy systems use RX3i CPUs that do not have specialized firmware for controlling redundancy operations. (These CPUs have a “CPE” or “CPU” designation.) In these systems, the application logic coordinates between CPUs that act as redundant partners and determines which CPU is the active unit and which are backup units.

Figure 3 illustrates the use of the redundant IP feature in a non-HSB redundancy system. Two non-HSB CPUs (designated primary and secondary) are linked by a communications connection. An Ethernet interface in each controller is configured with Redundant IP enabled so that they share a Redundant IP address. As in an HSB system, only the active Ethernet interface can communicate through the Redundant IP address to produce EGD exchanges or to initiate Channel operations.

The application logic must monitor the status of the Ethernet modules in the system to manage the active/backup status of each controller.

Figure 3: Basic Non-HSB System with Redundant IP



### 1.4.3 Effect of Redundancy Role Switching on Ethernet Communications

When a redundancy role-switch occurs, Ethernet communications switch to the backup unit, which does not know any communication state at the previously-active unit. The application must include logic to detect loss of communication during a redundancy role switch and to then reinitiate communication.

Remote hosts on the network view redundant systems as a single PLC with high reliability; the remote host only prioritizes the active unit. By using the Redundant IP address, the remote host always communicates with the active unit. When a redundancy role switch occurs, the formerly-active PLC gives up ownership of the Redundant IP address and takes down all connection-oriented communications currently using the Redundant IP address. The applications in the redundant system and remote hosts must reestablish any such communications; the new Redundant IP connections will use the newly active PLC.

The programmer can still communicate directly with each PLC in the redundant system (for example, to store new logic or configuration) using the direct IP address of each Ethernet interface.

## Role Switching in HSB Redundancy Systems

In HSB redundancy systems, a role switch is initiated automatically by the redundant CPU when one of the following occurs:

- An active unit detects a fatal fault
- An active unit is placed in Stop mode
- An active unit is powered off
- An HSB role switch is initiated manually or by the application logic

To perform a role switch manually in redundant systems that employ RMX modules, toggle the *Role Switch* button located on the front panel of the RMX module.

CPE400/CPL410 permits the operator to manually perform a role switch via the OLED display menu, using the RDN Command feature.

For additional information about role switches in HSB systems, refer to the *PACSystems Hot Standby CPU Redundancy User's Guide*, GFK-2308.

## Role Switching in Non-HSB Redundancy Systems

When redundant IP is enabled for an Ethernet module in a non-HSB CPU system, it is the responsibility of application logic to set the redundancy mode of the Ethernet module. The *Set Application Redundancy Mode Service Request* (SVC\_REQ 55) instruction is used to inform the Ethernet module of the current redundancy role of the host CPU. This SVC\_REQ should be used to provide redundancy role switch notification to all Ethernet interfaces in the controller that are configured for redundant IP operation.

After commanding a role switch for an Ethernet interface, the application logic can monitor the module's LAN interface Status (LIS) block to determine when it has activated the Redundancy IP address. For details about the LIS, refer to Section 12.6, *Monitoring the Ethernet Interface Status Bits*.

**Note:** The application must allow sufficient time for Redundant IP activation (at least 120 ms) before commanding another redundancy role switch.

When an Ethernet interface recognizes that a redundant IP address has been configured for it, the module sends a mail message to the CPU to register for redundancy role switch notification. In non-HSB systems, the Ethernet interface is initially put into backup mode. After power-up, the application logic must use an SVC\_REQ to set the redundancy state to the desired value. Once running, the CPU remembers the last commanded redundancy role sent to that Ethernet interface. When an Ethernet interface is restarted, the CPU automatically commands the Ethernet interface to its last redundancy state without an explicit action by the application logic.

## Going to Stop Mode

When a non-HSB CPU goes to Stop mode, Ethernet interfaces that are configured for redundant IP are automatically set to backup mode. When the CPU is subsequently returned to Run mode, the Ethernet interfaces remain in backup mode until the application logic sets the redundancy mode to active.

## Stop/IO Scan Enabled Mode

In this mode, I/O scanning including EGD service continues when the non-HSB CPU is stopped. However, Ethernet interfaces configured for redundant IP operation are automatically set to backup mode and normal EGD production for those interfaces is stopped. Only the EGD exchanges with *Produce in backup mode* enabled are produced while the CPU is in Stop/IO Scan Enabled mode. To stop production for all EGD produced exchanges including *Produce in backup mode* exchanges, choose the Stop/IO Scan Disabled mode of operation.

## Commanding a Role Switch in a Non-HSB Redundancy System

Use the Set Application Redundancy Mode service request (SVC\_REQ 55) with non-HSB CPUs to request that the CPU send redundancy role switch commands to all Ethernet interfaces in that PLC that are configured for redundant IP operation. For details on using the Service Request function, refer to the *PACSystems RX7i, RX3i, and RSTi-EP CPU Reference Manual*, GFK-2222.

SVC\_REQ 55 is recognized in non-HSB CPUs only. This service request sends a role switch command to all Ethernet interfaces in the PLC that are configured for redundant IP operation. The application must monitor the LAN interface Status (LIS) word for each Ethernet interface to determine whether the Redundant IP address is active at that interface. SVC\_REQ 55 does not affect Ethernet interfaces that are not configured for redundant IP operation.

### 1.4.4 SRTP Server Operation in a Redundancy System

Only the active unit maintains SRTP Server connections *at the Redundant IP address* and can respond to SRTP requests. The backup unit does not respond to the Redundant IP address. When an Ethernet interface changes from active to backup state, it takes down all SRTP Server connections and their underlying TCP connections that use the Redundant IP address.

Both the active and backup units maintain SRTP Server connections at the direct IP address for network communication with the programmer. Other remote hosts should use the Redundant IP address when communicating to a redundant system. Existing SRTP Server connections at the direct IP address are not disturbed when the Ethernet interface switches between active and backup states.

### 1.4.5 SRTP Client Operation in a Redundancy System

Only the active unit establishes and maintains SRTP Client connections (channels). The backup unit does not initiate any SRTP Client operations. If SRTP Client operations are attempted, a COMMREQ error status is returned to the local logic program. When the Ethernet interface changes from active to backup state, it takes down all SRTP Client connections and their underlying TCP connections.

Because it can take some time to take down a TCP connection, the redundant system should reserve a spare SRTP Client connection for each connection using the Redundant IP address. That will prevent temporary resource problems when establishing new SRTP Client connections to the new active unit while the previous connections to the old active unit are being taken down.



## 1.4.6 Modbus TCP Server Operation in a Redundancy System

Only the active unit maintains Modbus TCP Server connections at the Redundant IP address and can respond to Modbus TCP requests. The backup unit does not respond to the Redundant IP address. When an Ethernet interface changes from active to backup state, it takes down all Modbus TCP Server connections and their underlying TCP connections that use the Redundant IP address.

Remote hosts should use the Redundant IP address when communicating to a redundant system. Existing Modbus TCP Server connections at the direct IP address are not disturbed when the Ethernet interface switches between active and backup states.

## 1.4.7 Modbus TCP Client Operation in a Redundancy System

Only the active unit establishes and maintains Modbus TCP Client connections (channels). The backup unit does not initiate any Modbus TCP Client operations. If Modbus TCP Client operations are attempted, a COMMREQ error status is returned to the local logic program. When the Ethernet interface changes from active to backup state, it takes down all Modbus TCP Client connections and their underlying TCP connections.

Because it can take some time to take down a TCP connection, the redundant system should reserve a spare Modbus TCP Client connection for each connection using the Redundant IP address. That will prevent temporary resource problems when establishing new Modbus TCP Client connections to the new active unit while the previous connections to the old active unit are being taken down.

## 1.4.8 EGD Class 1 (Production & Consumption) in a Redundancy System

The active unit produces Ethernet Global Data exchanges to the network. The backup unit produces only the EGD exchanges for which Produce in Backup Mode is enabled. When the active Ethernet interface changes to the backup, it stops the production of all EGD exchanges.

When configured for Redundant IP operation, the active and backup Ethernet interfaces should also be configured to consume EGD exchanges via multicast host groups or the local subnet broadcast address. This permits both the active and backup units to receive the latest data from the network. Emerson does not recommend Unicast operation as the backup unit as it will not consume any unicast exchanges at the Redundant IP address.

The IP address the exchange is produced from should not typically matter to the Consumer. If the consumer is only capable of listening to communications from the Redundant IP address, the **Produce EGD on Redundant IP** setting in the Ethernet interfaces Settings tab should be set to True.

## 1.4.9 EGD Class 2 Commands in a Redundancy System

Remote hosts should use the Redundant IP address when communicating to a redundant system. Only the active unit responds to EGD commands. The backup unit does not respond to the Redundant IP address. When the active Ethernet interface changes to the backup, any in-process EGD commands over the Redundant IP address are abandoned.

When configured for Redundant IP operation, only the active unit sends EGD commands on the network. If the backup unit tries to initiate any EGD commands, a COMMREQ error status is returned to its application program. When the active Ethernet interface changes to the backup, any EGD commands in the process are ended. Issuing EGD commands to the direct IP address is not recommended; both the active and backup units will respond to EGD commands received at the direct IP address.

## 1.4.10 Web Server Operation in a Redundancy System

Only the active unit processes Web server requests at the Redundant IP address and responds to Web page requests. The backup unit does not respond to the Redundant IP address. When the active Ethernet interface changes to the backup, all Web server connections and their underlying TCP connections are disrupted. The Web server maintains its underlying TCP connection only long enough to process:

- A new Web page request
- A new TCP connection opened, used, or closed for each subsequent Web page display or update.

The Redundant IP address is transparent to the Web remote browser unless a Web page change or update is requested during the redundancy role switch. Any Web page request in process over the Redundant IP when a role switch occurs is terminated.

Although not recommended, the remote browser may issue Web server requests to the direct IP address. Both the active and backup units respond to Web server requests received at the direct IP address. Remote Web browsers are expected to use the Redundant IP address when communicating to a redundant system.

## 1.4.11 FTP Operation in a Redundancy System<sup>2</sup>

FTP operations can transfer setup and configuration data to the Ethernet interface. Using FTP operations for communication with the actual PLC application is not recommended. FTP operations should only be performed using the direct IP address.

## 1.4.12 SNTP Operation in a Redundancy System

A PACSystems Ethernet interface can operate as an SNTP client-only, which enables the interface to only receive broadcast time messages from an SNTP Server on the network. SNTP operation is unaffected by the current Ethernet redundancy state or by redundancy role switches.

---

<sup>2</sup> FTP is not supported by ETM001-Kxxx  
Introduction

## 1.4.13 Remote Station Manager Operation in a Redundancy System

The remote Station Manager should respond to the direct IP address whether the unit is active or backup or whether the Redundant IP is configured.

Only the active unit responds to remote Station Manager commands at the Redundant IP address. The backup unit does not respond to the Redundant IP address. (Station Manager responses from the Redundant IP address can be misleading because it is difficult to determine which Ethernet interface is responding.)

## 1.4.14 IP Address Configuration in a Redundancy System

Redundancy systems should explicitly configure both the direct IP address and the Redundant IP address. Do not set up the direct IP address via BOOTP.

The Redundant IP address must be configured on the same local sub-network as the direct IP address and gateway IP address (if used).



## Section 2 Installation and Start-up: RX3i/RSTi-EP Embedded Interface

The RX3i CPUs with CPExxx designation (CPE302, CPE305, CPE310, CPE330, and CPE400), the CPL410 and RSTi-EP CPE100/CPE115 provide an embedded Ethernet interface for programmer communications. This section describes user features and provides basic installation and startup procedures for this interface.

- Ethernet Interface Controls and Indicators
- Module Installation
- Connection to a 10Base-T/100Base-TX Network (all CPExxx) or a 1000Base-T (CPE330, CPE400, and CPL410 only)
- Pinging TCP/IP Ethernet interfaces on the Network









## 2.1 RX3i/RSTi-EP Embedded Ethernet Interface Indicators

Many of the Ethernet interfaces feature Ethernet ports with two LED indicators, *100* and *LINK*. The *100* LED indicates the network data speed (10 or 100 Mb/sec). This LED is lit if the network connection at that network port is 100 Mbps.







The *LINK* LED indicates the network link status and activity. This LED is lit when the link is physically connected. It blinks when traffic is detected at that network port.

### 2.1.1 Ethernet Port LEDs Operation






#### CPE302/CPE305/CPE310 Ethernet LED Operation

LED	LED State	Ethernet Port State
	 On  Blinking  Off	
100	 On, Green	The network data speed is 100 Mbps.
	 Off	The network data speed is 10 Mbps.
LINK	 On, Amber	The link is physically connected.
	 Blinking, Amber	Traffic is detected at the port.
	 Off	The Ethernet port is not physically connected.







#### CPE330 Ethernet LED Operation

LED	LED State	Operating State
LINK (upper)	 On Green	The corresponding link is physically connected.
	 Blinking Green	Traffic is detected at the corresponding port.
	 Off	No connection was detected at the corresponding port.
1 Gbps (lower)	 On Amber (LAN1) or  On Green (LAN2)	Corresponding network data speed is 1 Gbps.
	 Off	Corresponding network data speed is 100 Mbps or 10 Mbps.







## CPE400/CPL410 Front Panel Ethernet LED Operation (LAN1, LAN2, LAN3)

LED	LED State	Operating State
Link Status (upper)	 On Green	The corresponding link has been established.
	 Blinking Green	Traffic is detected at the corresponding port.
	 Off	No connection was established at the corresponding port.
Link Speed (lower)	 On Green	Corresponding data speed is 1 Gbps or 100 Mbps.
	 Off	Corresponding network data speed is 10 Mbps






## CPE400 Underside Ethernet LED Operation (EFA)

LED	LED State	Operating State
Link Status (upper)	 On Green	The corresponding link has been established.
	 Blinking Green	Traffic is detected at the corresponding port.
	 Off	No connection was established at the corresponding port.
Link Speed (lower)	 On Green	Corresponding network data speed is 1 Gbps.
	 On Yellow	EFA port only: network data speed is 100 Mbps
	 Off	Corresponding network data speed is 10 Mbps

## CPL410 Underside Ethernet LED Operation (ETH)

LED	LED State	Operating State
Link Status (upper)	 On Green	The corresponding link has been established.
	 Blinking Green	Traffic is detected at the corresponding port.
	 Off	No connection was established at the corresponding port.
Link Speed (lower)	 On Green	Corresponding network data speed is 1 Gbps.
	 On Yellow	GPOS port only: network data speed is 100 Mbps
	 Off	Corresponding network data speed is 10 Mbps

## CPE100/CPE115 Ethernet LED Operation (LAN1, LAN2)

LED	LED State	Operating State
Link Speed (upper)	 On Amber	Corresponding data speed is 100 Mbps.
	 Off	Corresponding network data speed is 10 Mbps
Link Status (lower)	 On Green	The corresponding link has been established.
	 Blinking Green	Traffic is detected at the corresponding port.
	 Off	No connection was established at the corresponding port.



## 2.1.2 Module Installation

For general information about CPU module and system, installation refers to the *PACSystems RX3i System Manual*, GFK-2314 Sections 2 & 3.

For the RSTi-EP CPU model, refer to *RSTi-EP System Manual*, GFK-2958D, or later.

## 2.2 Ethernet Port Connector

The RX3i CPE302/CPE305 and CPE310 CPUs provide a T/100Base-TX Ethernet network port connector. When a configured as a CPU320, Ethernet properties cannot be configured. However, the embedded Ethernet ports may with the default IP Addresses.



10Base-CPE330 is  
be used

### 2.2.1 Connection to a 10Base-T/100Base-TX Network

Either shielded or unshielded twisted pair cable may be attached to an Ethernet port. The 10Base-T/100Base-TX twisted pair cable must meet the applicable IEEE 802 standards. Category 5 cable is required for 100Base-TX operation.

The Ethernet port automatically senses the speed (10 Mbps or 100 Mbps), duplex mode (half-duplex or full-duplex), and cable configuration (straight-through or crossover) attached to it with no intervention required.

### 2.2.2 10Base-T/100Base-TX Port Pinouts

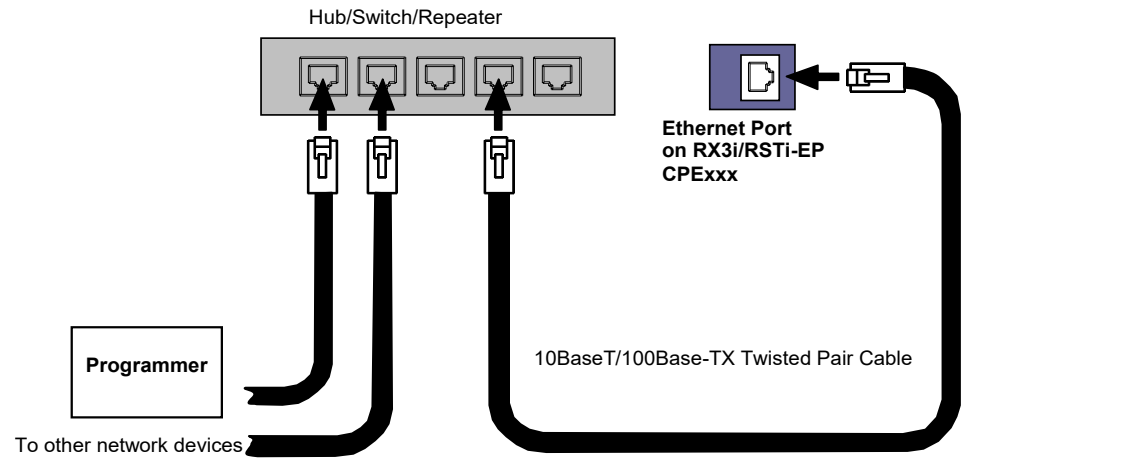
Pin Number <sup>3</sup>	Signal	Description
1	TD+	Transmit Data +
2	TD-	Transmit Data -
3	RD+	Receive Data +
4	NC	No connection
5	NC	No connection
6	RD-	Receive Data -
7	NC	No connection
8	NC	No connection

**Note:** Pin assignments are provided for troubleshooting purposes only. 10Base-T/100Base-TX cables are readily available from commercial distributors. We recommend purchasing rather than making 10Base-T/100Base-TX cables.

The programmer is connected to the Ethernet interface through a 10Base-T or 100Base-TX network.

<sup>3</sup> Pin 1 is at the bottom right of the Station Manager port connector as viewed from the front of the module.  
*Installation and Start-up: RX3i/RSTi-EP Embedded Interface*

Figure 5: Ethernet Cable Routing



## 2.3 Pinging TCP/IP Ethernet interfaces on the Network

PING (Packet InterNet Grouper) is the name of a program used on TCP/IP networks to test the reachability of destinations by sending them an ICMP echo request message and waiting for a reply.

You should ping each installed Ethernet interface. When the Ethernet interface responds to the ping, it verifies that the interface is operational and configured properly. Specifically, it verifies that acceptable TCP/IP configuration information has been downloaded to the interface.

For configuration details, including setting an initial IP address, refer to Section 4, *Configuration*.

### 2.3.1 Determining if an IP Address is Already Being Used

*It is very important not to duplicate IP addresses.* To determine if another node on the network is using the same IP address:

1. Disconnect your Ethernet interface from the LAN.
2. Ping the disconnected interface's IP address. If you get an answer to the ping, the chosen IP address is already in use by another node. You *must* correct this situation by assigning a unique IP address.

**Note:** This method does not guarantee that an IP address is not duplicated. It will not detect a device that is configured with the same IP address if it is temporarily off the network.



# Section 3 Installation and Start-up: Ethernet Module Interfaces

This chapter describes the features and basic installation procedures for Ethernet module interfaces (ETM001-Jx and ETM001-Kxxx Ethernet interface Controls and Indicators

- Ethernet LEDs
- Ethernet Restart Pushbutton
- Front Panel Port
- Ethernet Port Connections
- Module Installation
  - RX3i Rack-Based Ethernet Interface Modules
- Ethernet Port Connectors
  - Embedded Switch
  - Connection to a 10Base-T/100Base-TX/1000Base-T Network
- Station Manager Port
- Verifying Proper Power-Up of the Ethernet interface After Configuration
- Pinging TCP/IP Ethernet interfaces on the Network

### 3.1 Ethernet Module Interface Characteristics

There are two versions of the RX3i rack-based Ethernet module. Please note the differences in the table below:

ETM001-Jx	ETM001-Kxxx
<p>The ETM001-Jx Ethernet module provides:</p> <ul style="list-style-type: none"> <li>• An Ethernet 10Base-T/100Base-TX interface</li> <li>• Two RJ-45 Ethernet ports. Either or both of these ports can be attached to other Ethernet devices. Each port automatically senses the data rate (10 Mbps or 100 Mbps), duplex (half-duplex or full-duplex), and cabling arrangement (straight-through or crossover) of the attached link.</li> <li>• An embedded autodetect/auto-switch Ethernet switch, which provides a means to switch Ethernet data and allow daisy-chaining of Ethernet cabling, and provides a method to automatically detect Ethernet cable wire crossover.</li> </ul>	<p>The ETM001-Kxxx Ethernet module provides:</p> <ul style="list-style-type: none"> <li>• An Ethernet 10Base-T/100Base-TX/1000Base-T interface</li> <li>• Two RJ45 Ethernet ports are located on the underside of the module. Either or both ports can attach to other Ethernet devices. Each port automatically senses the data rate (10 Mbps/ 100 Mbps / 1000 Mbps), duplex (half-duplex or full-duplex), and cabling arrangement (straight-through or crossover) of the attached link.</li> <li>• An embedded autodetect/auto-switch Ethernet switch, which provides a means to switch Ethernet data and allow daisy-chaining of Ethernet cabling, and provides a method to automatically detect Ethernet cable wire crossover.</li> </ul>

Figure 6: ETM001-Jx Faceplate

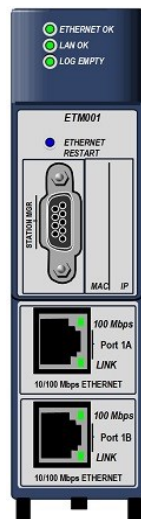


Figure 7: ETM001-Kxxx Faceplate



### 3.1.1 Front Panel Port

The revised Ethernet module ETM001-Kxxx has been updated with an RJ45 port. The Ethernet port supports Station Manager over IP. The default IP settings on the front panel port are 10.10.0.100 subnet 255.255.255.0, gateway 0.0.0.0. Note: The Front panel port does not support PACSAnalyzer only the Station Manager.

### 3.1.2 Ethernet Port Connections

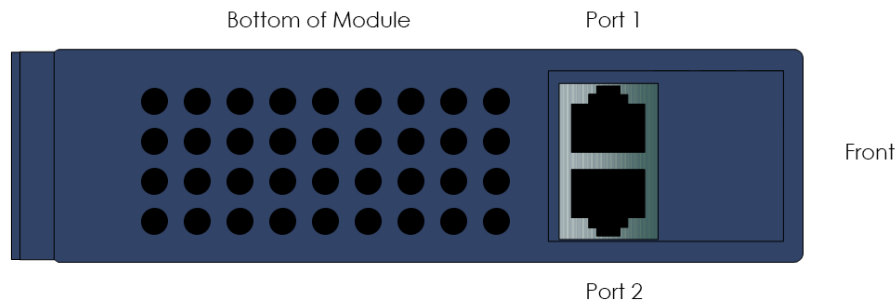
Each port on an ETM001 or ETM001-Kxxx operates independently, so devices that operate at different speeds and/or duplex modes may be attached to the ports. By default, all ports (even empty) are set for Automatic, which enables auto-negotiation for the widest range of options supported by the port. The port connection speed can be manually configured for slower speeds (10/100 Mbps) on the LAN1 tab in PME.

The ports can auto-negotiate to 10/100/1000 Mbps at full or half-duplex. The speed can be limited through Hardware Configuration (ETM001-Kxxx) or AUP settings (ETM001) to 10 Mbps or 100 Mbps. 1000 Mbps operation is only available with auto-negotiation on an ETM001-Kxxx.

On the ETM001-Kxxx, the half-duplex operation does not resend a packet that experiences a collision at the hardware level. A collision is reported as a send or receives an error and TCP retries or other retry mechanisms must resend the data.

Embedded switches have limited memory to store packets. If there is a speed difference between incoming traffic and outgoing traffic, there is an increased probability of network traffic being dropped due to resource limitations. For example, if Port 1 has an established 1000 Mbps link and Port 2 has an established 100 Mbps link, then traffic traveling in from Port 1 and out Port 2 must be buffered since traffic is leaving at a maximum rate of 10 times slower than the maximum incoming rate. As the network loading increases, the likelihood of data being dropped increases. Dropped packets are indicated in the Station Manager Tallies. If there are significantly dropped packets due to a speed mismatch, it may be useful to limit the speed at the ETM001(-Kxxx) to match the larger network. Speed changes in a system may be handled better with switches with more packet buffer memory.

**Figure 8: Ethernet Port Connectors on IC695ETM001-Kxxx**



### 3.1.3 LEDs on the RX3i Ethernet Interface Module

LEDs provide an immediate visual indication of the operational state of the Ethernet module and port link status. The LEDs and their operation are described in Chapter 2.1.1, *Ethernet Port LEDs Operation*.

### 3.1.4 Ethernet LEDs

Ethernet modules have LEDs to indicate the state and status of the Ethernet interface:

#### **ETM001-Jx**

The Ethernet module's Ethernet ports have two LEDs. The bottom LED is the link speed LED. This LED is on for 100 Mbps and off for all other speeds. The top LED is the link/activity LED. This will be on when there is a link at any speed. The top led will blink when there is inbound or outbound traffic.

#### **ETM001-Kxxx**

The Ethernet module's Ethernet ports will not have LEDs on the ports. Ethernet traffic speed is indicated on the module's front panel display, with discrete LEDs to indicate 10/100/1000 Mbps traffic.

#### **LAN OK LED Operation**

The LAN OK LED indicates access to the Ethernet network. During normal operation, the LAN OK LED blinks when data is being sent or received over the network directed to or from the Ethernet interface. It remains on when the Ethernet interface is not actively accessing the network, but the Ethernet physical interface is available and one or both of the Ethernet ports is operational.

It is off otherwise unless a firmware update is occurring.

#### **LOG EMPTY LED Operation**

The LOG EMPTY LED indicates the condition of the Ethernet interface in normal operational mode. If the LOG EMPTY LED is off, an event has been entered into the exception log and is available for viewing via the Station Manager interface. The LOG EMPTY LED is on during normal operation when no events are logged.

In the other states, the LOG EMPTY LED is either off or blinking and helps define the operational state of the module. For more information on LED behavior, refer to Section 12 Diagnostics.

#### **Ethernet OK LED Operation**

The Ethernet OK LED indicates whether the module can perform the normal operation. This LED is on for normal operation and flashing for all other operations. When hardware or unrecoverable runtime failure occurs, the ETM001-Jx will blink a two-digit error code identifying the failure. The ETM001-Kxxx will blink a four-digit code. For assistance troubleshooting errors, check Section 12.4, *ETHERNET OK/OK LED Blink Codes for Hardware Failures (ETM001-Jx)*. For assistance troubleshooting ETM001-Kxxx, please call Technical Support.

## Ethernet Port LEDs Operation (100Mb and Link/Activity)

The face of the Ethernet module is equipped with LEDs to indicate a physical connection at the network port and network traffic speed.

Ethernet interface module	Link/Activity Location	Link/Activity Behavior
IC695ETM001	Two Ethernet ports (1A and 1B) with two LED indicators (“100” and “LINK”) are located at the front of the interface module.	<p><b>100</b> – LED indicates the network data speed (10 or 100 Mbps). When the LED state is ON, the network connection at that network port is 100 Mbps.</p> <p><b>Link</b> – LED indicates network link status and activity. When the LED state is ON, the link is physically connected. When the LED blinks, traffic is detected at that network port. <i>(Traffic at the port does not indicate traffic is present at the Ethernet interface. Traffic may be introduced between ports of the switch.)</i></p>
IC695ETM001-Kxxx	<p>Six single color green LEDs are located on the face of the interface module. The LEDs are labeled 10, 100, 1000 for each of the two Ethernet ports.</p> <p>Two Ethernet ports (1 and 2) are located on the underside of the interface module.</p>	<p>When the LED state is ON, the link is physically connected. When the LED blinks, traffic is detected at that network port.</p> <p>The LED label indicates network traffic speed.</p>



### 3.1.5 Restart/Reset Pushbutton Operation

For PACSystems Ethernet interfaces, an Ethernet restart occurs when the restart/reset pushbutton is pressed and released. The duration that the restart/reset pushbutton is pressed determines the operation after the restart occurs.

If the Ethernet interface uses any optional Ethernet plug-in applications, these applications are ordinarily started upon each power-up or restart. To restart the Ethernet interface without starting any Ethernet plug-in applications, press and hold the Ethernet restart/reset pushbutton between five and 10 seconds.

If the Ethernet interface can restart into firmware update operation, press and hold the Ethernet restart pushbutton for more than 10 seconds. This is typically done during troubleshooting to bypass possibly invalid firmware and allow valid firmware to be loaded using WinLoader.

If the Ethernet interface uses the reset button to set a fixed temporary IP address, press and hold the reset button for more than five seconds. Entering the *IP Setup Mode* will display a light pattern on the front panel's traffic speed LEDs.

Pushbutton-controlled restart operations are listed below, with the LED indications for each:

Restart Operation	ETM001-Jx – Press and Hold Restart Functionality	ETM001-Kxxx - Press and Hold Reset Functionality
Restart the Ethernet interface normally, and start any optional Ethernet plug-in applications that are being used.	Less than five seconds	Less than five seconds
Restart the Ethernet interface without starting any Ethernet plug-in applications.	Five to 10 seconds	N/A
Put into <i>IP Setup Mode</i> to set an initial IP address.	N/A	More than five seconds
Restart the Ethernet interface into firmware update operation.	More than 10 seconds	After a restart, press and hold the reset button on power-up (until the top three LEDs flash)

When forced into firmware update operation, but before the firmware update begins, press the Ethernet Restart/Reset pushbutton again to exit the firmware update mode and restart with the existing firmware. Once the firmware update begins, the existing firmware is erased and the Ethernet Restart/Reset pushbutton is disabled until the firmware update is complete.

If the firmware update mode was entered mistakenly, simply remove and replace the Ethernet module from the backplane.

#### Setting a Temporary IP Address with the Reset Button

To use the Reset Button to set a temporary IP address, refer to the Section entitled *Assigning a Temporary IP Address Using the Programming Software*.

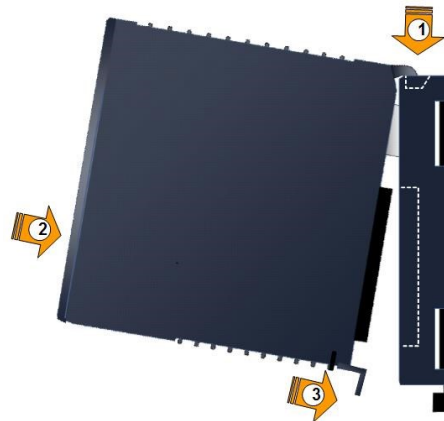
## 3.2 Ethernet Module Installation

For general information about the module and system installation, or if the installation requires CE Mark compliance, refer to the *PACSystems RX3i System Manual*, GFK-2314.

### 3.2.1 Module Installation

1. Holding the module firmly, align the module with the correct slot and connector.
2. Engage the module's rear pivot hook in the notch on the top of the backplane (1).
3. Swing the module down (2) until the module's connector engages the backplane's backplane connector.
4. ETM001-Kxxx: Secure the bottom of the module to the backplane using the machine screws provided with the module (3). ETM001: Secure bottom with the spring-loaded latch mechanism.

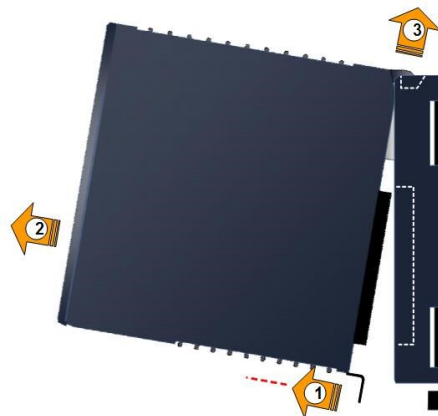
Figure 9: Install Module into RX3i Backplane



### 3.2.2 Module Removal

1. The ETM001-Jx and ETM001-Kxxx may be removed from the RX3i rack with the power supplied to the rack.
2. ETM001-Kxxx: Loosen the screws at the bottom of the module (1). ETM001: Release the spring-loaded latch mechanism on the underside of the module.
3. Pivot the module upward until its connector is out of the backplane (2).
4. Lift the module up and away from the backplane to disengage the pivot hook (3).

Figure 10: Remove Module from RX3i Backplane



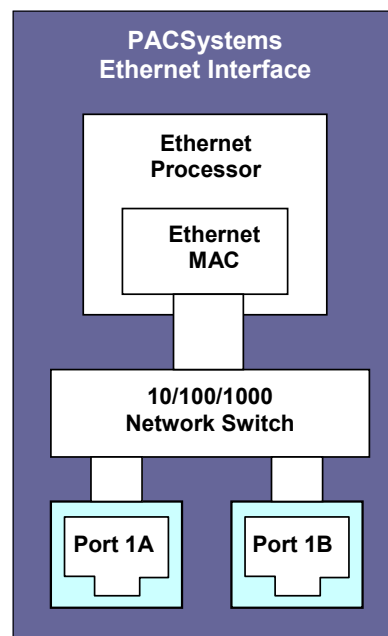
## 3.3 Ethernet Port Connectors

The Ethernet interface module has two Ethernet port connectors, each of which supports both 10Base-T, 100Base-TX, and 1000Base-T operation using either full-duplex or half-duplex operation. These 8-pin RJ45 connectors are used to connect the Ethernet interface to a hub, repeater, switch, or another Ethernet device.

### 3.3.1 Embedded Switch

The two Ethernet port connectors are controlled by an embedded network switch in the module. The module has only one interface to the network (one Ethernet address and one IP address).

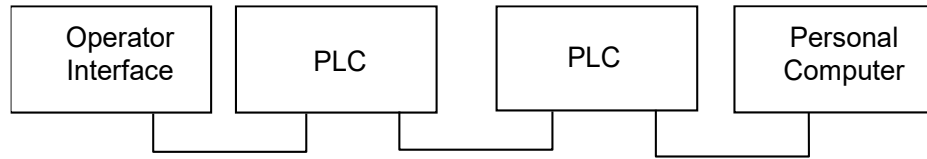
**Figure 11: Diagram of Embedded Ethernet Switch<sup>4</sup>**



<sup>4</sup> ETM001-jx is only capable of 100 Mbps speeds.  
*Installation and Start-up: Ethernet Module Interfaces*

For simple installations, the embedded switch allows devices to be connected without additional components.

**Figure 12: System Diagram: Ethernet Routing Using Embedded Switch**



Use extra care when placing PLCs in a daisy-chain configuration without additional components. Power loss or reset at an Ethernet interface can cause loss of communication to any devices downstream from that Ethernet interface in the daisy chain. Restarting the Ethernet interface (via the Ethernet Restart/RESET pushbutton, for example) disrupts daisy chain communication.

Each switch port defaults to the correct link speed and duplex mode for the device connected to the other end of the link. Each port operates independently; devices at two different speeds and/or duplex modes may be attached to the two ports. By default, each port will automatically detect the attached cable and will work properly with either straight-through or crossover cables.

### CAUTION

The two Ethernet ports on the Ethernet interface must not be connected, directly or indirectly, to the same device. The connections in an Ethernet network based on twisted pair cabling must form a tree and not a ring. Failure to follow this caution may cause duplication of packets and cause a network overload.

### CAUTION

The IEEE 802.3 standard strongly discourages the manual configuration of duplex mode for a port (as would be possible using Advanced User Parameters). Before manually configuring duplex mode for an Ethernet interface port using advanced user parameters (AUP), be sure that you know the characteristics of the link partner and are aware of the consequences of your selection. Setting both the speed and duplex AUPs on an IC698 Ethernet interface port will disable the port's auto-negotiation function. If its link partner is not similarly manually configured, this can result in the link partner concluding an incorrect duplex mode. Per the IEEE standard: "Connecting incompatible DTE/MAU combinations such as full-duplex mode DTE to a half-duplex mode MAU, or a full-duplex station (DTE or MAU) to a repeater or other half-duplex network, can lead to severe network performance degradation, increased collisions, late collisions, CRC errors, and undetected data corruption."

**Note:** If both speed and duplex mode of an Ethernet interface port is forced using the Advanced User Parameters file, that port will no longer perform automatic cable detection. This means that if you have the Ethernet interface port connected to an external switch or hub port you must use a crossover cable. If you have the Ethernet interface port connected to the uplink port on an

external switch or hub, or if you have the Ethernet interface port directly connected to another Ethernet device, you must use a normal cable.

### 3.3.2 Connection to a 10Base-T/100Base-TX/1000Base-T Network

Either shielded or unshielded twisted pair cable may be attached to a port. The 10Base-T/100Base-TX/1000Base-T twisted pair cables must meet the applicable IEEE 802 standards. Category 5 is supported for 100Base-TX operation. Category 5e or C6 is recommended for 1000Base-T speeds.

Each Ethernet port automatically senses whether it is connected to a 10Base-T, 100Base-TX, or 1000Base-T network, half-duplex or full-duplex. (The automatic negotiation of speed and/or duplex mode can be explicitly overridden using Advanced User Parameter settings).

#### 10Base-T/100Base-TX/1000Base-T Port Pinouts

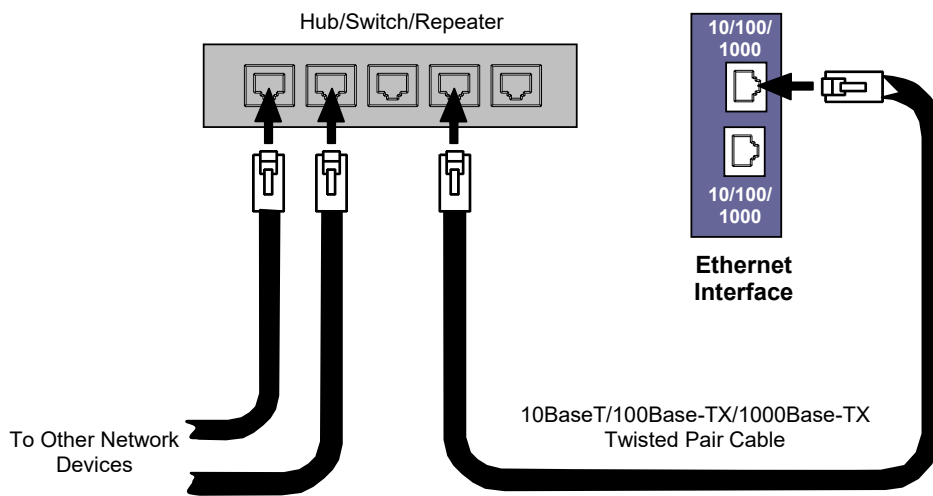
Pin Number <sup>3</sup>	Signal	Description
1	BI_DA+	Bi-directional pair A +
2	BI_DA-	Bi-directional pair A -
3	BI_DB+	Bi-directional pair B +
4	BI_DC+	Bi-directional pair C +
5	BI_DC-	Bi-directional pair C -
6	BI_DB-	Bi-directional pair B -
7	BI_DD+	Bi-directional pair D +
8	BI_DD-	Bi-directional pair D -

**Note:** Pin assignments are provided for troubleshooting purposes only. 10Base-T/100Base-TX/1000Base-T/ cables are readily available from commercial distributors. Purchasing commercial cables are recommended rather than making cables for this application.

## Connection Using a Hub/Switch/Repeater

The connection of the Ethernet interface to a 10Base-T, 100Base-TX, or 1000Base-T network is shown below.

**Figure 13: Connection Using Hub/Switch/Repeater**

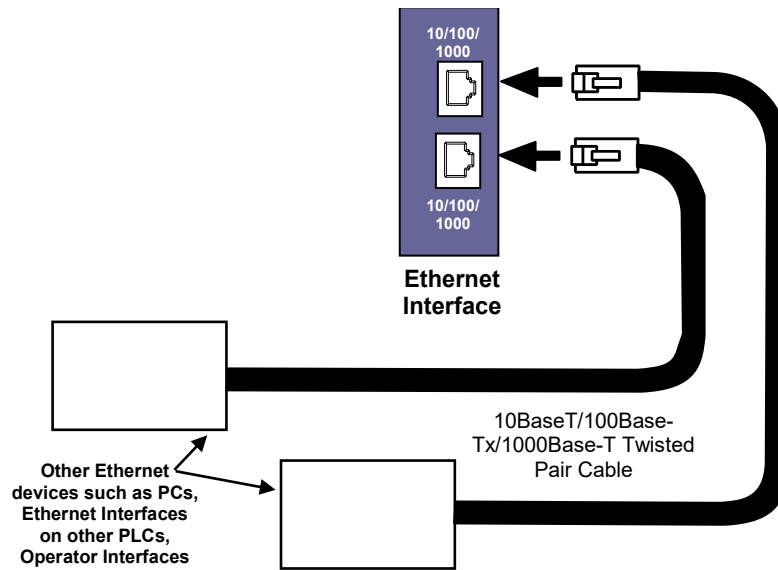


**Note:** Care must be taken with the use of active network control devices, such as managed switches. If a device inserts excessive latency, especially regarding the ARP protocol, produced EGD exchanges may generate PLC Fault Table entries indicating the loss of a consumer when the PLC transitions from STOP to RUN. EGD data will be successfully transferred after an initial delay.

## Direct Connection to the PACSystems Ethernet interface

Connection of Ethernet devices directly to the Ethernet interface is shown below:

Figure 14: Direct Connection to the Embedded Ethernet Ports



## 3.4 Station Manager Port

The ETM001-jx Ethernet interface module provides a dedicated RS-232 serial port for local Station Manager use. The nine-pin D-sub connector accepts a standard straight-through nine-pin RS-232 serial cable to connect to a standard AT-style RS-232 port.

The following cable is available: Cable, CPU Programming - IC200CBL001

**Note:** RX3i IC695ETM001-Kxxx replaced the Station Manager serial port for an Ethernet port.

### 3.4.1 Port Settings

The serial (COM) port of the terminal or computer that is connected to the Ethernet interface must use the same communications parameters as the Ethernet interface.

The default values for the Station Manager port are 9600 bps, 8 bits, no parity, and 1 stop bit. If the Ethernet interface is configured with default values for this port, or the Ethernet interface has not been configured, use these default values. If the Ethernet interface is configured with non-default values for this port, use those values for the serial port settings of the terminal or computer.






## Station Manager (RS-232) Port Pin Assignment

Pin No <sup>3</sup>	Signal	Direction	Description
1	DCD	IN	Data Carrier Detect
2	TX	OUT	Transmit Data
3	RX	IN	Receive Data
4	DSR	IN	Data Set Ready
5	GND		Signal Ground
6	DTR	OUT	Data Terminal Ready
7	CTS	IN	Clear to Send
8	RTS	OUT	Ready to Send
9	RI	IN	Ring Indicator

### 3.5 Verifying Proper Power-Up of the Ethernet Interface After Configuration

After configuring the interface as described in Section 4, turn the power OFF to the CPU for 3–5 seconds, then turn the power back ON. This starts a series of diagnostic tests. The ETHERNETOK LED will blink indicating the progress of power-up.

The Ethernet LEDs will have the following pattern upon successful power-up. At the time of this publication, the Ethernet interface is fully operational and online.

Ethernet Interface Online	LED Label (ETM001-Jx)	LED Label (ETM001-Kxxx)
 On	ETHERNET OK	OK
   On, blinking, or off depending on network activity	LAN OK	LAN OK
 On	LOG EMPTY	LOG EMPTY

If a problem is detected during power-up, the Ethernet interface may not transition directly to the operational state. If the interface does not transition to operational, refer to *Section 12*, Diagnostics for corrective action.



## 3.6 Pinging TCP/IP Ethernet interfaces on the Network

Packet InterNet Grouper (PING) is the name of a program used on TCP/IP networks to test the reachability of destinations by sending them an ICMP echo request message and waiting for a reply. Most nodes on TCP/IP networks, including the PACSystems Ethernet interface, implement a *ping* command.

You should *ping* each installed Ethernet interface. When the Ethernet interface responds to the *ping*, it verifies that the interface is operational and configured properly. Specifically, a *ping* verifies that acceptable TCP/IP configuration information has been downloaded to the interface.

For configuration details, including setting an initial IP address, refer to Section 4, *Configuration*.

### 3.6.1 Determining if an IP Address is Already Being Used

*It is very important not to duplicate IP addresses.* To determine if another node on the network is using the same IP address:

1. Disconnect your Ethernet interface from the LAN.
2. Ping the disconnected interface's IP address. If you get an answer to the ping, the chosen IP address is already in use by another node. You *must* correct this situation by assigning unique IP addresses.

**Note:** This method does not guarantee that an IP address is not duplicated. This method will not detect a device that is configured with the same IP address if it is temporarily off the network.

## 3.7 Ethernet Plug-in Applications<sup>5</sup>

Ethernet interface supports the use of additional firmware images called *Ethernet plug-in applications*, which may implement additional communication protocols. Up to three Ethernet plug-in applications can be loaded into the Ethernet interface along with the Ethernet firmware via the WinLoader utility. Each plug-in application is identified by a number (1-3). Once loaded, each Ethernet plug-in application is stored in non-volatile memory where it is preserved until it is either overwritten by WinLoader to create another Ethernet plug-in application with the same number, or it is explicitly deleted via the *pluginapp* Station Manager command. For more information on Station Manager commands, see the *PACSystems TCP/IP Ethernet Communications Station Manager*

All Ethernet plug-in applications are started during normal Ethernet power-up or restart. During troubleshooting, the Ethernet Restart/Reset pushbutton may be used to startup the Ethernet interface without the plug-in applications (refer to the Section entitled *Restart/Reset Pushbutton Operation*).

The functional operation, PLC interfaces, and Station Manager support for each Ethernet plug-in application are supplied separately from this user manual.

---

<sup>5</sup> ETM001-Kxxx does not support Ethernet Plug-in Applications  
*Installation and Start-up: Ethernet Module Interfaces*



# Section 4 Configuration

Before you can use the Ethernet interface, you must configure it using PME Logic Developer-PLC software.

This chapter includes configuration information for:

- RX3i/RSTi-EP Embedded Ethernet interface
  - Ethernet Configuration Data
  - Initial IP Address Assignment
  - Configuring the Ethernet interface Parameters
  
- RX3i rack-based Ethernet interfaces
  - Ethernet Configuration Data
  - Initial IP Address Assignment
  - Configuring Ethernet interface Parameters
  - Configuring Ethernet Global Data

## 4.1 RX3i/RSTi-EP Embedded Ethernet Interfaces

### 4.1.1 Ethernet Configuration Data

The PACSystems PLC is configured exclusively by the PME Logic Developer-PLC Programmer. For the initial Programmer connection, an initial IP address must be manually assigned to the Ethernet interface as described in this chapter. The PACSystems PLC does not support autoconfiguration.

#### Generating/Storing/Loading the Configuration

The RX3i/RSTi-EP embedded Ethernet interface is configured as a sub-module of the CPE module. The RX3i/RSTi-EP Embedded Ethernet interface uses an Ethernet configuration and an optional Advanced User Parameter (AUP) configuration. Both configurations are generated at the Programmer to be stored on the PLC as part of the *Hardware Configuration Store* sequence. The configuration may be loaded from the PLC to the Programmer as part of the Configuration Load sequence. The optional AUP file must be manually generated with a text editor and then imported into the Programmer. (See *Configuring Advanced User Parameters* for details.) Once the configuration is stored in the PLC, the CPU maintains the Ethernet configuration data in non-volatile memory over power cycles.

The following CPE/CPU/CPLs do not support an AUP file: CPE330, CPE400, CPL410, CPE100, or CPE115. The configurable AUP parameters for these CPUs are part of the configuration for the embedded Ethernet interface. For the CPE330, CPE400 and CPL410, this configuration interface is available in PME 8.60 SIM5 or later. For CPE100/CPE115, this configuration interface is available in PME 9.50 SIM 2 or later.

#### Backup Configuration Data

The RX3i/RSTi-EP embedded Ethernet interface maintains a backup copy of the most recent Ethernet configuration and AUP configuration in non-volatile memory. A PLC Configuration Clear does not affect this backup Ethernet configuration data. When the configuration was not stored from the Programmer or the PLC configuration has been cleared, the Ethernet interface uses its backup configuration.

#### Locally Edited Configuration Data

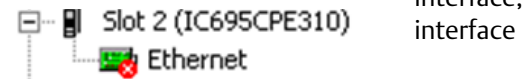
The embedded Ethernet configuration and AUP configuration cannot be locally edited via Station Manager. All configuration changes must be performed via the Programmer.

## 4.1.2 Initial IP Address Assignment

The RX3i/RSTi-EP embedded Ethernet interface comes from the factory with a default IP address (192.168.0.100). This address is intended only for the initial connection to complete the configuration and *must* be changed before connecting to the Ethernet network. The IP address must be selected for proper operation with your network and application. See your network administrator for the proper IP address value.

- Using PAC Machine Edition software, configure the Ethernet-enabled CPU in an RX3i target (or) configure the CPE100/CPE115 in an RSTi-EP target and assign a new IP address to the embedded Ethernet interface:
- To configure the embedded Ethernet interface, expand the CPU slot to display the Ethernet interface (Figure 15).
- Right-click the Ethernet interface to display its parameters: IP Address, Subnet Mask, and Gateway IP Address. Consult your network administrator for the proper values for these parameters.

**Figure 15: Expand CPU Slot to Display Ethernet Node**



Go online with the target and download the configuration. You can use one of the following methods for the initial connection to the CPE3xx:

**Method 1:** Through the embedded Ethernet port, using the factory-loaded default IP address (192.168.0.100). To set the IP address for PME to use to connect to the RX3i, open the target properties, set Physical Port to ETHERNET, and then enter the factory default IP address value.

**Note:** The factory-loaded default IP address is valid only when Hardware Configuration has never been stored to the Controller. This value is overwritten with the configured IP address each time that Hardware Configuration is stored to the Controller.

**Method 2:** Through the Ethernet connection of an ETM001-jx/ETM001-Kxxx in the same rack with a known IP address configuration.

**Method 3:** Through the RS-232 COM1 serial port – This is a data communications equipment (DCE) port that allows a simple straight-through cable to connect with a standard nine-pin AT-style RS-232 port.

**Method 4:** CPE310: Through the RS-485 COM2 serial port – Use SNP programming cable IC690ACC901

**Note:** CPE100/CPE115, CPE302, CPE305, and CPE310 do not support the alternate methods of setting a temporary IP address:

- Set Temporary IP Address tool in PME
- BOOTP
- The Station Manager *CHSOSW* command.

CPE330 supports the *Set Temporary IP Address* tool in PME. The Station Manager *CHSOSW* command is not supported.

Since the IP Addresses of the CPE400 and CPL410 may be displayed on its OLED display, these CPUs do not support the *Set Temporary IP Address* tool.

To restore the default IP Address of the CPE100/CPE115, refer to Section 4.1.3, *Configuring the Ethernet Interface Parameters*, of this document.

### 4.1.3 Configuring the Ethernet Interface Parameters

To establish communications between the computer hosting PME and the CPU, consider the following methods:

<p>Default IP Addresses for CPE302/CPE305/CPE310/CPE330 /CPE400/CPL410 Embedded Ethernet</p>	<p>Initial Ethernet communication with the CPU may be established using the default IP addresses programmed at the factory:</p> <p><b>Note:</b> The IP subnet 192.168.180.x is reserved on the CPE400 and CPL410. It is not available for configuration on any of the CPU's Ethernet ports.</p> <table border="1" data-bbox="589 630 1479 900"> <thead> <tr> <th></th> <th>CPE302/CPE305/CPE310 and CPE330/CPE400 LAN1</th> <th>CPE330/CPE400 LAN2</th> <th>CPE400 LAN3<sup>6</sup></th> </tr> </thead> <tbody> <tr> <td><b>IP Address:</b></td> <td>192.168.0.100</td> <td>10.10.0.100</td> <td>N/A</td> </tr> <tr> <td><b>Subnet Mask:</b></td> <td>255.255.255.0</td> <td>255.255.255.0</td> <td>N/A</td> </tr> <tr> <td><b>Gateway:</b></td> <td>0.0.0.0</td> <td>0.0.0.0</td> <td>N/A</td> </tr> </tbody> </table>		CPE302/CPE305/CPE310 and CPE330/CPE400 LAN1	CPE330/CPE400 LAN2	CPE400 LAN3 <sup>6</sup>	<b>IP Address:</b>	192.168.0.100	10.10.0.100	N/A	<b>Subnet Mask:</b>	255.255.255.0	255.255.255.0	N/A	<b>Gateway:</b>	0.0.0.0	0.0.0.0	N/A
	CPE302/CPE305/CPE310 and CPE330/CPE400 LAN1	CPE330/CPE400 LAN2	CPE400 LAN3 <sup>6</sup>														
<b>IP Address:</b>	192.168.0.100	10.10.0.100	N/A														
<b>Subnet Mask:</b>	255.255.255.0	255.255.255.0	N/A														
<b>Gateway:</b>	0.0.0.0	0.0.0.0	N/A														
<p>Default IP Addresses for RSTi-EP CPE100/CPE115 Embedded Ethernet</p>	<p>Initial Ethernet communication with the CPU may be established using the default IP addresses programmed at the factory:</p> <p><b>Note:</b> The IP subnet 192.168.180.XXX is reserved on the CPE100/115. It is not available for configuration on any of the CPU's Ethernet ports.</p> <table border="1" data-bbox="589 1081 1479 1320"> <thead> <tr> <th></th> <th>CPE100/115 LAN1</th> <th>CPE100/115 LAN2</th> </tr> </thead> <tbody> <tr> <td><b>IP Address:</b></td> <td>192.168.0.100</td> <td>0.0.0.0</td> </tr> <tr> <td><b>Subnet Mask:</b></td> <td>255.255.255.0</td> <td>0.0.0.0</td> </tr> <tr> <td><b>Gateway:</b></td> <td>0.0.0.0</td> <td>0.0.0.0</td> </tr> </tbody> </table>		CPE100/115 LAN1	CPE100/115 LAN2	<b>IP Address:</b>	192.168.0.100	0.0.0.0	<b>Subnet Mask:</b>	255.255.255.0	0.0.0.0	<b>Gateway:</b>	0.0.0.0	0.0.0.0				
	CPE100/115 LAN1	CPE100/115 LAN2															
<b>IP Address:</b>	192.168.0.100	0.0.0.0															
<b>Subnet Mask:</b>	255.255.255.0	0.0.0.0															
<b>Gateway:</b>	0.0.0.0	0.0.0.0															
<p>Connecting to CPE302/CPE305/CPE310 Embedded Ethernet when IP Addresses are not known</p>	<p>If the IP address of the CPE302/CPE305/CPE310 embedded Ethernet interface is not known, communication may be established using one of these methods to set a permanent IP address:</p> <ul style="list-style-type: none"> <li>• Connect to the CPE302/CPE305/CPE310 via its serial port and assign an IP Address to the embedded Ethernet interface by downloading a <i>Hardware Configuration</i>.</li> <li>• Connect to the CPE302/CPE305/CPE310 with PME using an IC695ETM001 module with a known IP address and located in the same rack. Download a new <i>Hardware Configuration</i> with the desired IP address for the embedded Ethernet interface.</li> </ul>																

<sup>6</sup> When the CPE400/410 Target property is configured for *Enable Redundancy = True*, LAN3 will display as *3 Redundancy* in PME and *Switched* for both LAN3 port with *LAN 3 = Redundancy* (greyed out). LAN3 will not be available to configure.

<p>Connecting to CPE330 Embedded Ethernet when IP Addresses are not known</p>	<p>If the IP addresses of the CPE330 embedded LAN1 and LAN2 Ethernet interfaces are not known, communication may be established using one of these methods to set new IP addresses:</p> <ul style="list-style-type: none"> <li>• Setting a Temporary IP Address using the <i>Set Temporary IP Address</i> tool in PME. After setting the temporary address, connect to the selected CPE330 LAN using PME and download a new <i>Hardware Configuration</i> with the desired permanent IP addresses.<sup>7</sup></li> <li>• Connect to the CPE330 with PME using an IC695ETM001 module with a known IP address and located in the same rack. Download a new <i>Hardware Configuration</i> with the desired permanent IP addresses for the CPE330 embedded Ethernet interfaces.</li> </ul>
<p>Connecting to CPE400 or CPL410 Embedded Ethernet when IP Addresses are not known</p>	<p>Use the OLED display to read the IP Address of any LAN.</p> <p><b>Note:</b> The <i>Set Temporary IP Address</i> tool is not available for CPE400 or CPL410.</p>
<p>Connecting to CPE100/CPE115 Embedded Ethernet when IP Addresses are not known</p>	<p>If the IP addresses of the CPE100/CPE115 embedded LAN1 Ethernet interfaces are not known, communication may be established using the below method to set default IP addresses:</p> <p>Power-up CPE100 with the Power pushbutton pressed and wait until the OK LED flashes twice. This forces the CPE100/CPE115 LAN1 to reset to the default IP address of 192.168.0.100.</p> <p><b>Caution:</b> Resetting to the default IP address using the above procedure also erases the stored <i>Hardware Configuration</i>, logic, and contents of the backup RAM.</p>

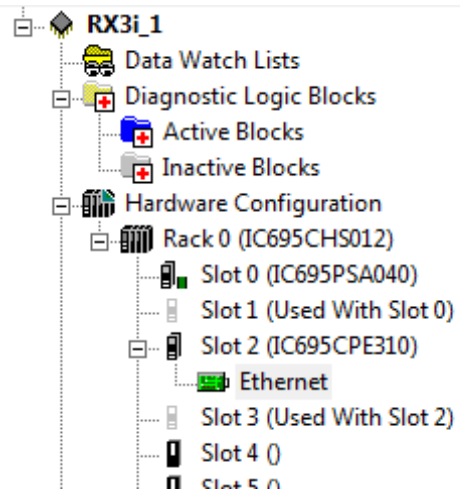
---

<sup>7</sup> This method is not supported in firmware 9.80 or later.  
*Configuration*

## Configuring an RX3i/RSTi-EP Embedded Ethernet interface

1. In the Project tab of the Navigator, expand the PACSystems Target, the *Hardware Configuration*, and the main rack (Rack 0).
2. Expand the CPU slot (Slot 2). The Embedded Ethernet interface is displayed as “Ethernet” (Figure 16).
3. Right-click the daughterboard slot and choose “Configure.” The Parameter Editor window displays the Ethernet interface parameters.
4. To add the Ethernet Global Data component, right-click the Target. Select “Add Component” and then “Ethernet Global Data.”
5. Select the desired tab, and then click in the appropriate Values field.

**Figure 16: Expand RX3i CPU Node to Configure Embedded Ethernet interface**





## Ethernet Parameters (Settings Tab)

To access the Settings tab for an RX3i/RSTi-EP embedded control unit, locate the desired unit in the Navigator pane, and double click the unit icon to view its settings.

**Configuration Mode:** The Configuration Mode is fixed as TCP/IP.

**Adapter Name:** The Adapter Name is automatically generated based upon the rack/slot location of the Ethernet interface. For example, a module in Rack 0, Slot 1 would be designated as “0.1”.

**IP Addresses:** These values should be assigned by the person in charge of your network (the network administrator). TCP/IP network administrators are familiar with these parameters. These parameters must be correct, otherwise, the Ethernet interface may be unable to communicate on the network, and/or network operation may be corrupted. Each node on the network must be assigned a *unique* IP address.

If you have no network administrator and are using a simple *isolated network* with no gateways, you can use the following range of values for the assignment of local IP addresses:

10.0.0.1	First Ethernet interface
10.0.0.2	Second Ethernet interface
10.0.0.3	Third Ethernet interface
.	.
.	.
.	.
10.0.0.255	Programmer TCP or host

Also, in this case, set the subnet mask to 255.0.0.0 and the Gateway IP address to 0.0.0.0.

**Note:** If the isolated network is connected to another network, the IP addresses 10.0.0.1 through 10.0.0.255 must not be used; and the subnet mask and gateway IP address must be assigned by the network administrator. The IP addresses must be assigned so that they are compatible with the connected network.

**Network Time Sync:** Options are “None” and “SNTP.” Select SNTP if the CPU will be synchronized to the network clock.

**Status Address:** The Status Address is the reference memory location for the Ethernet interface status data. The Ethernet interface automatically maintains 16 LA interface Status (LIS) bits in this location. The Status address can be assigned to valid %I, %Q, %R, %AI, %AQ or %W memory. The default value is the next available %I address.

The meaning of the Channel Status portion of the Ethernet Status bits depends upon the type of operation for each channel. For details of the status bits and their operation, refer to Section, 12.6 Monitoring the Ethernet Interface Status Bits.

**Note:** Do not use the 80 bits configured as Ethernet Status data for any other purpose or data will be overwritten.

**Note:** If the Ethernet interface’s Variable Mode property is set to true, the Status Address parameter is removed from the Settings tab. Instead, Ethernet Status references must be defined as I/O variables on the Terminals tab.

**Length:** This is the total length of the Ethernet interface status data. This is automatically set to either 80 bits (for %I and %Q Status address locations) or 5 words (for %R, %AI, %AQ and %W Status address locations).

**Redundant IP:** The Redundant IP option selects whether Redundant IP operation is *Enabled* or *Disabled*. When this parameter is set to *Enabled*, the Redundant IP address must be entered via the Redundant IP Address parameter. The default value is False.

**Redundant IP Address:** Redundant IP Address is an optional IP Address that will be shared with another device on the network in a Redundant System. Both devices must use the same subnet mask. This parameter is available only when the Redundant IP parameter is set to *Enabled*. This address defaults to 0.0.0.0, which is not a valid IP address; a valid Redundant IP address must be explicitly configured. See Section 1, *Introduction*, for more information about Ethernet redundancy. This IP address is assigned in addition to the device's primary IP address.

**Produce EGD on Redundant IP:** Specifies that EGD Producer exchanges will use the Redundant IP address as the source address IP address on the network instead of the local IP Address configured above.

**I/O Scan Set:** Specifies the I/O scan set to be assigned to the Ethernet interface. Scan sets are defined in the CPU's Scan Sets tab. The valid range is 1 through 32; the default value is 1.

**Ethernet Global Data:** Settings tab for CPE100/CPE115/ CPE330/CPE400/CPL410 has additional EGD configuration parameter entries. PME 8.60 SIM 5 (or later) is required for the EGD configuration parameter entries for CPE330. The EGD parameter entries are exclusive to the CPE100/CPE115/CPE330/CPE400/CPL410.

**Note:** In earlier CPU models these EGD configuration parameters were configured via AUP files. An AUP file is not supported, nor is it needed, by the CPE330, CPE400, CPL410, or CPE100/CPE115.

**Startup Delay Time for Produced Exchanges (ms):** Corresponds to the gp\_phase AUP parameter.

**Stale Consumed Exchanges:** Corresponds to the gnostale AUP parameter.

**TTL for Unicast Messages:** Corresponds to the gucast\_ttl AUP parameter.

## CPE100/CPE115/CPE330/CPE400/CPL410

Figure 17: CPE330/CPE400/CPL410/CPE100/CPE115/ETM001-Kxxx Settings tab

Settings   Advanced EGD	
Parameters	Values
Configuration Mode	TCP/IP
Adapter Name	P.0.0.0
IP Address	192.168.0.100
Subnet Mask	255.255.255.0
Gateway IP Address	0.0.0.0
Status Address	%I00001
Length	80
Redundant IP	Enable
Redundant IP Address	192.168.0.103
Produce EGD on Redundant IP	False
I/O Scan Set	1
--- Ethernet Global Data ---	
Startup Delay Time for Produced Exchanges (ms)	0
Stale Consumed Exchanges	Send Stale Status
TTL for Unicast Messages	16

### CPE330/CPE400/CPL410/CPE100/CPE115 LAN1:

**TTL for Multicast Messages:** Corresponds to the gmcast\_ttl AUP parameter.

**IP Address for Multicast Group X:** Corresponds to the gXX\_addr AUP parameters. XX identifies the group (1 – 32).

### CPE330/CPE400/CPL410/CPE100/CPE115 LAN2:

**TTL for Multicast Messages:** Corresponds to the gmcast\_ttl2 AUP parameter.

### IP Address for Multicast Group X:

Corresponds to the gXX\_addr2 AUP parameters. XX identifies group (1 – 32).

## CPE330/CPE400/CPL410/CPE100/CPE115 LAN1 and LAN2 Advanced EGD Settings

**Figure 18: CPE330 Advanced Ethernet Configuration LAN1 & LAN2**

Parameters	
IP Address	3.16.233.121
Subnet Mask	255.255.252.0
Gateway IP Address	3.16.232.1
--- Ethernet Global Data ---	
TTL for Multicast Messages	1
IP Address for Multicast Group 1	224.0.7.1
IP Address for Multicast Group 2	224.0.7.2
IP Address for Multicast Group 3	224.0.7.3
IP Address for Multicast Group 4	224.0.7.4
IP Address for Multicast Group 5	224.0.7.5
IP Address for Multicast Group 6	224.0.7.6
IP Address for Multicast Group 7	224.0.7.7
IP Address for Multicast Group 8	224.0.7.8
IP Address for Multicast Group 9	224.0.7.9
IP Address for Multicast Group 10	224.0.7.10
IP Address for Multicast Group 11	224.0.7.11
IP Address for Multicast Group 12	224.0.7.12
IP Address for Multicast Group 13	224.0.7.13
IP Address for Multicast Group 14	224.0.7.14
IP Address for Multicast Group 15	224.0.7.15
IP Address for Multicast Group 16	224.0.7.16
IP Address for Multicast Group 17	224.0.7.17
IP Address for Multicast Group 18	224.0.7.18
IP Address for Multicast Group 19	224.0.7.19
IP Address for Multicast Group 20	224.0.7.20
IP Address for Multicast Group 21	224.0.7.21
IP Address for Multicast Group 22	224.0.7.22
IP Address for Multicast Group 23	224.0.7.23
IP Address for Multicast Group 24	224.0.7.24
IP Address for Multicast Group 25	224.0.7.25
IP Address for Multicast Group 26	224.0.7.26
IP Address for Multicast Group 27	224.0.7.27
IP Address for Multicast Group 28	224.0.7.28
IP Address for Multicast Group 29	224.0.7.29
IP Address for Multicast Group 30	224.0.7.30
IP Address for Multicast Group 31	224.0.7.31
IP Address for Multicast Group 32	224.0.7.32

Figure 19: CPE100/CPE115/CPE400/CPL410 Advanced Ethernet Configuration LAN1 & LAN2

Settings   Advanced EGD	
Parameters	
--- Ethernet Global Data ---	
TTL for Multicast Messages	1
IP Address for Multicast Group 1	224.0.7.1
IP Address for Multicast Group 2	224.0.7.2
IP Address for Multicast Group 3	224.0.7.3
IP Address for Multicast Group 4	224.0.7.4
IP Address for Multicast Group 5	224.0.7.5
IP Address for Multicast Group 6	224.0.7.6
IP Address for Multicast Group 7	224.0.7.7
IP Address for Multicast Group 8	224.0.7.8
IP Address for Multicast Group 9	224.0.7.9
IP Address for Multicast Group 10	224.0.7.10
IP Address for Multicast Group 11	224.0.7.11
IP Address for Multicast Group 12	224.0.7.12
IP Address for Multicast Group 13	224.0.7.13
IP Address for Multicast Group 14	224.0.7.14
IP Address for Multicast Group 15	224.0.7.15
IP Address for Multicast Group 16	224.0.7.16
IP Address for Multicast Group 17	224.0.7.17
IP Address for Multicast Group 18	224.0.7.18
IP Address for Multicast Group 19	224.0.7.19
IP Address for Multicast Group 20	224.0.7.20
IP Address for Multicast Group 21	224.0.7.21
IP Address for Multicast Group 22	224.0.7.22
IP Address for Multicast Group 23	224.0.7.23
IP Address for Multicast Group 24	224.0.7.24
IP Address for Multicast Group 25	224.0.7.25
IP Address for Multicast Group 26	224.0.7.26
IP Address for Multicast Group 27	224.0.7.27
IP Address for Multicast Group 28	224.0.7.28
IP Address for Multicast Group 29	224.0.7.29
IP Address for Multicast Group 30	224.0.7.30
IP Address for Multicast Group 31	224.0.7.31
IP Address for Multicast Group 32	224.0.7.32

## SNTP PME configuration for the CPE302/CPE305/CPE310/CPE330/CPE400/CPL410 CPU settings

**Figure 20: SNTP PME configuration for the CPE302/CPE305/CPE310/CPE330/CPE400/CPL410 CPU settings**

Settings	Scan	Memory	Faults	Scan Sets	Power Consumption	Access Control	SNTP	Time	OPCUA	
<b>Parameters</b>							<b>Values</b>			
Passwords							Enabled			
Stop-Mode I/O Scanning							Disabled			
Watchdog Timer (ms)							200			
Logic/Configuration Power-up Source							Always RAM			
Data Power-up Source							Always RAM			
Run/Stop Switch							Enabled			
Memory Protection Switch							Disabled			
Power-up Mode							Last			
Modbus Address Space Mapping Type							Disabled			
Universal Serial Bus							Enabled			
LAN 1 Mode							Ethernet			
LAN 2 Mode							Ethernet			
Network Time Sync							<b>SNTP</b>			
Coordinated Universal Time(UTC)							False			
Day Light Savings Time(DST)							False			

**Figure 21: SNTP Multicast/Broadcast or Unicast Mode Settings**

Settings	Scan	Memory	Faults	Scan Sets	Power Consumption	Access Control	SNTP	Time	OPCUA	
<b>Parameters</b>							<b>Values</b>			
SNTP Mode							Multicast/Broadcast			
Poll Interval							32			
Poll Count							3			
Poll Timeout							2			

**SNTP Mode: Multicast/Broadcast, Unicast**

**Poll Interval**<sup>8</sup> (Interval for unicast, in seconds, at which new time requests are sent to the server):

Low Limit =16, High Limit=1024, modulus 2

**Poll Count** (Number of retransmissions that will be sent when no timely response is received from the server): *Low Limit =1, High Limit=100*

**Poll Timeout** (The time, in seconds, to wait for a response from the server): *Low Limit =2, High Limit=100*

**Figure 22: UTC Time Zone Settings**

Settings   Scan   Memory   Faults   Scan Sets   Power Consumption   Access Control   SNTP   Time   OPCUA									
Parameters					Values				
UTC Offset					-05:00 Eastern Time (US, Canada)				

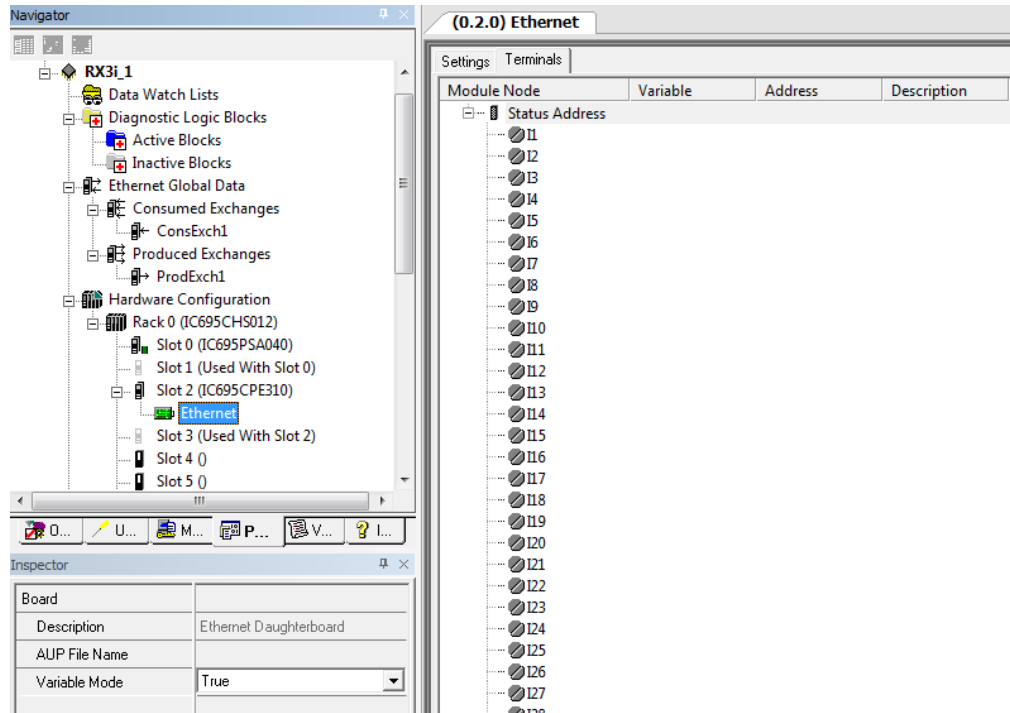
Local time zone offset with respect to UTC time. Valid range: Select the closest appropriate time zone for your location.

<sup>8</sup> Unicast must be enabled for *Poll Interval* parameter to display.  
 Configuration

## Terminals Tab

The Terminals configuration tab (Figure 23) is displayed only when the Variable Mode property of the Ethernet interface is set to True. When Variable Mode is selected, the Ethernet Status bits are referenced as I/O variables. The I/O variables are mapped to the Ethernet status bits via this configuration tab.

**Figure 23: Terminals Tab Settings in PAC Machine Edition**



The use of I/O variables allows you to configure the Ethernet interface without having to specify the reference addresses to use for the status information. Instead, you can directly associate variable names with the status bits. For more information, refer to the section on *I/O Variables* in the *PACSystems RX7i, RX3i, and RSTi-EP CPU Reference Manual, GFK-2222*.

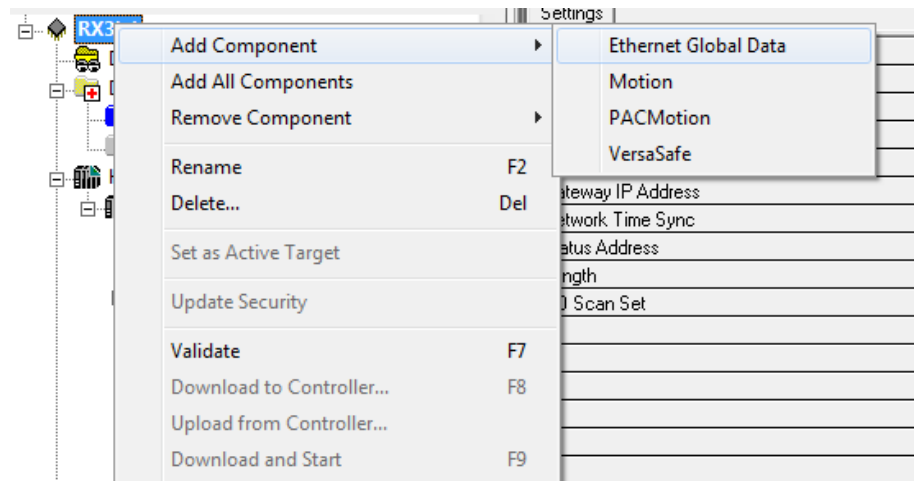


## Configuring Embedded Ethernet for Ethernet Global Data (EGD)

This section describes how to configure the parameters of an RX3i embedded PACSystems Ethernet interface. See also *Configuring Ethernet Global Data* for more information.

In the event the CPU will be used to produce or consume Ethernet Global Data (EGD), right-click on the device icon and, using the “Add Component” drop-down list, select “Ethernet Global Data,” as shown in Figure 24.

**Figure 24: Adding Ethernet Global Data (EGD) to the Configuration**



Once the EGD component has been added, it is possible to define the EGD data to be produced (Figure 25) and the EGD data to be consumed (Figure 26) by the embedded Ethernet interface, per the following screenshots.

Figure 25: Defining EGD Produced Data Exchange

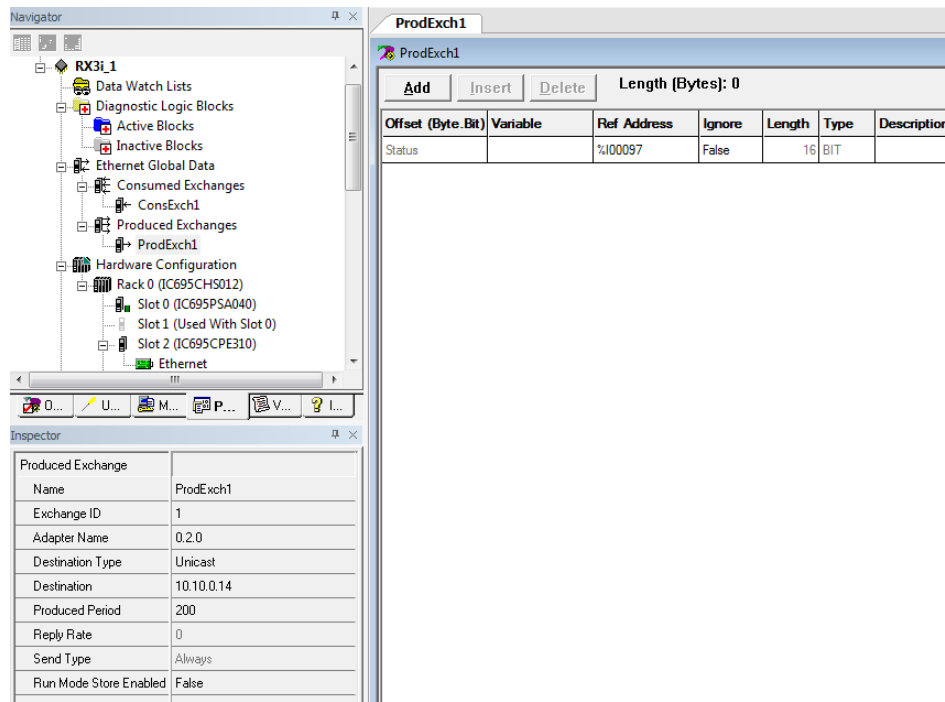
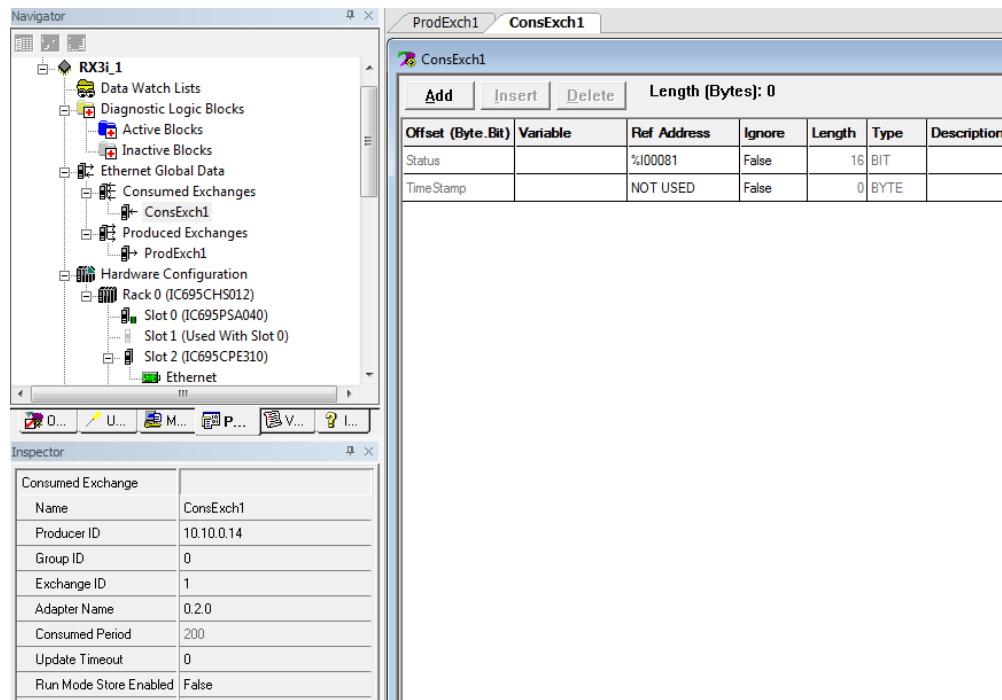


Figure 26: Defining EGD Consumed Data Exchange



The parameters to be entered and their relevance is discussed in the sections entitled *Configuring an Ethernet Global Data Exchange for a Producer* (page 80) and *Configuring an Ethernet Global Data Exchange for a Consumer* (page 82).

For more information, please see Section 5, Ethernet Global Data.

Produced exchanges (Multicast and Broadcast) configured for the CPE330's embedded Ethernet interface will have the additional parameter *Network ID* that allows the user to select LAN1 or LAN2. (Refer to the following figures.)

The Network ID parameter is only visible on produced Multicast and Broadcast exchanges.

**Figure 27: Configuring Multicast & Broadcast EGD on LAN1**

**InfoViewer (0.2.0) Ethernet**

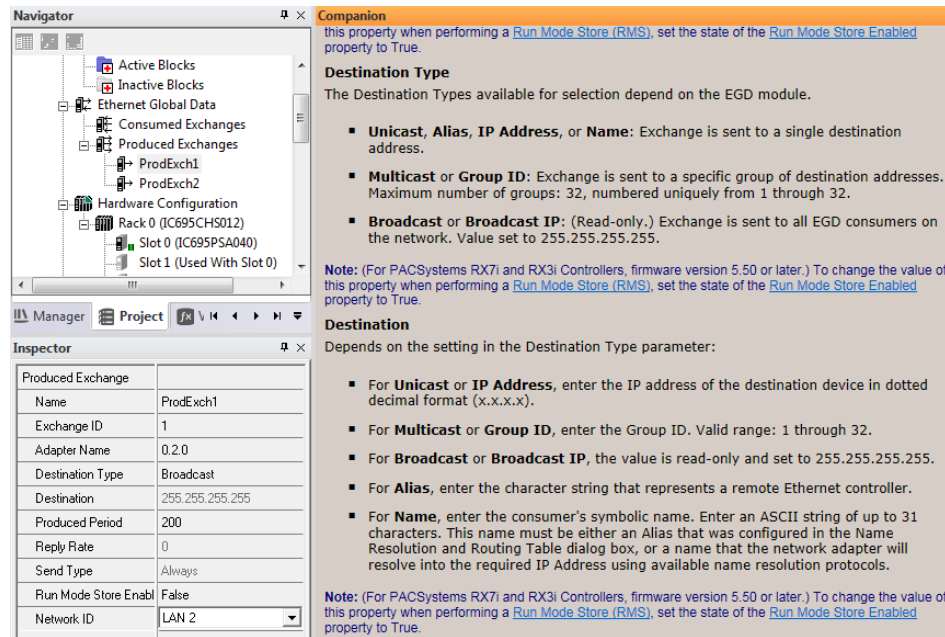
► To configure multicasts, that is, to send a produced exchange to multiple consumers:

1. In the Project tab of the Navigator, expand a GE IP Controller target's Ethernet Global Data folder.
2. Expand the Produced Exchanges, right-click the Produced Exchange to send to multiple consumers, and choose Properties. The Inspector displays the exchange's properties.
3. Set the Destination Type property as follows:
  - For a PACSystems RX7i or RX3i target, select **Multicast**.
  - For any other EGD-supporting target, select **Group ID**.
4. Set the Destination property to a value ranging from 1 through 32.
5. In the Project tab of the Navigator, expand a GE IP Controller target's Ethernet Global Data folder.
6. Expand the Consumed Exchanges and select a Consumed Exchange that is to belong to the group. The Inspector displays the exchange's properties.
7. Set the Group ID property to the same value as the Destination property of the produced exchange.
8. Repeat steps 5 through 7 for each consumer that is to belong to the group.

Produced Exchange	
Name	ProdExch1
Exchange ID	1
Adapter Name	0.2.0
Destination Type	Multicast
Destination	1
Produced Period	200
Reply Rate	0
Send Type	Always
Run Mode Store Enabl	False
Network ID	LAN 1

LAN1 will display a Network ID of 0.

**Figure 28: Configuring Multicast & Broadcast EGD on LAN2**



LAN2 will display a Network ID of 1.

## 4.2 RX3i Rack-Based Ethernet Interface Modules

The configuration process for the RX3i rack-based Ethernet interface modules include:

Assigning a temporary IP address for initial network operation, such as connecting the programmer to download the *Hardware Configuration*.

- Configuring the characteristics of the Ethernet interface.
- Configuring Ethernet Global Data (if used).
- (Optional, not required for most systems.) Setting up the RS-232 port for Local Station Manager operation. This is part of the basic Ethernet interface configuration.
- (Optional, not required for most systems.) PME provides default values for the ETM001's serial port, which will be compatible with most applications. However, the ETM001-Kxxx's front panel will require a valid IP address that does not conflict with the main Ethernet interface's IP address.
- (Optional, not required for most systems.) Both Ethernet interface modules have advanced parameters that can be configured. The Advanced User Parameters (AUP) values should only be changed in exceptional circumstances by experienced users. The ETM001-Jx can generate a separate ASCII parameter file (AUP file) to be stored to the PLC. The ETM001-Kxxx's AUP can be updated in PME's *Hardware Configuration*. If ETM001-Kxxx is being used as a drop-in replacement in the rack, the ETM001-Kxxx will adopt AUPs from a stored AUP file from the original configuration when using an original ETM001 PME configuration. (Note: some AUP parameters are no longer available. Please refer to Section *AUP Support by Ethernet Interface*.)
- (Optional) Setting up the PLC for Modbus/TCP Server operation. See Section 9 for information about configuring Modbus/TCP Server operation.

This chapter discusses only the configuration of the PACSystems Ethernet interface. Information about overall system configuration is available in other PACSystems documentation and in the Logic Developer online help.

## 4.2.1 Ethernet Configuration Data

The PACSystems PLC is configured exclusively by the PME PLC Logic Developer-PLC programmer. The Programmer can be connected over the Ethernet network. For initial programmer connection, an initial IP address must be manually assigned to the Ethernet interface as described next in this chapter. The PACSystems PLC does not support auto-configuration.

### Generating/Storing/Loading the Configuration

The PACSystems Ethernet interfaces use several types of configuration data: Ethernet Configuration, optional Ethernet Global Data Configuration, and optional Advanced User Parameter (AUP) Configuration. These configuration parameters are generated at the programmer, stored from the programmer to the PLC CPU as part of the *Hardware Configuration Store* sequence, and may be loaded from the PLC CPU into the programmer as part of the Configuration Load sequence. The optional AUP file must be manually generated with a text editor and then imported into the Programmer. The Programmer then stores any AUP files to the PLC within the Configuration Store operation. Once stored to the PLC, the PACSystems main CPU maintains the configuration data over power cycles.

### Backup Configuration Data

The PACSystems Ethernet interface saves a backup copy of the most recent Ethernet Configuration and AUP Configuration in non-volatile memory for use when the PLC is cleared. (Ethernet Global Data configuration is maintained only in the PLC CPU.) The PACSystems Ethernet interfaces maintain the backup configuration data in nonvolatile memory without battery power. (A PLC Configuration Clear does not affect the backup configuration data in the Ethernet interface.)

When the PLC configuration was not stored from the programmer, the Ethernet interface uses its backup configuration data is valid. If that data is invalid or has never been configured, factory default configuration values are used.

### Locally Edited Configuration Data

If implementing an ETM001-jx Ethernet module the *CHSOSW* and *CHPARM* Station Manager commands can be used to locally edit Ethernet configuration or AUP configuration data, if the PLC configuration was not stored from the programmer. These Station Manager commands are not active if the PLC configuration has been stored by the programmer.

Locally edited configuration changes cannot be retrieved into the PLC and loaded to the programmer. Locally edited configuration changes are always overwritten when a PLC configuration is stored into the PLC from the programmer.

## 4.2.2 Initial IP Address Assignment

Each PACSystems Ethernet interface comes from the factory with a default IP address (0.0.0.0). Since the default address is not valid on any Ethernet network, an initial IP address must be assigned for initial network operation, such as connecting the programmer to download the first *Hardware Configuration*. The initial IP address must be selected for proper operation with your network and application. See your network administrator for the proper initial IP address value.

An IP address can be set up using the following methods:

**Method 1:** On an ETM001, the *Set Temporary IP Address* utility within PME can be used to set a setting to connect PME and store the permanent network settings to the module.

**Method 2:** If using an ETM001 or ETM001-Kxxx, the IP address can be configured in *Hardware Configuration* and store the configuration over a serial connection.

**Method 3:** If the ETM001 Ethernet interface module has the factory default IP Address 0.0.0.0, a temporary IP address can be set using BOOTP over the Ethernet network (if a BOOTP server is present).

**Method 4:** If using an ETM001, an initial IP address can be set via the *CHSOSW* command from a local serially connected Station Manager terminal. For more information, see the *PACSystems TCP/IP Ethernet Communications Station Manager User Manual*, GFK-2225.

**Method 5:** If a serial connection through a Station Manager port is not available (for example, the IC695ETM001-Kxxx), the user can assign an IP address by connecting to the CPU and storing a *Hardware Configuration* to assign the IP settings.

**Method 6:** On an ETM001-Kxxx, the reset button can be used to enter *IP Setup Mode* and a new configuration can be downloaded to configure the permanent network settings to the module.

### Assigning a Temporary IP Address Using the Programming Software

To initiate Ethernet communications with the programmer, you first need to set up a temporary IP address. After the Programmer is connected, the actual IP address for the Ethernet interface (as set up in the *Hardware Configuration*) should be downloaded to the PLC. The temporary IP address remains in effect until the Ethernet interface is restarted, power cycled, or until the *Hardware Configuration* is downloaded or cleared.

- To use the Set Temporary IP Address utility, the PLC CPU must not be in RUN mode. IP address assignment over the network will not be processed until the CPU is stopped and is not scanning outputs.
- The current user logged on to the PC running the Set Temporary IP Address utility must have full administrator privileges.
- The Set Temporary IP Address utility can be used in communications with the networked PACSystems target travel across network switches and hubs. It does not work if communications travel through a router.
- The target must be located on the same sub-network (subnet) as the computer running the Set Temporary IP Address utility. The sub-network is specified by the computer's subnet mask and the IP addresses of the computer and the PACSystems Ethernet interface.

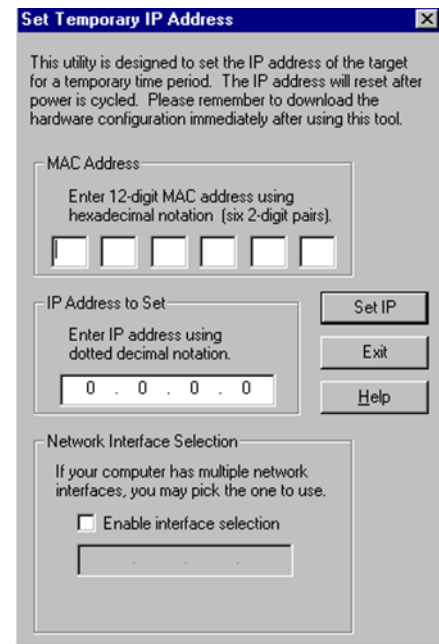
To set the IP address, you need the MAC to address of the Ethernet interface. The MAC address is located on a label on the module, as shown in 0. Connect the PACSystems Ethernet interface to the Ethernet network.

In the Project tab of the Navigator, right-click the PACSystems target. Choose Offline Commands, then Set Temporary IP Address. The Set Temporary IP Address dialog box appears.

1. In the Set Temporary IP Address dialog box, do the following:
  - Specify the MAC address of the Ethernet interface.
  - In the IP Address to Set box, specify the temporary IP address you want to assign to the Ethernet interface.
  - If the computer has multiple Ethernet network interfaces, select the Enable Network interface Selection check box and specify the network interface on which the PACSystems Ethernet interface being set up is located.
2. When the fields are properly configured, click the Set IP button.

The Set Temporary IP Address utility verifies that the specified IP address is not already in use, then it sets the target Ethernet interface to the specified IP address. Finally, the utility verifies that the target Ethernet interface responds at the selected IP address. Any error or successful completion is reported. These operations may take up to a minute

**Figure 29: Setting Temporary IP Address**



## CAUTION

The temporary IP address set by the Set Temporary IP Address utility is not retained through a power cycle. To set a permanent IP Address, you must set configure the target's IP Address and download the *Hardware Configuration* to the PACSystems target.

The Set Temporary IP Address utility can assign a temporary IP address even if the target Ethernet interface has previously been configured to a non-default IP address. (This includes overriding an IP address previously configured by the programmer.)

Use this IP Address assignment mechanism with care.

## Assigning a Temporary IP Address Using BOOTP<sup>9</sup>

To use BOOTP, the *Use BootP for IP Address* configuration option must be TRUE, and the IP Address, Subnet Mask, and Gateway IP Address must be set to 0.0.0.0.

When the PACSystems Ethernet interface receives the default IP address (0.0.0.0), either from *Hardware Configuration* or from internal backup configuration, it attempts to obtain a temporary IP address from a BOOTP server on the Ethernet network, since the Ethernet interface acts as a BOOTP client. The Ethernet interface issues a BOOT Request to the network. If a BOOTP server on the network recognizes the Ethernet interface, that server will return a BOOT Reply containing an IP address (and optionally a subnet mask and gateway IP address) to the requesting Ethernet interface.

Typically, the BOOTP server must be manually configured with the MAC address and IP address (and possibly other information such as subnet mask and gateway) for each supported client device. Each supported client must be identified by its globally unique MAC address. The Ethernet interface's MAC address is specified on its MAC Address Label as described in 0, Installation.

The BOOTP server must not be separated from the PACSystems Ethernet interface by a router. BOOTP uses broadcast messages, which typically do not pass through routers. Consult your network administrator for more details.

### CAUTION

The temporary IP address set by BOOTP is not retained through a power cycle. To set a permanent IP Address, you must configure the Ethernet interface's IP Address at the programmer and download the *Hardware Configuration* to the PLC.

Redundancy systems using should explicitly configure both the direct IP address and the Redundant IP address. For redundancy operation, do not set up the direct IP address via BOOTP.

## Assigning a Temporary IP Address Using Telnet

The temporary IP address assignment performed by the programmer's Set Temporary IP Address utility can be performed manually from a computer's DOS command window if the programming software is not available. This method uses an attempted Telnet connection to transfer the IP address, even though the PACSystems target Ethernet interface does not support normal Telnet operation.

---

<sup>9</sup> BOOTP cannot be used to specify an IP Address on the IC695ETM001-Kxxx.  
*Configuration*



## CAUTION

The Telnet method can assign a temporary IP address whether or not the Ethernet interface already has an IP address, even if the Ethernet interface has been previously configured to a non-default IP address. (This includes overriding an IP address previously configured by the programming software.)

### Use this IP Address assignment mechanism with care.

To temporarily set the IP address over the network, the PLC CPU must not be running. IP address assignment over the network will not be processed until the CPU is stopped and is not scanning outputs.

1. Obtain the Ethernet interface's MAC address from its MAC Address Label as shown in 0.
2. On the computer, open a standard DOS command window. Associate the desired IP address for the Ethernet interface with the MAC address of the Ethernet interface using the following method. In the DOS command window, enter:

```
> ARP -s ip_address mac_address
```

for *ip\_address* enter the IP address being assigned to the Ethernet interface, and for *mac\_address* enter the MAC address of the Ethernet interface.

Issue a Telnet command to the IP address (*ip\_address*) being assigned to the Ethernet interface via the following command:

```
> telnet ip_address 1
```

(This command is always sent to port 1.) This Telnet command will fail, but the IP address provided with the Telnet command will be passed to the Ethernet interface and will be temporarily activated.

The IP address assigned over the network remains in effect until the Ethernet interface is restarted, power-cycled, or until the configuration is downloaded or cleared. Once connected, the intended IP address should be permanently downloaded to the Ethernet interface via the *Hardware Configuration* data.

## Assigning a Temporary IP Address Using the Reset Button

The ETM001-Kxxx Ethernet module supports setting a temporary IP address through an *IP Setup Mode*. This mode is designed to temporarily specify the IP address of the ETM001-Kxxx. The utility is most useful during the initial configuration of an RX3i system with no serial port.

To enable the temporary IP address, begin by entering *IP Setup Mode* on the module by pressing and holding the reset button for at least five seconds while the CPU is in STOP mode. Upon the release of the reset button, the Ethernet module will enter *IP Setup Mode* and a light pattern will appear on the network speed LEDs (10/100/1000).

While in this mode, the front panel port's configuration will be:

- IP Address: 10.10.0.100
- Subnet Mask: 255.255.255.0
- Gateway IP Address: 0.0.0.0

The bottom ports' configuration will be:

- IP Address: 192.168.0.100
- Subnet Mask: 255.255.255.0
- Gateway IP Address: 0.0.0.0

*IP Setup Mode* has a timeout of 15 minutes. To exit *IP Setup Mode*, the reset button may be pressed again, or the module can be safely removed and reinstalled. If run mode is entered while in *IP Setup Mode*, the *IP Setup mode* will be automatically exited and there will be a sub-second delay before communications are available.

Once the module is in *IP Setup Mode* (the PME connection IP Address of the specified target must be set to Ethernet module's temporary setting, 192.168.0.100), select the target in the *Hardware Configuration* and navigate to the *Target* tab in the top ribbon menu. Select *Download* to download the hardware configuration while in temporary mode.

After entering the *IP Setup Mode*, the following opportunities are available:

- **Setting a permanent IP address on the device.** Creating an HWC for an RX3i, the target's IP address can be set to the desired permanent IP Address and the HWC can be downloaded to the device.
- **Acquiring the current permanent IP address.** The HWC can be uploaded to obtain its real IP address (if it had been previously configured), which can then be used for further development.

**NOTE:** There is a 500 ms communication delay after the ETM001-Kxxx has entered *IP Setup Mode*.

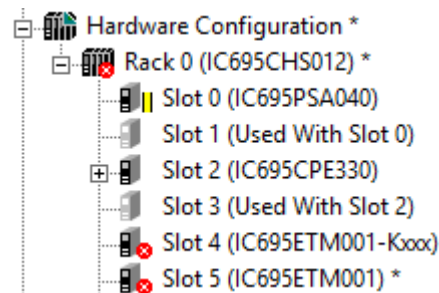
## 4.2.3 Configuring Ethernet Interface Parameters

This section describes how to configure the parameters of a rack-based PACSystems Ethernet interface.

### Configuring a Rack-based Ethernet Interface Module in PAC Machine Edition

1. In the Project tab of the Navigator, expand the PACSystems Target, the *Hardware Configuration*, and the main rack (Rack 0).
2. Right-click an empty slot and choose *Add Module*. The Module Catalog opens.
3. Click the Communications tab, select the IC695ETM001-Jx or IC695ETM001-Kxxx module, and click *OK*. The Ethernet module is placed in the rack and its parameters are displayed in the Parameter Editor window.
4. To add the Ethernet Global Data component, right-click the target element. Select *Add Component* and then Ethernet Global Data. Select the desired tab, then click in the appropriate Values field.
5. To edit parameters of a module that is already configured in the rack, right-click the slot containing the module and choose *Configure*.

**Figure 30: Install ETM001-Jx Module in Rack/Slot & Expand to Configure**



### Ethernet Parameters for Ethernet Modules in PAC Machine Edition

To access the Ethernet parameters for your device, locate your device in the Navigator pane and double click its icon to reveal its Settings tab. The following table documents which parameters are available for each Ethernet module type:

	IC695ETM001-Jx	IC695ETM001-Kxxx
<b>Ethernet Parameters</b>		
Configuration Mode	X	X
Adapter Name	X	X
BOOTP for IP Address	X	
IP Addresses	X	
Subnet Mask	X	
Gateway IP Addresses	X	
Name Server IP Addresses	X	
Max Web Server Connections	X	
Max FTP Server Connections	X	
Network Sync Type	X	X
UTC Offset		X

Day Light Savings Time (DST)		X
Status Address	X	X
Length <sup>10</sup>	X	X
Redundant IP	X	X
Produce EGD on Redundant IP		X
I/O Scan Set	X	X
<b>SNTP</b>		
CPU TOD Clock Sync		X
SNTP Mode (Unicast/Multicast/Broadcast)		X
Primary IP Address		X
Secondary IP Address		X
Poll Interval		X
Poll Count		X
Poll Timeout		X
<b>RS232- Port (Station Manager)</b>		
Baud Rate	X	
Parity	X	
Flow Control	X	
Stop Bits	X	
<b>Terminals Tab</b>		
Variable Mode	X	X
<b>Ethernet Global Data</b>		
Startup Delay Time for Produced Exchanges (ms)		X
Stale Consumed Exchanges		X
TTL for Unicast Messages		X
TTL for Multicast Messages		X
IP Address for Multicast Group 1-32		X
<b>LAN1</b>		
IP Address		X
Subnet Mask		X
Gateway IP Address		X
Ethernet Speed for Port 1		X
Ethernet Speed for Port 2		X
<b>Front Panel</b>		
IP Address		X
Subnet Mask		X
Gateway IP Address		X
<b>Time</b>		
UTC Offset		X

**Configuration Mode:** The configuration mode value is fixed as TCP/IP.

**Adapter Name:** The adapter name is automatically generated based upon the rack/slot location of the Ethernet interface. For example, if the module is in Rack 0, Slot 1, the adapter name would be “0.1”

<sup>10</sup> This parameter is not configurable for either Ethernet module.

**Use BOOTP for IP Address:** This selection specifies whether the Ethernet must obtain its working IP address over the network via BOOTP. The default is “False,” in which the IP Address value must be configured by the IP Address parameter on the same screen. When set to “True,” the IP Address parameter is forced to 0.0.0.0 and becomes non-editable.

**Note:** The IP Address, Subnet Mask, and Gateway IP Address must all be set to 0.0.0.0 to use BOOTP to obtain the IP address.

**IP Addresses:** These values should be assigned by the person in charge of your network (the network administrator). TCP/IP network administrators are familiar with these parameters. It is important that these parameters are correct, or the Ethernet interface may be unable to communicate on the network and/or network operations may be corrupted. It is especially important that each node on the network is assigned a *unique* IP address.

If you have no network administrator and are using a simple *isolated network* with no gateways, you can use the following range of values for the assignment of local IP addresses:

10.0.0.1	First Ethernet interface
10.0.0.2	Second Ethernet interface
10.0.0.3	Third Ethernet interface
.	.
.	.
.	.
10.0.0.255	Programmer TCP or host

Also, in this case, set the subnet mask to 255.0.0.0 and the gateway IP address to 0.0.0.0.

**Note:** If the isolated network is connected to another network, the IP addresses 10.0.0.1 through 10.0.0.255 must not be used and the subnet mask, and gateway IP address must be assigned by the network administrator. The IP addresses must be assigned so that they are compatible with the connected network.

**Subnet Mask:** Contact your network administrator for the value to assign. Both subnetting and supernetting are supported. For an isolated network with no gateways, use the default value: 0.0.0.0. To learn more about subnet mask usage, read *Subnet Addressing and Subnet Masks*.

**Gateway IP Address:** Contact your network administrator for the value to assign. Both subnetting and supernetting are supported. For an isolated network with no gateways, use the default value: 0.0.0.0. To learn more about

Gateways.

**Name Server IP Address:** This parameter must be set to 0.0.0.0

**Max Web Server Connections:** (Available only when the Ethernet interface supports web server operation.) The maximum number of web server connections. This value corresponds to the number of TCP connections allocated for use by the web server, rather than the number of web clients. The valid range is 0 through 16. Default is two.

**Max FTP Server Connections:** This value corresponds to the number of TCP connections allocated for use by the FTP server, rather than the number of FTP clients. Each FTP client uses two TCP connections when an FTP connection is established. The valid range is zero through 16. Default is two.

**Note:** The sum of Max Web Server Connections and Max FTP Server Connections must not exceed 16 total connections.

**Network Time Sync:** Selects the method used to synchronize the real-time clocks over the network. The choices are:

- None (for no network time synchronization)
- SNTP (for synchronization to remote SNTP servers on the network)

If None is selected, the time stamp value for a consumed EGD exchange is obtained from the local clock of the producing Controller or PLC. Timestamps of exchanges produced by a PLC with this setting are not synchronized with the timestamps of exchanges produced by other PLCs.

Refer to Section 7.4, *Time-Stamping of Ethernet Global Data Exchanges*, for more information.

**Status Address:** The Status Address is the reference memory location for the Ethernet interface status data. The Ethernet interface will automatically maintain 16 LAN Interface Status (LIS) bits in this location and 64 Channel Status bits in this location for a total of 80 bits.

The Status address can be assigned to valid %I, %Q, %R, %AI, %AQ or %W memory.

The default value is the next available %I address. See Section 12 *Diagnostics*, for definitions of the LAN interface Status (LIS) portion of the Ethernet Status data.

The definition of the Channel Status depends on the type of operation for each channel.

For details of the status bits and their operation, refer to 12.6, *Monitoring the Ethernet Interface Status Bits*

**Note:** Do not use the 80 bits configured as Ethernet Status data for other purposes or data will be overwritten.

**Note:** If the Ethernet interface's Variable Mode property is set to true, the Status Address parameter is removed from the Settings tab. Instead, Ethernet Status references must be defined as I/O variables on the Terminals tab (see section *Terminals Tab*). The Terminals tab will become available after the Variable Mode property is set to true.

**Length:** The Length parameter is the total length of the Ethernet interface status data. The value is automatically set to either 80 bits (for %I and %Q Status address locations) or 5 words (for %R, %AI, %AQ and %W Status address locations).

**Redundant IP:** The Redundant IP option selects whether Redundant IP operation is *Enabled* or *Disabled*. When this parameter is set to *Enabled*, the Redundant IP address must be entered via the Redundant IP Address parameter. The default value is False.

**Redundant IP Address:** Redundant IP Address is an optional IP Address that will be shared with another device on the network in a Redundant System. Both devices must use the same subnet mask. This parameter is available only when the Redundant IP parameter is set to *Enabled*. This address defaults to 0.0.0.0, which is not a valid IP address; a valid Redundant IP address must be

explicitly configured. See Section 1, *Introduction*, for more information about Ethernet redundancy. This IP address is assigned in addition to the device's primary IP address.

**Produce EGD on Redundant IP:** Specifies that EGD Producer exchanges will use the Redundant IP address as the source address IP address on the network instead of the local IP Address configured above.

**I/O Scan Set:** The I/O Scan Set parameter specifies the I/O scan set to be assigned to the Ethernet interface. Scan sets are defined in the CPU's Scan Sets tab. The valid range is one through 32; the default value is one.

**Note:** The Ethernet interface delivers its Ethernet Status (including Channel Status bits) during its input scan. Each channel's data transfer updates the Channels Status bits, so channel performance may be reduced if the Ethernet interface is configured to use an I/O Scan Set that runs more slowly than the PLC logic sweep.

If the Ethernet interface is configured to use an inactive I/O Scan Set, the Channels Status bits will not be transferred and channel operations will not complete.

## RS-232 Port (Station Manager) Tab

These parameters are for the RS-232 Station Manager serial port. The following defaults should be used for most applications.

**Baud Rate:** Data rate (bits per second) for the port. Choices are 1200, 2400, 4800, 9600, 19.2k, 38.4k, 57.6k, 115.2k. The default value is 9600.

**Parity:** Type of parity to be used for the port. Choices are None, Even, or Odd; the default value is None.

**Flow Control:** Type of flow control to be used for the port. Choices are None or Hardware. (The Hardware flow control is RTS/CTS crossed). The default value is None.

**Stop Bits:** The number of stop bits for serial communication. Choices are One or Two; the default value is One.

## Terminals Tab

The Terminals tab is displayed only when the Ethernet interface's Variable Mode property is set to *True*.

Select View from the primary menu bar and check the checkbox next to Inspector to enable to the Inspector pane. With the device selected, the Variable Mode value can be configured.

When Variable Mode is enabled, the Ethernet Status bits are referenced as I/O variables that are mapped to the Ethernet status bits on this configuration tab.

The use of I/O variables allows you to configure the Ethernet interface without having to specify the reference addresses to use for the status information. Instead, you can directly associate variable names with the status bits. For more information, refer to the section on *I/O Variables* in the *PACSystems RX7i, RX3i and RSTi-EP CPU Reference Manual*, GFK-2222.

## Time Tab

Coordinated Universal Time (UTC) and Day Light Savings Time (DST) configuration settings. Set the UTC offset and DST start/end times here.

**UTC Offset:** Local time zone offset with respect to UTC time. Valid range: Select the closest appropriate time zone for your location. Default: [UTC-5] Eastern Standard Time.

**DST Offset:** The offset between DST and standard time in hours and minutes. Minutes are limited to the values 0, 15, 30, 45. Default: zero.

**DST Start/End Month:** The month when DST starts or ends. Valid range: January to December. Default: January.

**DST Start/End Day:** The day when DST starts or ends. Valid range: Sunday to Saturday. Default Sunday.

**DST Start/End Week:** The week of the month when DST starts or ends. Valid range: 1 to 5.

**DST Start/End Time:** Valid range: 0:00 to 23:59. Default: zero.

**DST Ref Zone:** Indicates the time zone of reference for the DST Start and End times. Start and End times may be relative to either UTC or Local time. Choices: UTC, Local Time. Default: UTC.

## SNTP Tab

**CPU TOD Clock Sync:** Specifies whether the Ethernet module should sync with the TOD clock on the controller's CPU. Choices are *Enabled* or *Disabled*.

**SNTP Mode:** The timestamp value for a consumed EGD exchange is obtained from a central clock, located on a user-supplied SNTP time server on the network. All controllers with this setting (SNTP) have synchronized exchange time stamps. Synchronized time stamps enable you to compare groups of data to determine the order in which they were produced or to determine which contain the most current values. Choices are *Unicast* or *Multicast*

**Primary IP Address:** (Only available in **unicast** mode.) The primary address identifies hosts on a single network link. The primary address should specify the primary server. Use of **BOOTP** must be set to True for IP address of 0.0.0.0

**Secondary IP Address:** (Only available in **unicast** mode.) If the primary server is down, the secondary address can be polled.

**Poll Interval:** (Interval for unicast, in seconds, at which new time requests are sent to the server): *Low Limit = 16, High Limit = 1024, modulus 2*

**Poll Count:** (Number of retransmissions that will be sent when no timely response is received from the server): *Low Limit = 1, High Limit = 100*

**Poll Timeout:** **Poll Timeout** (The time, in seconds, to wait for a response from the server): *Low Limit = 2, High Limit = 100*



## Ethernet Global Data Section

**Startup Delay Time for Produced Exchanges (ms):** Startup delay time, for successively produced exchanges. Valid range: zero through 65535. Default is zero.

**Stale Consumed Exchanges:** Indicates how to handle stale consumed exchanges. Choices are *Send Stale Status* or *Do Not Send Stale Status*. Default value: *Send Stale Status*

**TTL for Unicast Messages:** IP time-to-live for unicast messages. Valid range: 0 through 255. Default is 16.

### LAN1 Tab (ETM001-Kxxx Only)

**IP Address:** Should be assigned by the network administrator. Both subnetting and supernetting are supported. If you have no network administrator and a simple, isolated network with no gateways, assign a unique value to IP Address in the format 3.0.0.x (where x ranges from 1 to 255).

**Subnet Mask:** Contact your network administrator for the value to assign. Both subnetting and supernetting are supported. For an isolated network with no gateways, use the default value.

**Gateway IP address:** Contact your network administrator for the value to assign. For an isolated network with no gateways, use the default value.

**Ethernet Speed for Port 1:** The Ethernet ports on the bottom of the module can have their traffic speed limited. Choices are *Auto-detect*, *10 Mbps*, or *100 Mbps*.

**Ethernet Speed for Port 2:** The Ethernet ports on the bottom of the module can have their traffic speed limited. Choices are *Auto-detect*, *10 Mbps*, or *100 Mbps*.

### Front Panel Tab (ETM001-Kxxx Only)

**IP Address:** Should be assigned by the network administrator. Both subnetting and supernetting are supported. If you have no network administrator and a simple, isolated network with no gateways, assign a unique value to IP Address in the format 3.0.0.x (where x ranges from 1 to 255).

**Subnet Mask:** Contact your network administrator for the value to assign. Both subnetting and supernetting are supported. For an isolated network with no gateways, use the default value.

**Gateway IP address:** Contact your network administrator for the value to assign. For an isolated network with no gateways, use the default value.

## 4.2.4 Configuring Ethernet Global Data

For more information about *Ethernet Global Data*, see Section 5.

Ethernet Global Data can be configured in two ways. The most convenient way is to use the Ethernet Global Data server that is provided with the PLC programming software. This server holds the EGD configurations for all the devices in the EGD network. When the Configuration Server is used, the EGD configuration for the entire EGD network can be validated for accuracy before the configuration is stored into the devices of the network. This can greatly decrease the time needed to commission a network or implement changes in a network.

EGD exchanges can also be configured without using the server. Both methods are described in this chapter. The choice of whether to use the Configuration Server can be made individually for each device.

**Note:** Some items in this discussion do not apply to Ethernet network interface units when using ENIU templates. For configuration of EGD with ENIUs, refer to the *PACSystems RX3i Ethernet NIU User Manual*, GFK-2439.

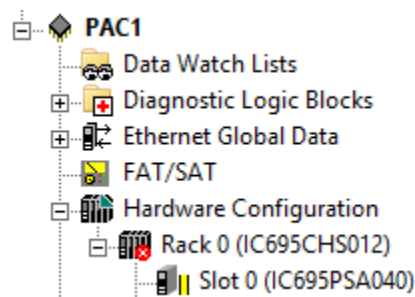
### Basic EGD Configuration

To use an EGD Configuration, follow the steps below:

If Ethernet Global Data does not appear as shown, open the Project folder and expand the target node for the PLC (**PLC1** in this example).

1. Right-click the PLC icon and select *Add Component*
2. Select Ethernet Global Data
3. To configure the Local Producer ID, right-click the Ethernet Global Data node and choose *Properties*. The Local Producer ID is shown in the Inspector pane. This parameter must be unique on the network.

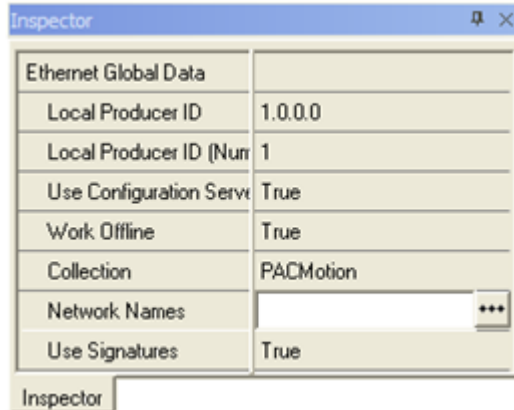
**Figure 31: Expand Node to View Ethernet Global Data**



The Local Producer ID (Figure 32) is a 32-bit value that uniquely identifies this Ethernet Global Data device across the network. It can either be expressed as a dotted-decimal value in the same way an IP address value is specified or specified as an integer. It is recommended that this value be set to the address of the Ethernet interface with the lowest rack/slot location in the system. The same Producer ID applies to all exchanges produced by this CPU, regardless of which Ethernet interface is used to send the exchange to the network.

While the form of the Producer ID is sometimes the same as that of an IP address and an IP address is used as its default value, the Producer ID is not an IP address. See Section 5, *Ethernet Global Data*, for more information on how the Producer ID is used.

Figure 32: Local Producer ID



## Configuring Ethernet Global Data Using the EGD Configuration Server

The EGD Configuration Server is supplied with PAC Machine Edition (PME) software, but it is not automatically installed with PME. To use the EGD Configuration Server and its associated tools, the server must be installed on the computer as described below.

### Installing the EGD Configuration Server

To install the EGD Configuration server, insert the PME software CD (PAC\_MachineEdition\_v9.50.0.7677\_English or later). Browse to directory *EGD Server and Tools* and locate the Windows Installer file *PAC EGD*. Complete the install with the installation wizard.

### Configuring the EGD Configuration Server

To configure the Ethernet Global Data, click on the Options tab located at the bottom of the Navigator window. In the *Machine Edition* folder, select the *EGD* item to display the configuration options for the configuration server.

Figure 33: Configuring the EGD Configuration Server

The screenshot shows a window titled "Inspector" with a table of configuration parameters for the EGD server. The parameters and their values are as follows:

EGD Preferences	
Local Server Cache Path	C:\Users\Public\Documents\GE Intelligi
Base Path	/EGD
Host Name	127.0.0.1
Server Port	7938
Timeout	20000
Configuration Server	Located
Publish on Validate	True

**Local Server Cache Path:** This parameter sets the path to be used for caching data from the configuration server. This cache is used if the server becomes inaccessible. (For example, if the server is on another machine and that machine is inaccessible due to loss of network communications.) You can also choose to work offline from the server and use this cache. This mode of operation is explained below.

**Base Path:** Typically, this field should not be changed from the default of */EGD*. This is the path portion of the URL used to get to the server. This is the default Base Path set by the Windows Installer.

**Host Name:** The hostname for the computer on which the configuration server runs. This can be specified as "localhost" if the server is on the local machine.

**Server Port:** This parameter is typically left at the default of 7938. If changed, it must be changed on both the programming software and on the server. This value is global and not project-specific. It will be used as the default by other projects created on that computer and by other tools such as the EGD Management Tool that require access to the server.

**Timeout:** The number of milliseconds the programming software will wait for a reply from the server before deciding that the server is not going to respond.

**Configuration Server:** This read-only parameter displays the value *Located* if the configuration server can be accessed and *Unable to Locate* if the server is not accessible.

When using the configuration server, the producer of data normally defines the exchange. See below for a step-by-step description of defining an exchange in the producer. After the producer of the data defines the exchange, consumers may make use of the exchange. Each consumer selects the desired exchange from the list of produced exchanges and defines the local PLC memory to be used for the variables of interest from the exchange. Consumers can be resynchronized with any changes in the producer on request. Consistency between the producer and consumer(s) is verified during the build and validate the process.

## Enabling the use of the EGD Configuration Server for a Device

To enable the use of the configuration server for a device, right-click the Ethernet Global Data node and choose Properties. To use the EGD Configuration Server, selected True next to the *Use Configuration Server* option.

To work offline from the configuration server, select the *Work Offline* parameter and set it to *True*. The programmer keeps a local copy or cache of the EGD configuration information at a configurable path (see *Configuring the EGD Configuration Server*).

Setting the *Local Server Cache Path* to a location on the local machine and setting the *Work Offline* to *True* allows EGD configuration data to be updated using the cached information without accessing the server. Setting the *Work Offline* parameter to *False* and performing a *Validate* command will synchronize the server with the data from the cache.

## Network Names and Collections

To perform validation between producers and consumers, it is necessary to verify whether the producer and the consumer are on the same network. The EGD Configuration Server and its validation libraries use the network name to perform this check. The validation assumes that two devices that have the same network name are connected to the same network. To set the network name, right-click the Ethernet Global Data node and choose Properties. The Network Name option is displayed in the Properties Inspector window. This parameter may be set to the name of the network to which the device is connected. To assign or edit a network name for target, click

## Setting up Collections for the EGD Management Tool

The EGD Management Tool is an optional utility that can be used to provide a system-level look at all the Ethernet Global Data devices in a system. Installation and use of the EGD Management Tool are described in Section 12, *Diagnostics*.

The EGD Management Tool can look at subsets of EGD devices, called a Collection. A Collection is a logical grouping of EGD devices (for example a manufacturing cell or a machine). To make an EGD device part of a collection, right-click the Ethernet Global Data node and choose Properties. The Collection option is displayed in the Properties Inspector window. This parameter may be set to the name of the collection for the device (by default the collection for a device is the *Machine Edition* project name).

## Configuring an Ethernet Global Data Exchange for a Producer

The information to be sent by the producer and the exchange details are defined in the Properties for each produced exchange. When an individual produced exchange is selected, the Properties Inspector window permits user configuration of the following information.

<b>Name</b>	A name was assigned for this exchange. Defaults to “ProdExchX” where X is a sequential number.
<b>Exchange ID</b>	A unique ID number that identifies a specific exchange to be sent by the producing device.
<b>Adapter Name</b>	The Adapter Name is the name of a specific Ethernet interface, which is identified by its rack and slot location within the consuming PLC.

<b>Destination Type</b>	<p>Specifies whether the data's destination will be:</p> <ul style="list-style-type: none"> <li>▪ An IP address (Unicast)</li> <li>▪ A Group ID (Multicast)</li> <li>▪ All EGD nodes on the subnet (Broadcast). Choosing Broadcast will cause the EGD packets to be received by any node on the network. This can impact performance if there are non-EGD devices on the network. Check with the system's network administrator if you are unsure about whether to use Broadcast.</li> </ul>
<b>Destination</b>	<p>Identifies the data's consuming device, based on the Destination Type selected above:</p> <ul style="list-style-type: none"> <li>▪ a dotted-decimal IP address if Destination Type is IP Address</li> <li>▪ the group's ID (1–32) if Destination Type is Group ID</li> <li>▪ the value 255.255.255.255 if Broadcast IP is the Destination Type.</li> </ul>
<b>Produced Period</b>	<p>The scheduled repetition period at which the data is produced on the network. Configure a value in the range of zero or two–3,600,000 (2 milliseconds to 1 hour). The value zero means data will be produced at the end of each PLC scan, but not less than two milliseconds from the previous production. Set the production period to ½ the period at which the application needs the data in this exchange. Round this value up to the nearest two milliseconds (2 ms).</p>
<b>Reply Rate</b>	<p>Not used.</p>
<b>Send Type</b>	<p>Fixed at <i>Always</i>. In the PLC, production of EGD is controlled by the I/O state: when enabled, EGD production is enabled, and when disabled, EGD production is disabled.</p>
<b>Run Mode Store Enabled</b>	<p>When set to True, allows you to modify or delete this exchange and store the changes while in Run mode. You can add exchanges in Run mode regardless of the setting of this parameter. It is recommended that you keep this parameter at its default setting, False, unless your application has a specific need to modify this exchange in Run mode.</p>
<b>Network ID</b>	<p>Allows the user to select either <i>LAN1</i> or <i>LAN2</i> for Multicast or Broadcast production.</p> <p>Note: The <i>Network ID</i> field is only visible if the <i>Destination Type</i> is configured for Multicast or Broadcast (not Unicast) AND the <i>Adaptor Name</i> selects a device rack/slot that physically supports multiple LANs (for example, the CPE330).</p>

## Configuring the Exchange Variables

Double-clicking on the produced exchange opens a window to configure the variables within the exchange. Each exchange has its own variable list. These variables contain the data that is produced for the network. Each variable contains the following information:

<b>Offset (Byte.Bit)</b>	The location within the data area for this exchange where the start of the data for this variable is located. The offset is expressed as <i>Byte.Bit</i> , where <i>Byte</i> is a zero-based byte offset and <i>Bit</i> is a zero-based bit position within that byte. (Valid bit values are 0–7. Bit 0 is the least-significant bit within the byte; bit 7 the most significant.)
<b>Variable</b>	The name is defined for this variable. It may be an existing variable, or it may be defined using the variable declaration facilities of the programmer such as the variable list in the Navigator.
<b>Ref Address</b>	The PLC memory reference address contains the start of the data for this variable.
<b>Ignore</b>	The default value is <i>False</i> . Not used for Produced exchange.
<b>Length</b>	The size of the data for this variable is expressed in units of the data type.
<b>Type</b>	The data type of the variable.
<b>Description</b>	An optional text description of this variable.

To add a new variable to the end of the exchange, click the *Add* button. This does not change the data offsets of any existing variables within that exchange.

To insert a new variable among the existing variables, on an existing variable adjacent to where you want to insert the new variable. Click the *Insert* button and a new variable will be created *ahead* of the selected existing variable. This changes the data offsets of all following variables in the exchange and will change the signature major number if you are using signatures.

Once a new variable has been entered, double-click a data field within the row to edit that value.

To delete an existing variable, click on the variable row, and then click the *Delete* button. If you are using signatures, this will cause the *signature major* number to change.

The sum of the length for all variables in the exchange must not exceed 1400 bytes. The total length of the exchange is displayed as *Length (Bytes)*: above the variable list. PACSystems CPUs with firmware version 5.0 and later support a maximum of 30,000 variables for all exchanges. Earlier firmware versions support approximately 12,000 variables for all exchanges.

A variable is automatically created for the local exchange status that is returned to the PLC logic application. The exchange status is not part of the produced exchange data and is not available to the network.

### Configuring an Ethernet Global Data Exchange for a Consumer

To create a new consumed exchange, right-click the “Consumed Exchanges” node and select “New.”

**NOTE:** If the *new* menu option is unavailable, ensure that the Local Producer ID property of the Ethernet Global Data folder is set to a value other than 0.0.0.0.

A dialog box lists all produced exchanges in the EGD network that have been published to the EGD Configuration Server. Select the exchange to be consumed. Once selected, the exchange is populated with the variable, length, type, and description information defined in the producer. The variable name consists of the target name, an underscore, and the variable name in the producer. (See the table below for information about name generation.) You must either enter a reference address or select *ignore* for each variable in the exchange. You must also assign an adapter name and a timeout for the exchange. With these steps, the configuration of the consumer is complete.

When an individual consumed exchange is selected, the following parameters can be configured in the Properties Inspector window. Typically, only the adapter name and the update timeout need to be specified for the exchange and the reference address specified for the variables in the exchange. Changing any other values in a consumed exchange should only be done with expert help.

<b>Name</b>	A name was assigned for this exchange. Defaults to the target name of the producer, an underscore, and the exchange ID in the producer. Changing the name (or its syntax) may prevent resynchronization of the variable.
<b>Producer ID</b>	The ID of the PLC producing the exchange. Producer ID is defined by the producer. Changing the Producer ID may prevent resynchronization with the server.
<b>Group ID</b>	Used only if the produced exchange has been configured with a Destination Type of Multicast. The group ID is defined by the producer. Changing the Group ID may prevent the producer from consuming the data.
<b>Exchange ID</b>	Identifies a specific data exchange to be received by the consuming device. Exchange ID is defined by the producer. Changing the Exchange ID may prevent resynchronization with the server.
<b>Adapter Name</b>	The Adapter Name is the name of a specific Ethernet interface, which is identified by its rack and slot location within the consuming PLC.
<b>Consumed Period</b>	This is a non-editable field. It will always display 200 milliseconds.
<b>Update Timeout</b>	A value in the range of zero to 3,600,000 milliseconds (1 hour). The Ethernet interface will declare a refresh error if the first or subsequent packet of data does not arrive within this time. The Update Timeout should be at least double the value of the producer period and should allow for transient network delays. The default, zero, indicates no timeout. Resolution is in two milliseconds (2ms) increments.



<p><b>Run Mode Store Enabled</b></p>	<p>When set to True, the Run Mode Store Enabled parameter allows the user to modify or delete this exchange and store the changes while in Run mode. You can add exchanges in Run mode regardless of the setting of this parameter.</p> <p>It is recommended that you keep this parameter at its default setting, False, unless your application has a specific need to modify this exchange in Run mode.</p>
--------------------------------------	---

## Name Generation for Consumed Variables

Consumed variables are created automatically. They are based on the variable name in the producer. The name consists of up to seven characters of the beginning of the target name of the producer followed by an underscore character “\_” followed by up to 21 characters of the beginning of the variable name of the variable in the producer. Since the PLC programming software allows names of up to 32 characters, it is possible that the generated name for a consumed variable will not be unique. This can occur when the target names of producers have the same first seven characters and variable names have the same first 21 characters. When the generated variable is not unique, the variable in the consumer has an underscore character and a two-digit number appended to it to make it unique.

## Synchronizing a Consumed Exchange with Changes in the Producer

If a produced exchange is changed, it is necessary to reflect these changes in the consumers. This can be done very quickly with the EGD configuration server. Once the new definition of the produced exchange has been published to the server, select the consumed exchange in each consumer, right-click and select synchronize to the server. The new definition of the produced exchange will be brought in from the server. Any variables that have been added to the producer must have reference addresses assigned if they are to be used or they must be selected as *ignore*. No other action is necessary for the consumer.

## Validating the EGD for a Device

One advantage of using the EGD configuration server is the ability to validate the EGD configuration before downloading the configuration to the device. If you right-click on the Ethernet Global Data node in the Navigator, you will see a selection for *Bind and Build*. Selecting this menu item causes the EGD definitions for the target to be cross-checked against the definitions in the server. Each consumed exchange is compared to the produced exchange published by the producer. The system will advise of any discrepancies. See the previous table for a list of any errors detected.

It is also possible, by selecting the menu item *Unconsumed Data Report* to generate a report to list any variables in produced exchanges that are not being used by a consumer. Producing data that is not being consumed is not necessarily an error. If producing data is not consumed, the consumer may not be able to publish its information to the EGD configuration server or the application design may have chosen to publish data that is not needed immediately. Each unconsumed variable may be an indication of an error or oversight in one or more consumers in the application.

## Looking at the Entire EGD Network

The EGD Management Tool can be used to display information about the entire EGD network both offline and online to that network. You can launch the EMT by right-clicking on the Ethernet Global Data node in the Navigator and selecting *Launch EGD Management Tool*. The EGD Management Tool will come up in a separate frame. It allows you to visualize, analyze, and debug an EGD network. See Section 12 *Diagnostics*, for more information on the online capabilities of the EMT. Also, see the EMT help for information about running the EMT.

## Configuring EGD Devices Not Supported by the EGD Configuration Server

Some devices, for example, certain Ethernet NIUs cannot be configured using the EGD configuration server. Configuration tools for third-party devices that support Ethernet Global Data may not support the EGD configuration server. Rather than not using the server in applications that contain these devices, there is an alternative that allows the EGD configuration for such devices to be put into the server so that it can be used for consumption and validation in other devices.

The programmer distribution includes a tool called the EGD Generic Device Editor. This tool allows you to describe the EGD configuration of a device and publish it to the EGD configuration server. Configuration tools for other devices can use the EGD configuration published by the EGD Generic Device Editor for consumption or validation purposes.

## Installing the EGD Generic Device Editor

The EGD Generic Device Editor is supplied with PME, but is not installed with the programmer. To install the EGD Generic Editor, insert the PAC Machine Edition software CD (PAC\_MachineEdition\_v9.50.0.7677\_English). Browse to directory *EGD Server and Tools* and locate the Windows Installer file *EgdGenericEditor Setup* Windows installer. Complete the install with the installation wizard.

## Running the EGD Generic Device Editor

Installing the EGD Generic Device Editor adds it to the Start – Programs menu of the computer’s Windows system. You will find it under Programs - GE Industrial Systems-EGD Generic Editor. The Windows help for this tool describes its operation.

## Using Signatures in Ethernet Global Data

EGD signatures can be used to make sure that the format of the data from the producer matches what is expected by the consumer. The EGD signature is a numeric value that has two parts: the major number and the minor number.

An EGD Signature has the format <i>MAJ.MIN</i>	
Major Number (MAJ)	Reflects the primary format of the data
Minor Number (MIN)	Reflects backward-compatible changes made to the EGD exchange (such as adding data to the end of the exchange)

For the consumer to consume the data, the signature major number must match between the producer and the consumer for the consumer to consume the data. Packets that are received when produced and consumed exchange signatures are enabled and incompatible (different major signature values) will result in an error consumed exchange status.

## Major Number

The primary format of the data is first established when the EGD exchange is defined. *By default, the signature is assigned the value of 1.0.* Any change that reorders, removes, renames, or changes the type or offset of a variable in the exchange is a primary format change that causes the signature major number to be incremented.

## Minor Number

The signature minor number is incremented when backward-compatible changes are made in the format of the produced data. Backward-compatible changes are made by adding data to unused areas of the exchange including adding data to the end of the exchange.

If the signature minor number in a sample is greater than the signature minor number configured for the exchange in the consumer, then the consumer will consume the data truncating any unexpected data at the end of the sample. (The consumer can do this because the minor number change guarantees that only backward-compatible changes have been made in the format of the data.)

## Incompatible Signatures

If the signature of a produced exchange is specified as zero, then consumers will not check it. If the signature of a consumed exchange is configured as zero, then any signature from a producer will be accepted and the data used if the length of the data exactly matches the expected length.

Use of signatures is enabled by default for new RX3i projects and is disabled for other targets and existing projects.

Only the PACSystems RX3i support non-zero signatures. All other targets force the signature for both produced and consumed exchanges to be zero.

## Using Signatures with Run Mode Stores of EGD

If your application will use run mode stores of EGD, the use of signatures is highly recommended. Do not use EGD commands specifying a signature value of zero because a value of zero effectively disables the signature checking function. For information about the use of signatures with run mode stores of EGD, refer to Section 5.6.1, *Run Mode Store of EGD*.

## Configuring EGD Signatures

To select the signature option for a device, right-click the Ethernet Global Data node and choose Properties. The Use Signatures option is displayed in the Properties Inspector window. This parameter may be set to True to enable signature support or to False to disable signature support in the device.

**Note:** Note that both the producer and consumer must have signatures enabled or signatures are ignored. If signatures are ignored, only the exchange size is used to determine compatibility.

## Configuring Ethernet Global Data without Using the EGD Configuration Server

If the EGD Configuration Server is not used, each Ethernet Global Data exchange must be configured in both the producer and the consumer. To add exchanges, expand the Ethernet Global Data node in the Project tab. Right-click the Consumed Exchanges or the Produced Exchanges node and choose New. The new exchange appears under the selected list node.

1. For each Consumed and Produced Exchange, configure the parameters described here.

- To specify the variable ranges for each exchange, right-click the exchange and choose Configure Ranges. The EGD Variable Range Editor window opens.

## Configuring an Ethernet Global Data Exchange for a Producer

The information to be sent by the producer and the exchange details are defined in the Properties for each Produced exchange (also called a “page”).

When an individual produced exchange is selected, the Properties inspector window permits user configuration of the following information:

<b>Name</b>	A name is assigned for this exchange. Defaults to the target name of the producer, an underscore, and the exchange ID in the producer. Changing the name (or its syntax) may prevent resynchronization of the variable.
<b>Exchange ID</b>	Identifies a specific data exchange to be received by the consuming device. Exchange ID is defined by the producer. Changing the Exchange ID may prevent resynchronization with the server.
<b>Adapter Name</b>	The Adapter Name is the name of a specific Ethernet interface, which is identified by its rack and slot location within the consuming PLC.
<b>Destination Type</b>	Specifies whether the data’s destination will be: <ul style="list-style-type: none"> <li>▪ An IP address (Unicast)</li> <li>▪ A Group ID (Multicast)</li> <li>▪ All EGD nodes on the subnet (Broadcast IP).</li> </ul>
<b>Destination</b>	Identifies the data’s consuming device, based on the Destination Type selected: <ul style="list-style-type: none"> <li>▪ a dotted-decimal IP address if Destination Type is IP Address</li> <li>▪ the group’s ID (1–32) if Destination Type is Group ID</li> <li>▪ the value 255.255.255.255 if Broadcast IP is the Destination Type.</li> </ul>
<b>Produced Period</b>	The scheduled repetition period at which the data is produced on the network. Configure a value in the range of zero or 2–3,600,000 (2 milliseconds to 1 hour). The value zero means at the end of the next PLC scan, but not less than two milliseconds (2 ms) from the previous production. Set the production period to ½ the period at which the application needs the data in this exchange. Round this value to the nearest 2 milliseconds.
<b>Send Type</b>	Fixed at <i>Always</i> . In the PLC, production of EGD is controlled by the I/O state: when enabled, EGD production is enabled, and when disabled, EGD production is disabled.
<b>Reply Rate</b>	Not used.
<b>Run Mode Store Enabled</b>	When set to True, the Run Mode Store Enabled parameter allows the user to modify or delete this exchange and store the changes while in Run mode. You can add exchanges in Run mode regardless of the setting of this parameter.  It is recommended that you keep this parameter at its default setting, False, unless your application has a specific need to modify this exchange in Run mode.

Double-clicking on the produced exchange opens a window for configuring the variables within the exchange. Each exchange has its own variable list. These variables contain the data that is produced for the network. Each variable contains the following information:

<b>Offset (Byte.Bit)</b>	The location within the data area for this exchange where the start of the data for this variable is located. The offset is expressed as <i>Byte.Bit</i> , where <i>Byte</i> is a zero-based byte offset and <i>Bit</i> is a zero-based bit position within that byte. (Valid bit values are 0-7. Bit 0 is the least-significant bit within the byte; bit 7 the most significant.)
<b>Variable</b>	The name is defined for this variable.
<b>Ref Address</b>	The PLC memory reference address contains the start of the data for this variable.
<b>Ignore</b>	Not used for Produced exchange.
<b>Length</b>	Size of the data for this variable expressed in units of the selected PLC reference memory type.
<b>Type</b>	The data type of the selected PLC reference memory type. (Automatically set up by the Ref Address selection.)
<b>Description</b>	An optional text description of this variable.

To add a new variable to the end of the exchange, click the *Add* button. This does not change the data offsets of any existing variables within that exchange.

To insert a new variable among the existing variables, click on an existing variable. When you click the *Insert* button, a new variable will be created *ahead* of the selected existing variable. This changes the data offsets of all subsequent variables in the exchange.

Once a new variable has been entered, double-click a data field within the row to edit that value.

To delete an existing variable, click on the variable row and then click the *Delete* button.

The sum of all variables in the exchange must not exceed 1400 bytes. The total length of the exchange (in bytes) is displayed as *Length (Bytes)*: at the top of the exchange window above the variable list. PACSystems CPUs with firmware version 5.0 and later support a maximum of 30,000 variables for all exchanges. Earlier firmware versions support approximately 12,000 variables for all exchanges.

A variable is automatically created for the required Status variable. This variable contains the local exchange status that is returned to the PLC logic application. The exchange status is not part of the produced exchange data and is not available to the network.

## Configuring an Ethernet Global Data Exchange for a Consumer

The exchange details are defined in the Properties for each Consumed exchange.

When an individual consumed exchange is selected, the Properties inspector window permits user configuration of the following information:

<b>Name</b>	A name is assigned for this exchange. Defaults to the target name of the producer, an underscore, and the exchange ID in the producer. Changing the name (or its syntax) may prevent resynchronization of the variable.
<b>Producer ID</b>	The PLC producing the exchange. This value, conventionally expressed as a dotted-decimal number uniquely identifies the Ethernet Global Data device across the network.
<b>Group ID</b>	Used only if the produced exchange has been configured with a Destination Type of Group ID. This Group ID (1-32) must match that of the producer.
<b>Exchange ID</b>	Identifies a specific data exchange to be received by the consuming device. It must match the Exchange ID specified in the produced exchange.
<b>Adapter Name</b>	The Adapter Name is the name of a specific Ethernet interface, which is identified by its rack and slot location within the consuming PLC.
<b>Consumed Period</b>	Not used in PACSystems. (Always displayed as 200 milliseconds; not editable.)
<b>Update Timeout</b>	A value in the range of zero to 3,600,000 milliseconds (1 hour). The Ethernet interface will declare a refresh error if the first or subsequent packet of data does not arrive within this time. The Update Timeout should be at least double the value of the producer period and should allow for transient network delays. The default, zero, indicates no timeout. Resolution is in two milliseconds (2ms) increments.
<b>Run Mode Store Enabled</b>	When set to True, the Run Mode Store Enabled parameter allows the user to modify or delete this exchange and store the changes while in Run mode. You can add exchanges in Run mode regardless of the setting of this parameter.  It is recommended that you keep this parameter at its default setting, False, unless your application has a specific need to modify this exchange in Run mode.

Double-clicking on the consumed exchange opens a window for this exchange for configuring the variables within the exchange. Each exchange has its own variable list. These variables contain the data that is consumed from the network. Each variable contains the following information

<b>Offset (Byte.Bit)</b>	The location within the data area for this exchange where the start of this data for this variable is located. The offset is expressed as <i>Byte.Bit</i> , where <i>Byte</i> is a zero-based byte offset and <i>Bit</i> is a zero-based bit position within that byte. (Valid bit values are 0-7. Bit 0 is the least-significant bit within the byte; bit 7 the most significant.)
<b>Variable</b>	The name is defined for this variable.
<b>Ref Address</b>	The PLC memory reference address contains the start of the data for this variable. For consumed exchanges, %S memory types and override references are not allowed. (This field is non-editable when the Ignore selection is set to True.)
<b>Ignore</b>	Allows a consumer to ignore this variable. Setting Ignore to True means this variable is not sent to the PLC reference table. Defaults to False.
<b>Length</b>	The size of the data for this variable is expressed in units of the selected PLC reference memory type.
<b>Type</b>	The data type of the selected PLC reference memory type. (Automatically set up by the Ref Address selection.)
<b>Description</b>	An optional text description of this variable.

To add a new variable to the end of the exchange, click the *Add* button. This does not change the data offsets of any existing variables within that exchange.

To insert a new variable among the existing variables, click on an existing variable. When you click the *Insert* button, a new variable will be created *ahead* of the selected existing variable. This changes the data offsets of all subsequent variables in the exchange.

Once a new variable has been entered, double-click a data field within the row to edit that value.

To delete an existing variable, click on the variable row, and then click the *Delete* button.

The sum of all variables in the exchange must not exceed 1400 bytes. The total length of the exchange (in bytes) is displayed as *Length (Bytes)*: at the top of the exchange window above the variable list. PACSystems CPUs with firmware version 5.0 and later support a maximum of 30,000 variables for all exchanges. Earlier firmware versions support approximately 12,000 variables total for all exchanges.

A variable is automatically created for the required Status variable. This variable contains the local exchange status that is returned to the PLC logic application. The exchange status is not part of the consumed exchange data.

A variable is automatically created for the optional Timestamp variable. This variable contains the timestamp of the last received data packet (generated when the exchange was produced) that is returned to the PLC logic application. Set the Ref Address to NOT USED to ignore the timestamp variable.

Any consumed data variable may be ignored by setting the Ignore selection to True. See Selective Consumption, below.

**Note:** If the total data length of a consumed exchange does not match the length of the produced exchange received from the network, PLC Faults and Ethernet exceptions will occur.

## Selective Consumption

Not all data ranges within a produced exchange need to be consumed by each PLC. For example, a producer is producing an exchange consisting of a 4-byte floating-point value, followed by a two-byte integer, followed by a two-byte analog value. If the consuming PLC wants to consume only the analog value and place it into %AI003, the consumer might be configured as shown below.

Offset	Variable	Ref Address	Ignore	Length	Type	Description
0.0		Ignore	True	6	Byte	Ignore float and integer
6.0	Var01	%AI0003		1	WORD	

**Note:** Where EGD signatures are not used the total length of the exchange must be the same in producer and consumer, even if the consumer is ignoring a portion of the exchange. Failure to configure any ignored bytes in the consumed exchange will result in exchange exception log and fault table entries, error status in the exchange status data, and no data is transferred for the exchange.

## EGD Configuration for Redundancy Systems

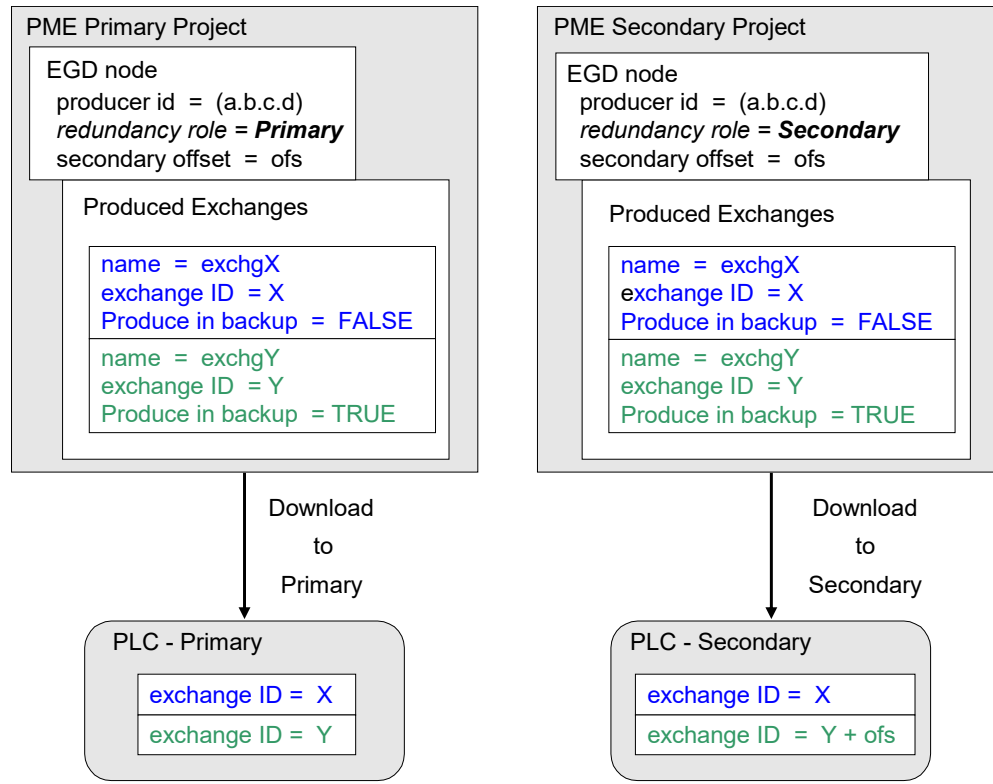
For exchanges that are produced in backup mode, an offset must be added to the Exchange ID. This ensures that the Exchange ID is unique for those exchanges that are produced simultaneously by the active and backup controllers.

The Secondary Produced Exchange Offset parameter is available in the Ethernet Global Data properties when redundancy is enabled and at least one produced exchange is configured to produce in backup mode. A sample EGD configuration with an Exchange ID offset in an Ethernet Redundancy System is illustrated in Figure 33.

Non-HSB targets have an additional Ethernet Global Data property, Redundancy Role, which appears when any Ethernet interface in the system is configured for redundant IP operation. This parameter is used only within the programming software and is not delivered to the PLC. The Redundancy Role parameter is not displayed for HSB systems.



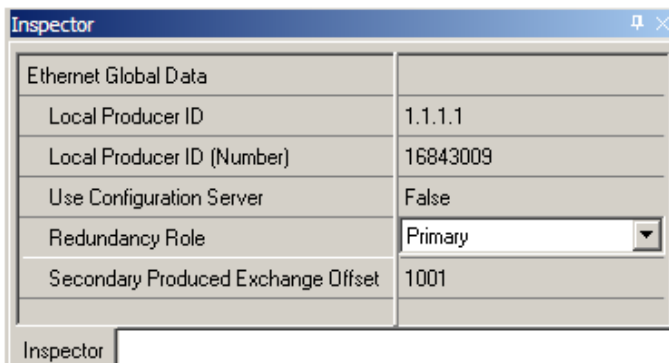
Figure 34: Exchange ID Offset in an Ethernet Redundancy System



### Configuring Redundancy Roles for EGD

To configure a Redundancy Role, open the Inspector pane with the EGD node selected (Figure 35). Repeat this instruction for each affected Project. To further support a redundant configuration, the user should specify that the same offset value is specified for the primary and secondary target projects.

Figure 35: Configuring Redundancy for Ethernet Global Data



### Configuring a Produced Exchange for Redundancy Systems

The *Produce in Backup Mode* parameter appears in the properties for each produced exchange (Figure 36).

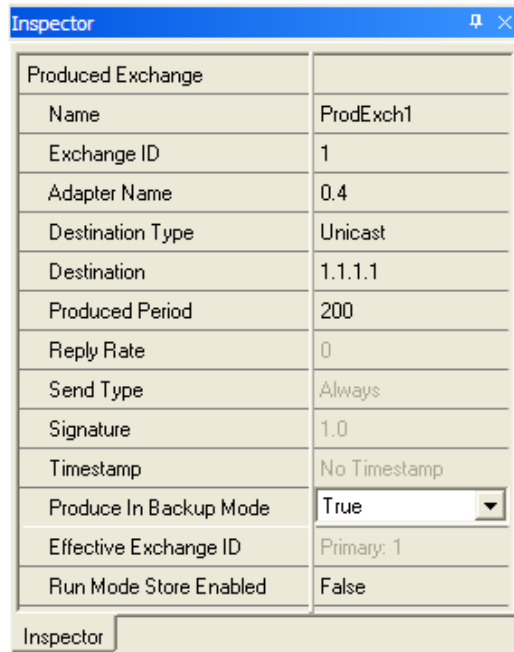
There are two modules for Consuming EGD Exchanges in a Redundant system:

1. **Redundant IP:** The HMI or requesting device can request info from the Ethernet interface on a consistent IP address regardless of whether the Primary or Secondary is set to be the Active Redundant Controller. Produced exchanges are produced with a configured ProducerID and ExchangeID for consumption. The IP address the exchange is produced from should not typically matter to the Consumer. If the consumer is only capable of listening to communications from the Redundant IP address, the **Produce EGD on Redundant IP** setting in the Ethernet interfaces Settings tab should be set to True.

Redundant IP	Enable
Redundant IP Address	10.10.10.10
Produce EGD on Redundant IP	True

2. **Produce in Backup Mode:** Both Redundant Controllers will produce the EGD Exchange on their Ethernet interface’s native IP addresses.

**Figure 36: Configuring Produce in Backup Mode Parameter**



# Section 5 Ethernet Global Data

This chapter describes the basic Ethernet Global Data (EGD) features. The topics covered are:

- Ethernet Global Data Operation
- EGD Exchanges
- The Content of an EGD Exchange
  - The Data Ranges (Variables) in an EGD Exchange
  - Valid Memory Types for Ethernet Global Data
  - Planning Exchanges
  - Using Ethernet Global Data in a Redundancy System
- Sending an Ethernet Global Data Exchange to Multiple Consumers
  - Multicasting Ethernet Global Data
  - Broadcasting Ethernet Global Data
- Ethernet Global Data Timing
  - Configurable Producer Period for an EGD Exchange
  - Consumer Update Timeout Period
  - EGD Synchronization
- Time-stamping for Ethernet Global Data Exchanges
- Effect of PLC Modes and Actions on EGD Operations
- Run Mode Store (RMS) of EGD
- Monitoring Ethernet Global Data Exchange Status

## 5.1 Comparison Model for Ethernet Global Data Support

EGD Capabilities for RX3i Ethernet Interfaces										
	CPE302	CPE305	CPE310	CPE330	CPE400	CPL410	CPE100	CPE115	ETM001-jx	ETM001-Kxxx
EGD 1	X	X	X	X	X	X	X	X	X	X
EGD 2									X	X

EGD Class 1 is configured exchanges with no logic control of EGD operation.

EGD Class 2 is EGD Commands which are logic-driven EGD exchanges using COMMREQs.

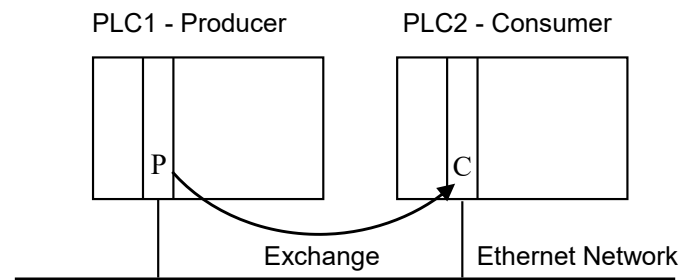
## 5.2 Ethernet Global Data Operation

Ethernet Global Data is data that is automatically sent from one Ethernet device to one or more others. Once Ethernet Global Data has been configured, the data is sent automatically during system operation. No program interaction is necessary to produce or consume the global data.

The device that sends the Ethernet Global Data is called the *producer*. Each device that receives Ethernet Global Data is called a *consumer*. Each unique Ethernet Global Data message is called an *exchange* (also sometimes referred to as a *page*).

An Ethernet interface can be configured to both produce and consume Ethernet Global Data at the same time, using separate exchanges.

**Figure 37: Producing & Consuming Ethernet Global Data**



## 5.2.1 EGD Producer

The producer of an exchange periodically sends new samples of data from its local internal memory. The producer of exchange is uniquely identified by its Producer ID. The Producer ID can be expressed as a dotted-decimal number (for example, 0.0.0.1). Even when expressed in IP address form, it is not used as an IP address. It is used to identify a specific PLC on the network. Since the Producer ID identifies only the PLC producing the exchange, the number of Ethernet interfaces installed will not affect the PLC.

When using the EGD configuration server, each PLC that transfers EGD must be assigned a Producer ID even if that PLC produces no exchanges. The Producer ID uniquely identifies each EGD device in the configuration server and must be present if the server is used.

## 5.2.2 EGD Consumers

A consumer is a device that will update its local internal memory based on the data in exchange. The consumer is identified at the producer by an IP Address, a Group ID, or a Subnet Mask, depending on the Destination Type selected.

The Consumed Exchange configuration allows *Selective Consumption* of a produced EGD exchange. *Selective Consumption* occurs when the consumer takes in the whole exchange from the network but does not send all the exchange data to the PLC memory. A Consumed Exchange can be set to ignore the data ranges (variables) that are not needed.

## 5.3 EGD Exchanges

EGD Exchange Support for RX3i Ethernet Interfaces										
	CPE302	CPE305	CPE310	CPE330	CPE400	CPL410	CPE100	CPE115	ETM001-Jx	ETM001-Kxxx
<b>Number of EGD 1 Exchanges</b>	255	255	255	255	255	255	8	8	255	255

Each exchange in EGD is identified by its Producer ID and Exchange ID. Up to 255 exchanges can be configured for a PACSystems Ethernet interface. They can be divided into any combination of produced and consumed exchanges. Each exchange can be up to 1400 bytes in length.

Some produced exchanges can accept partial data when exchanges are produced at different rates or sent to different consumers. Consumed Exchanges should not duplicate where the data is put as variable conflicts will occur and data will be overwritten by the multiple exchanges.

## CAUTION

Ethernet Global Data is designed for simple, efficient communication of sampled data between devices. It is not intended for event notification where the possible loss of a sample of data would be significant.

Some EGD devices support EGD pages. An EGD page consists of one or more exchanges that are produced on the same schedule to the same destination. Pages remove the 1400-byte size limitation of EGD exchanges. PME does not currently show information about EGD pages, but the user may see the constituent exchanges for each page.

### 5.3.1 Content of an Ethernet Global Data Exchange

Each Ethernet Global Data exchange is composed of one or more data ranges transmitted as a sequence of 1 to 1400 bytes of data. The data ranges are commonly called variables; they may be configured to correspond to PLC variables. The content of the data is defined for both the producer and consumers of the data. In this example, a producer sends an 11-byte exchange consisting of the current contents of %R00100 through %R00104 followed by the current contents of %I00257 through %I00264. The same exchange can be configured for each consumer to suit the needs of the application.

Address	Length	Type	Description
%R00100	5	WORD	Conveyor1 in PLC1
%I00257	1	BYTE	Conveyor1 limit switch in PLC1

### 5.3.2 Data Ranges (Variables) in an Ethernet Global Data Exchange

The variables within an exchange are defined in the Ethernet Global Data configuration in *Hardware Configuration*. There can be:

- A length of 1 byte to 1400 bytes per exchange. The total size of an exchange is the sum of the data lengths of all the data ranges configured for that exchange.
- A maximum of 30,000 data ranges for all exchanges in the target, for CPUs with firmware version 5.0 or later. (Earlier firmware versions allow approximately 12,000 EGD data ranges per target.)

Different produced exchanges may share some or all the same data ranges even if the exchanges are produced at different rates. A consumer does not have to consume all the data from a produced exchange. A consumed exchange may be configured to ignore specified data ranges.

### 5.3.3 Valid Memory Types for Ethernet Global Data

The PLC memory types listed below can be included in EGD exchanges. Discrete point references such as %I or %Q are configured as Byte-Array, Word-Array, or Dword-Array variables. That means a

variable with discrete point references must be defined in blocks of 8 points if it is defined as a Byte-Array, 16 points if Word-Array, and 32 points if Dword-Array. Discrete memory must be byte-aligned. Boolean type and Boolean-Array variables are not allowed.

Memory Type	Description	P-Producer C-Consumer
%R	Register memory in word mode	P/C
%W	Word memory in word mode	P/C
%AI	Analog input memory in word mode	P/C
%AQ	Analog output memory in word mode	P/C
%I	Discrete input memory in byte mode	P/C
%Q	Discrete output memory in byte mode	P/C
%T	Discrete temporary memory in byte mode	P/C
%M	Discrete momentary memory in byte mode	P/C
%SA	Discrete system memory group A in byte mode	P/C
%SB	Discrete system memory group B in byte mode	P/C
%SC	Discrete system memory group C in byte mode	P/C
%S	Discrete system memory in byte mode	P
%G	Discrete global data table in byte mode	P/C
Symbolic Variables	Symbolic variables	P/C

To use a symbolic variable in an EGD exchange, it must exist in the Variables definition for the target. To add it to an exchange, double click the Variable field to open a selection dialog box as illustrated in Figure 38.

Figure 38: Adding Symbolic Reference to Ethernet Global Data Exchange

<span>Add</span> <span>Insert</span> <span>Delete</span> Length (Bytes): 22					
Offset (Byte.Bit)	Variable	Ref Address	Ignore	Length	Type
Status		%I00081	False	16	BIT
TimeStamp		NOT USED	False	0	BYTE
0.0		%R00001	False	10	WORD
20.0	Sym_1	<Symbolic>	False	1	INT

## 5.3.4 Planning Exchanges

It is possible to configure more Ethernet Global Data than a PLC can transfer (especially on 10Mbit networks). If high levels of consumer timeouts occur in some or all the consumed exchanges, the EGD load can be reduced by:

- Increasing the production period (especially if the period is more frequent than double the minimum time in which the data is needed).
- Defining fewer exchanges, each with more data.
- Using EGD groups or broadcasting to subnets. Rather than producing a directed exchange to several destinations, a single exchange can contain all the data and each consumer can transfer only the data it needs from the exchange.
- Adding another Ethernet interface module to the rack and spreading the EGD exchanges.

## 5.3.5 Using Ethernet Global Data in a Redundancy System

When configured for Redundant IP operation, the active unit produces all EGD exchanges to the network. The backup unit produces only EGD exchanges that have their *Produce in Backup Mode* property set to *True*. When the active Ethernet interface changes to the backup, it stops the production of all EGD exchanges except those that are configured to produce in backup mode.

When configured for Redundant IP operation, the active and backup Ethernet interfaces should be configured to consume EGD exchanges via multicast host groups or the local subnet broadcast address. Redundancy roles permit both the active and backup units to receive the latest data from the network. Unicast operation is not recommended. The backup unit does not consume exchanges at the Redundant IP address.

For additional information about redundancy systems, refer 1.3.3, *Ethernet Interface Specifications*.

## 5.4 Sending an Ethernet Global Data Exchange to Multiple Consumers

There are two ways to send an EGD Exchange to multiple consumers at the same time: by Multicasting it to a predefined group of consumers or by Broadcasting it to all the consumers on a subnet. Both methods allow many consumer devices to simultaneously receive the same data from one producing EGD device. If an exchange is a Broadcast or Multicast, the same exchange must be configured at the producer and at each consumer. Each consumer can use all the data or just a selected portion, as configured for the consumed exchanges.

For more information about Multicasting and Broadcasting, refer to Section 13, *Network Administration*.

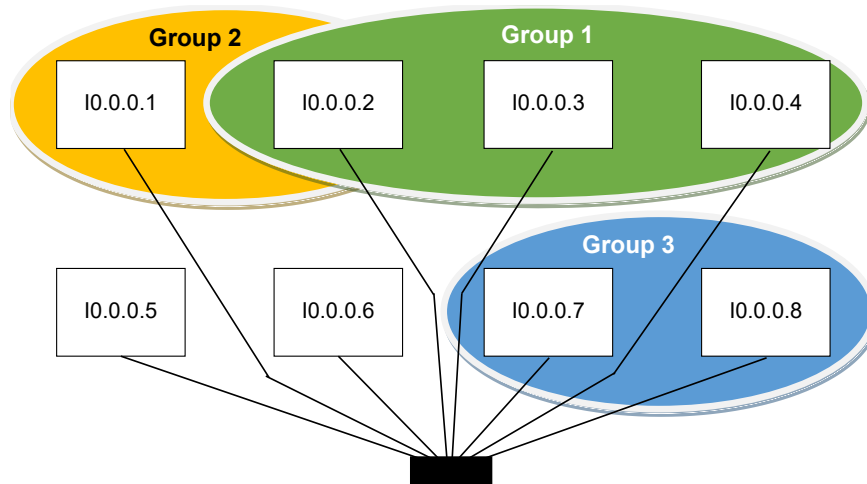


## 5.4.1 Multicasting Ethernet Global Data

If more than one device on the network should consume a Global Data exchange, those devices can be set up as a group. The network can include up to 32 numbered groups. Groups allow each sample from the producer to be seen simultaneously by all consumers in the group.

A device can belong to more than one group, as illustrated below. In the following example, device 10.0.0.2 consumes exchanges from Group 2 and Group 1. Each device in a group responds to the group's assigned ID number from 1 to 32.

**Figure 39: Grouping of Devices for Ethernet Global Data Multicasting**



**Note:** Each device on the network using EGD should have a unique local producer ID. If the devices using multicast EGD do not have unique local producer IDs, unexpected results can occur when using group addressing for EGD exchanges.

Each Group ID corresponds to a Multicast (Class D) IP address reserved by the Internet authorities. The default Multicast IP addresses used by Ethernet Global Data are:

Group ID	IP Address	Note
1	224.0.7.1	For EGD class1 on Embedded Ethernet interface of CPE302/CPE305/CPE310, Multicast network can include only up to 31 numbered groups.
2	224.0.7.2	
.	.	CPE330/CPE400/CPL410/CPE100/CPE115 Embedded Ethernet interface supports Multicast groups 1 – 32.
32	224.0.7.32	

**Note:** CPE330/CPE400/CPL410/CPE100/CPE115 do not support the AUP file. The configurable AUP parameters for these CPUs are part of the Hardware Configuration for the corresponding Embedded Ethernet interface in PME.

Group Multicast IP Addresses used by Ethernet Global Data should not be changed unless the defaults would cause a network conflict. If necessary, they can be changed within the reserved range of multicast IP addresses (224.0.0.0 through 239.255.255.255). The change must be made using an Advanced User Parameter File.

## 5.4.2 Broadcasting Ethernet Global Data

The same Ethernet Global Data exchange can be sent to all the consumers on a subnet by configuring the Produced Exchange to use a Destination Type of “Broadcast.” The “Destination” of that exchange then changes to the value 255.255.255.255. (The Ethernet interface converts this value to the appropriate subnet broadcast mask for this network.) As with a Group ID, each consumer on the subnet can be configured to use some or all of the exchange.

## 5.4.3 Changing Group ID in Run Mode

With the ability to perform a run-mode store of EGD, it is possible to change the Group ID or Destination Type of a produced or consumed exchange at run-time. The effects of such changes will depend upon the configurations of the local PLC *and* other devices on your network.

### Broadcast

Changing the Destination Type of a produced exchange from unicast or multicast to broadcast causes samples to be sent to all nodes on your network. Samples are subsequently processed if the local device has a consumed exchange configured with matching Producer ID and Exchange ID. Otherwise, they are ignored.

### Multicast

Changing the Destination Type of a produced exchange from unicast or broadcast to multicast causes samples to be sent to a subset of the nodes on your network. Samples are visible to all devices on the network that has *any* exchange(s) configured to consume from the specified Group ID. Samples are subsequently processed only if the local device has a consumed exchange configured with matching Producer ID and Exchange ID.

This means that modifying a multicast exchange so that it produces to a different Group ID may or may not affect its consumption. If the remote device has any exchanges configured to consume from the new producer ID, consumption will not be interrupted. However, consumption will be affected if the remote device is not configured to consume any exchanges from the new Group ID. In the latter case, updates to the consumed exchange configuration will be necessary to resume consumption.

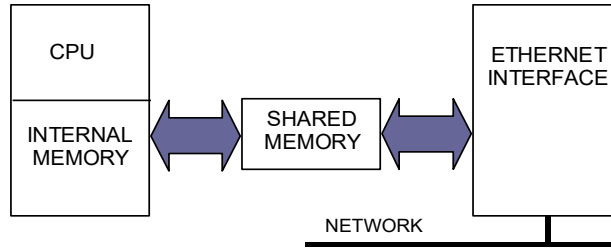
### Unicast

Transitioning from a multicast or broadcast exchange to unicast production causes samples to be sent to a single node. Thus, the exchange will now only be visible to a single remote node and processed only if that node contains a consumed exchange with matching Producer ID and Exchange ID.

## 5.5 Ethernet Global Data Timing

The Ethernet interface and PLC CPU share internal memory for Ethernet Global Data operations.

**Figure 40: Memory Sharing between PLC and Ethernet interface**



In the producing PLC, the CPU updates its shared internal memory with a data sample when requested by its Ethernet interface. The update affects the length of the PLC sweep only for that specific exchange and will have little effect on the PLC average sweep time. When the Ethernet interface's producer period expires, it produces the data sample from shared internal memory onto the network.

In a consuming PACSystems PLC, shared internal memory is updated as soon as the Ethernet interface gets a data sample from the network. There is no user-configurable consumer period. The CPU updates its reference tables from shared internal memory at the end of the sweep after it is notified by the Ethernet interface that fresh data has arrived for a specific exchange. The data is made available to the application on the next PLC sweep after it is received. Some other types of Ethernet interfaces implement a consumption period timer.

### 5.5.1 EGD Synchronization

Ethernet Global Data attempts to provide the most up-to-date process data, consistent with the configured schedule.

#### CPU Sweep Time

The Ethernet interface maintains a timer for each produced exchange. When the timer for the exchange expires, the Ethernet interface requests that the data for the exchange be transferred from reference memory during the output scan portion of the CPU sweep. At the output portion of the sweep, the CPU puts the data into the shared memory. Once the data has been transferred by the CPU sweep, the Ethernet interface immediately formulates a sample and transfers the sample to the network. (If updated data is not available at the next production timer expiration, the Ethernet interface produces a sample containing the previous data to the network.)

#### Best Scheduling Practices

The result of this scheduling method for Ethernet Global Data is a variability of up to one producer CPU sweep time in the interval between samples produced on the network. This variability in the time between samples is present to assure that the most up-to-date data is being transferred.

In general, it is neither useful nor necessary to configure the production period to be less than the time it takes for the CPU to complete a sweep. If the producer period for an exchange is set lower than the CPU sweep time, the Ethernet interface will send a stale sample (a sample containing the same data as previously sent) at the configured interval. When the latest CPU data becomes available at the end of the sweep, the Ethernet interface will immediately send another sample with the latest data. The timer of the produced exchange is not reset when this sample is sent.

This can result in more samples in the network than would be expected from the configured period.

## 5.5.2 Configurable Producer Period for an EGD Exchange

The Producer period for an EGD exchange can be two milliseconds (2 ms) to one hour. In the PLC, the Ethernet interface attempts to produce the data at this interval. The exchange production may vary from the configured interval by up to one production period or one producer CPU sweep period, whichever is reported sooner.

The producer period is configurable in increments of 2 ms. If the Producer Period is set to zero, production is scheduled for every scan or every 2 ms, whichever is slower. In a PLC with rapid scan times, scheduling a produced exchange at zero results in a very high load on the network and on the Ethernet interface, which can degrade overall Ethernet performance. Scheduling multiple exchanges for a zero period in a PLC with a low scan time can result in the Ethernet interface is unable to produce all the required data and will degrade SRTP communication.

## 5.5.3 Consumer Update Timeout Period

For each consumed exchange, an Update Timeout period can be configured. It determines how long the Ethernet interface will wait for the starting or subsequent packet of data in the exchange before declaring a refresh error. The update timeout period for the consumer should be set to at least twice the producer period. At very small producer periods, the update timeout should also allow for network transfer variation. Otherwise, the PLC may occasionally falsely report refresh faults. Use zero for the update timeout period of a consumed exchange to disable timeout detection.

### Producer Period Guidelines for PLCs

Do not produce and consume data faster than is required by your application. This reduces the load on the network and the devices, providing capacity for other transfers.

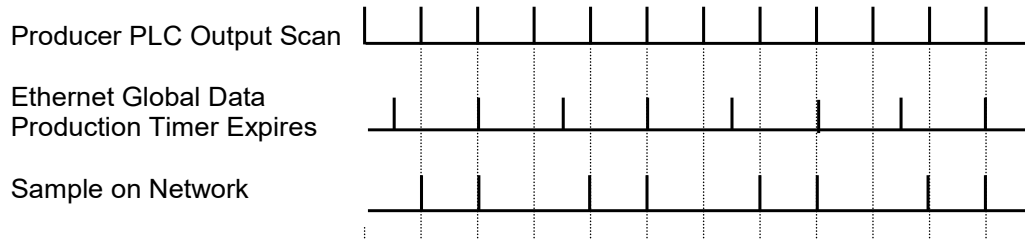
The following illustrations (Figure 41 and Figure 42) show the relationship between the PLC output scan time, the produced exchange timer, and data samples on the network.

### Timing Example 1

Only one sample is produced on the network per producer period expiration. The time between samples can vary up to the producer CPU sweep time.

**Figure 41: EGB Timing Example #1**

**Producer Period Expiration = 1.5 Times per CPU Sweep**

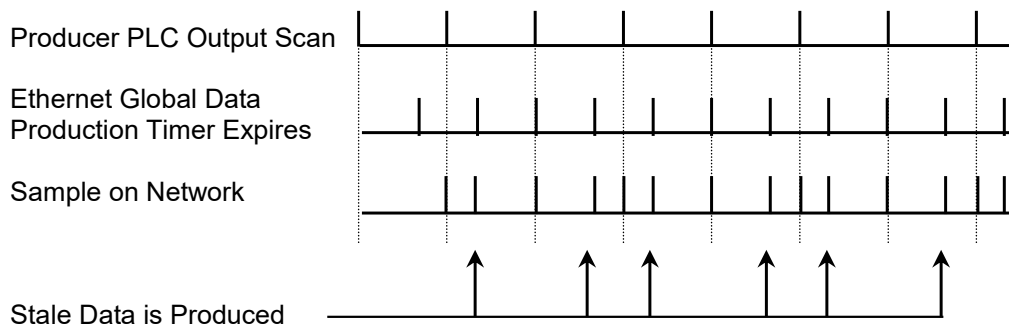


### Timing Example 2

More than one sample can be produced per producer period expiration and stale samples are produced to the network.

**Figure 42: EGB Timing Example #2**

**Producer Period Expiration = 2/3 the Time of CPU Sweep**



## 5.6 Effect of PLC Modes and Actions on EGD Operations

The configuration and operation of Ethernet Global Data may be affected by the PLC's current mode and by certain PLC actions:

- The normal PLC mode for EGD operation is RUN with Outputs enabled. In this PLC mode, Ethernet Global Data remains configured and exchanges are both produced and consumed.
- If the PLC mode is set to STOP with I/O disabled, the Producer ID remains configured, but production and consumption stops. Note that while consumed data is not transferred to the PLC memory in this mode, data from the network is still transferred to the shared memory so that the latest data is available immediately when the PLC transitions out of STOP with I/O disabled mode.
- If the configuration is lost, the Ethernet Global Data configuration must be stored again.

PLC Mode or Action	Producer ID remains configured	Configuration-Based Exchanges continue to be		
		Configured	Produced	Consumed
<b>PLC Mode</b>				
RUN-Outputs Enabled	YES	YES	YES	YES
RUN-Outputs Disabled	YES	YES	NO	YES
RUN-SUSPEND I/O <sup>11</sup>	YES	YES	YES	YES
STOP-I/O Enabled	YES	YES	YES	YES
STOP-I/O Disabled	YES	YES	NO	NO
<b>PLC Action</b>				
RUN-Store Logic	YES	YES	YES	YES
STOP-Store Logic	YES	YES	<sup>12</sup>	<sup>12</sup>
STOP-Clear Logic	YES	YES	<sup>12</sup>	<sup>12</sup>
STOP-Config Store	Replaced <sup>13</sup>	Replaced <sup>11</sup>	NO <sup>13</sup>	NO <sup>13</sup>
STOP-Clear Config	NO	NO	NO	NO
PLC Power Cycle	YES	YES	<sup>12,14</sup>	<sup>12,14</sup>
Ethernet Interface Restart	YES	YES	<sup>12,14</sup>	<sup>12,14</sup>

<sup>11</sup> RUN-SUSPEND I/O refers to the SUSIO logic function. (The DOIO logic function does not affect EGD production or consumption.)

<sup>12</sup> Production and consumption is controlled by the PLC Mode as described above.

<sup>13</sup> Producer ID and exchange definitions are replaced.

<sup>14</sup> Producer ID and exchange states depend on the PLC mode and configuration prior to the action.

## 5.6.1 Run Mode Store of EGD

### CAUTION

Modifying an exchange using an RMS can cause an interruption in the transfer of EGD data or possibly take the exchange offline. This is particularly a concern for exchanges used with remote IO, such as exchanges between the CPU and NIU. Do not use this feature unless you are sure you understand the possible results.

The user can modify EGD exchanges in a running controller without first transitioning to stop mode. Each exchange can be configured individually to allow or disallow changing or deleting the exchange in run mode. You can add exchanges in run mode without changing any configuration settings.

### Added Exchanges

Added exchanges begin consumption/production shortly after the activation of any logic that is part of the run mode store sequence.

### Deleted Exchanges

Deleted exchanges cease consumption/production shortly before the activation of any logic that is part of the run mode store.

### Modified Exchanges

Modified exchanges will be offline for a short time during the activation of the new logic that is part of the RMS. This amount of time depends on factors, such as sleep mode and sweep time. All variables associated with a modified exchange will hold their last state during the pause in consumption. The consumption timeout is restarted for each modified consumed exchange.

### Run Mode Store and PLC Sweep Times

The effect a run mode store has on PLC sweep times depends on communication window configuration and the magnitude of the changes in the run mode store. Depending on the application's configuration, modifying exchanges in a producer with increased sweep times may cause consumption timeouts on exchanges that are modified in applications with very low tolerances.

If the modification creates an incompatibility between the producer and consumer, the exchange will cease to be consumed.

Any modification to an exchange's parameters resets the *stat g* station manager data for that exchange.

## Modifying an Exchange’s Parameters

The parameters that define the exchange can be modified in a run mode store. Changing some parameters such as Exchange ID essentially redefines the exchange. *This is the equivalent of deleting an existing exchange and adding a new exchange in a single run mode store.* These changes affect signature compatibility with the associated producer or consumer(s). Changing other parameters simply alters the operation of an existing exchange and does not affect compatibility.

For details on the use of signatures to determine compatibility, refer to Section *Using Signatures in Ethernet Global Data*.

### Common EGD Parameters

Parameters that are shared among all exchanges cannot be modified during an RMS. These parameters are properties of the Ethernet Global Data folder in the target.

Parameter	Behavior
Local Producer ID	This setting cannot be changed in a run mode store.
Use Signatures (only available when Configuration Server is used)	This setting cannot be changed in a run mode store.
Secondary Produced Exchange Offset	(Redundancy systems only.) This setting cannot be changed in a run mode store.
Redundancy Role	(Redundancy systems only.) This setting cannot be changed in a run mode store.



## Effects of Modifying Consumed Exchange Parameters

For consumed exchanges, the combination of Producer ID and Exchange ID uniquely identifies the exchange. Modifying any of these parameters will make the exchange incompatible and require an update to the producer to restore compatibility.

Parameter	Behavior
Producer ID	Redefines the exchange. Causes a major signature change in the producer. The exchange will be incompatible.
Group ID	Determines the producer of the exchange and may affect compatibility. For details, refer to Section 5.4, <i>Sending an Ethernet Global Data Exchange to Multiple Consumers</i> .
Exchange ID	Redefines the exchange. Causes a major signature change in the producer. The exchange will be incompatible.
Adapter Name	<p>Deletes an exchange from one Ethernet module and adds an exchange to another. Assuming no other parameters change, this will not affect compatibility</p> <p>To any EGD Class 2 device sending commands that operate on this exchange, it will appear that the exchange has been deleted. The Class 2 device must be updated to direct the commands to the IP address of the adapter where the exchange has been moved.</p>
Update Timeout	Modifies existing exchange. Does not affect compatibility. Note that decreasing a consumed exchange's update timeout without updating the corresponding producer's production period may cause timeouts.

## Effects of Modifying Produced Exchange Parameters

Parameter	Behavior
Exchange ID	Redefines the exchange. Causes a major signature change in the producer.
Adapter Name	Deletes an exchange from one Ethernet module and adds an exchange to another. Assuming no other parameters change, modifying this parameter does not affect compatibility.
Destination Type	Determines the consumer(s) of the exchange and may affect compatibility. For details, refer to Section 5.4, <i>Sending an Ethernet Global Data Exchange to Multiple Consumers</i> .
Destination	Determines the consumer(s) of the exchange. Affects compatibility.
Produced Period	Modifies the existing exchange. Does not affect compatibility.
Produce In Backup Mode	<p>If the unit is in backup mode, modifying this parameter will cause the production of the exchange to start if being set to TRUE and stop if being set to FALSE.</p> <p>If the primary unit is the active unit, modifying this parameter will have no immediate effect.</p> <p>If the secondary unit is the active unit, modifying this parameter will cause an incompatibility because it changes the exchange ID.</p> <p><b>Note:</b> If this option is set to FALSE for all exchanges in a system, this setting cannot be modified in a run mode store. If at least one exchange has this setting as TRUE in the prior stop mode store, then this setting can be modified for other exchanges in a run mode store.</p>

## Modifying an Exchange's Variable Lists

When modifying the variable list for an exchange, the operation differs depending on whether EGD signatures are enabled or not. The use of EGD signatures is strongly recommended when doing run mode stores of EGD.

## Modifying Exchange Variable Lists with EGD Signatures Enabled

Modifying the variable list with signatures enabled results in either a major signature change or a minor signature change.

A major signature change in a run mode store will cause incompatibility between a producer and consumer(s). When a consumer that supports dynamic rebinding recognizes a major signature change, the consumer will request a new configuration from an EGD configuration server without user intervention.

A minor signature change in a run mode store to a producer will cause the exchange not to be produced for a short time, but will not cause the consumer(s) to stop consuming.

Type of Change	Resulting Signature Change
Adding a variable to the end of the variable list	Minor
Adding a variable at the beginning or middle of the list	Major
Deleting or modifying a variable	Major
Changing a variable's name, type, or array dimensions	Major
Changing other variable properties such as reference address and publish state	None

## Modifying Exchange Variable Lists without EGD Signatures Enabled

In applications without EGD signatures, a consumer determines compatibility solely by the number of bytes of data in the exchange. Modifying an exchange so that the length of the produced data does not match the expected length by the consumer(s) causes the consumer(s) to no longer consume that exchange. A store to update the corresponding producer/consumer is required to resume consumption of the exchange(s).

### CAUTION

With signatures disabled, it is possible for a Run Mode Store to cause incompatibility between a producer and/or consumer. For example, replacing an exchange variable with a different variable of the same size does not change the size of the exchange. Since the size of the exchange is the same, the consumer will continue to consume that exchange when the new definition has been Run Mode Stored to either the producer or the consumer.

## Modifying Exchange Variables on Targets that use EGD Commands

PACSystems targets can service EGD commands from other devices. Some commands read or write an exchange based solely on an offset into that exchange. If EGD signatures are not used, the exchange offset and length requested are validated against the length of the exchange. Without EGD signatures, the definition of the exchange can be changed entirely by a Run Mode Store and the EGD command would be serviced as long as the offset and length in the command are valid. For this reason, caution should be used when modifying EGD exchanges on a target that services EGD commands. Adding variables to the end of such exchanges would not cause a problem, but modifying or deleting variables should only be done with caution.

PACSystems targets can also be EGD command clients. EGD commands can be sent to other devices via COMMREQs in user logic. If EGD will be modified using a Run Mode Store, the exchange signature should be set to the signature value of the device that will service the command. Do not set the signature value to zero. Setting the signature value to zero disables signature checking.

## 5.7 Monitoring Ethernet Global Data Exchange Status

### 5.7.1 Exchange Status Word

A unique Exchange Status word location must be configured for each exchange. The Exchange Status word is used to store status information about an EGD exchange. The PLC writes status codes into the Exchange Status word whenever an exchange is transferred or a consumer timeout occurs.

The Exchange Status word is typically set to 1, indicating that data transfer occurred successfully. The application program can monitor for error conditions reported in the Exchange Status word by setting it to 0 once a non-zero value is written to it. In all cases, if the least significant bit of the exchange status is set to a 1, then data was transferred successfully. Status values other than 1 with the least significant bit set (e.g. 3, 5, and 7) give information about the data that was transferred, the producer, or the network that are noteworthy in the application.

### 5.7.2 Exchange Status Errors

The program should also monitor the *LAN Interface OK* Status bit (see *Section 12, Diagnostics*) for each of the Ethernet interfaces performing EGD. The Exchange Status word is invalid if the bit is 0.

Note that when an EGD exchange message received from the network contains an invalid Protocol Version Number, the Ethernet interface cannot decode the message to identify the exchange. In this case, the Exchange Status Word cannot be updated.

### 5.7.3 Exchange Status Word Error Codes

The following table shows the error codes that can be written to the Exchange Status word in the Producer Exchange and Consumer Exchange. The Exchange Status Word value for each exchange may be displayed via the STAT G Station Manager command. The Exchange Status Word value is also part of the EGD Exchange PME configuration.

Value (Dec.)	P/C	Error	Description
0	P/C	No new status event has occurred.	Produced: Initial value until the first producer period refresh occurs. Consumed: The data has not been refreshed since the previous consumption scan and the consumer timeout has not expired.
1	P	No error currently exists.	The exchange is producing data. This value should be ignored in the Output Disabled PLC modes.
1	C	No error, data consumed.	The data has been refreshed on schedule since the previous consumption.
3	C	SNTP error.	The Ethernet interface in the producer is configured for network time synchronization, but is not synchronized to an SNTP server. The data was refreshed on schedule.
4	P/C	Specification error.	During exchange configuration, an invalid configuration parameter was received by the Ethernet interface or an error occurred in communication with the PLC CPU.
5	C	Stale or invalid data sample	The producer has indicated that the data sent was stale or otherwise not valid at the time it was produced.
6	C	Refresh timeout without data.	The exchange's timeout period is configured to a non-zero value and the data has not been refreshed within the timeout period.
7	C	Data after refresh timeout.	The data has been refreshed since the previous consumption, but not within the timeout period.
10	P/C	IP Layer not currently initialized.	This status can be set during exchange configuration <sup>15</sup> if the Ethernet interface detects that it cannot currently access a network. This temporary status can change if successful network access becomes possible.
12	P/C	Lack of resource error.	Local resources are not available to establish the exchange during exchange configuration <sup>15</sup> . The PLC Fault Table may provide more detail on the specific error.
14	C	Data size mismatch error	The data size of a consumed exchange does not match the exchange definition. The exchange is ignored.
18	P/C	Loss of Ethernet interface error	This error can occur if the CPU no longer recognizes the Ethernet interface within the PLC rack. A loss of module PLC Fault Table entry will also be present. The error can also occur if the module in the given slot of the PLC rack does not match the module specified in the configuration (configuration mismatch).
30	C	Major signature mismatch	Producer and consumer signatures are different, indicating a mismatched configuration. The exchange is ignored.

**NOTE:** PACSystems does not support EGD exchange status values 16, 22, 26, and 28. These exchange status values were used in Series 90 products only.

<sup>15</sup> Exchange configuration occurs when either 1) Hardware Configuration containing EGD is stored to the PLC, 2) a PLC containing EGD configuration powers up, or 3) an Ethernet interface configured for EGD is restarted.

## Section 6 Programming EGD Commands

This chapter describes a set of commands that can be used in the application program to read and write data over the Ethernet network (also known as EGD Class2 logic-driven commands):

- Read PLC Memory
- Write PLC Memory
- Read EGD Exchange
- Write EGD Exchange
- Masked Write to EGD Exchange<sup>16</sup>

**Note:** The EGD commands described in this chapter are sent using the Communications Request (COMMREQ) function. For more information on COMMREQ, please reference *PACSystems RX7i, RX3i, and RSTi-EP CPU Programmer's Reference Manual*, GFK-2950, Section 4.8.6, *Communication Request (COMMREQ)*

---

<sup>16</sup> The RSTi-EP CPE100/CPE115 does not support this feature at the time of publication.  
*Programming EGD Commands*

## 6.1 General Use of EGD Commands

EGD Commands can be used in the application program to read and write data into PACSystems PLCs or other EGD Class 2 devices. These are one-time operations whereas EGD Exchanges are repetitive, periodic operations.

The Ethernet interface supports a maximum of 10 simultaneous EGD commands.

## 6.2 Using EGD Commands in a Redundancy System

When two Ethernet interfaces are configured for a Redundant IP only the active unit sends or responds to EGD commands using the Redundant IP address. The backup unit does not send or respond to the Redundant IP address. If the backup unit tries to send an EGD command, a COMMREQ error status is returned to its application program.

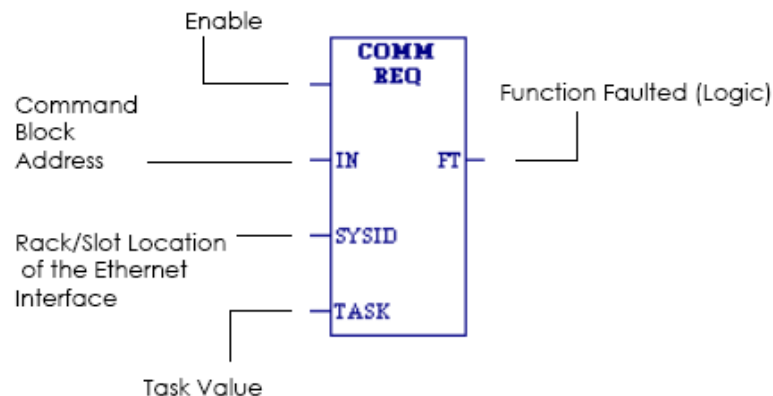
If the active Ethernet interface changes to backup status, it takes down all reliable datagram services (RDS) sessions that use the Redundant IP address. Any EGD command currently in process over the Redundant IP address when a role switch occurs is ended.

Although not recommended, EGD commands may be issued to the direct IP address. The active and backup units will respond to EGD commands received at the direct IP address. (Remote hosts should use the Redundant IP address when communicating to a redundant system.)

## 6.3 COMMREQ Format for Programming EGD Commands

The EGD commands described in this chapter are sent using the Communications Request (COMMREQ) function. The Communications Request is triggered when the logic program passes power to the COMMREQ Function Block. Figure 43 shows a sample COMMREQ format:

Figure 43: Sample COMMREQ Ladder Diagram



For the EGD commands, the parameters of the COMMREQ are:

Command	Description
<b>ENABLE</b>	Control logic for activating the COMMREQ Function Block.
<b>IN</b>	The location of the Command Block. The Command Block contains the parameters of the COMMREQ request. It can be located at any valid address within a word-oriented memory area (%R, %AI, %AQ, %P, %L, or %W) in the PACSystems PLC. Parameters for the EGD commands are described on the following pages.
<b>SYSID</b>	A hexadecimal word value that gives the rack (high byte) and slot (low byte) location of the Ethernet interface. For example, an Ethernet interface in rack zero, slot six would use the value 6 for this parameter. For the PACSystems CPU embedded Ethernet interface, enter the rack/slot location of the CPU module.
<b>TASK</b>	For the PACSystems CPU embedded Ethernet interface, Task must be set to the value 65536 (10000H) to address the CPU's Ethernet daughterboard. For a PACSystems Ethernet module, the Task must be set to zero.
<b>FT OUTPUT</b>	The FT output is set if the PLC CPU is unable to deliver the COMMREQ to the Ethernet interface. When the FT output is set, the Ethernet interface is unable to return a COMMREQ status word to the PLC logic application.



## 6.4 COMMREQ Status for the EGD Commands

Words 3 and 4 of every COMMREQ Command Block specify a memory type and location to receive status information about the execution of the command.

Word 3 specifies the memory type for the COMMREQ status word. The memory types are listed in the table below:

Type	Value (Decimal)	Value (Hex.)	Description
%R	8	08H	Register memory (word mode)
%AI	10	0AH	Analog input memory (word mode)
%AQ	12	0CH	Analog output memory (word mode)
%I	16	10H	Discrete input memory (byte mode)
	70	46H	Discrete input memory (bit mode)
%Q	18	12H	Discrete output memory (byte mode)
	72	48H	Discrete output memory (bit mode)
%T	20	14H	Discrete temporary memory (byte mode)
	74	4AH	Discrete temporary memory (bit mode)
%M	22	16H	Discrete momentary internal memory (byte mode)
	76	4CH	Discrete momentary internal memory (bit mode)
%G	56	38H	Discrete global data table (byte mode)
	86	56H	Discrete global data table (bit mode)
%W	196	C4H	Word memory (word mode; limited to %W1-%W65536)

Word 4 of the COMMREQ Command Block specifies the offset within the memory type selected. **The status word address offset is a zero-based number.** For example, if %R1 should be the location of the status word, you must specify a zero for the offset. The offset for %R100 would be 99 decimal. (When using %W memory, the maximum offset value that can be entered is 65535, signifying %W65536.)

### 6.4.1 COMMREQ Status Values

The Ethernet interface reports the status of the COMMREQ back to the status location. See Section 12, *Diagnostics for COMMREQ* status values that may be reported for the EGD commands.

## 6.5 Read PLC Memory (4000)

The Read PLC Memory command can be used to read memory locations from a remote PACSystems PLC. This command does not require the configuration of a produced/consumed exchange in the PLCs. The Read PLC Memory command can only be sent to an individual IP Address; it cannot be sent to a Group ID (multicast).

### 6.5.1 Read PLC Memory Command Block

Word Offset	Value	Description
Word 1	Length of command data block	Always 16
Word 2	0	Always 0 (no-wait mode request)
Word 3	(See the previous page)	The memory type of COMMREQ Status Word
Word 4	0-based	The offset of COMMREQ Status Word
Word 5	0	Reserved
Word 6	0	Reserved
Word 7	4000 (fa0H))	Read PLC Memory command number.
Word 8	Retry time, in milliseconds	The time between retries of command transfers. Default is 1000ms.
Word 9	Local read buffer memory type	Memory type for the data to be placed in the local PLC.
Word 10	Local read buffer reference table starting address (least significant word)	1-based offset in the local PLC
Word 11	Local read buffer reference table starting address (most significant word)	

(Continued ...)

#### Further Consideration of Memory Locations 1-11

**(Word 7) EGD Command Number:** Word 7 requests that a read PLC memory operation occur. If the command is processed successfully, it will result in PLC reference memory data being retrieved from the server to the client.

**(Word 8) Command Retry Time:** Word 8 specifies the time (in milliseconds) the Ethernet interface will wait between retries when transferring the command. A total of four tries will be made to send the command. If no response is received after the four tries (i.e. after four times the retry time value), an error status will be returned in the COMMREQ status word. If the command retry is specified as zero, the default value of one second is used.

**(Word 9) Local PLC - Memory Type:** Words 9-11 specify the location in the local PLC where the Ethernet interface will store data received from the remote PLC. The amount of data to be transferred is specified by the number of memory units of the data read from the remote PLC (Word 15). For more information, refer to *Word 9 Memory Types Available* in this section.

**(Words 10 - 11) Local PLC - Memory Starting Address:** Words 10 and 11 determine the starting address in the local PLC in which the data from the remote PLC is to be stored. The value entered is the 32-bit offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 9. Word 10 contains the least significant 16 bits of the offset; Word 11 contains the most significant 16 bits of the offset. This offset will be either in bits, bytes, or words depending on the mode specified. Valid ranges of values depend on the PLC's memory ranges. The user is responsible for assuring that this area is large enough to contain the requested data without overwriting other application data.

Word Offset	Value	Description
Word 12	Remote read location memory type	Memory type from which data will be read in the remote PLC
Word 13	Remote reference table read location starting address (least significant word)	1-based offset in the remote PLC
Word 14	Remote reference table read location starting address (most significant word)	
Word 15	Remote reference table length (in remote memory units)	The number of remote memory units to be read.
Word 16	Network address type	Must be 1. Indicates an IP address will be used.
Word 17	Network address length	Must be 4 for IP address. Group ID (multicast) is not permitted.
Word 18 – Word 21	IP Address of the remote PLC	Four integers, specified as one integer per word of the dotted-decimal IP address of the remote PLC. It may not be a group IP address.
Word 22	Reserved	Always 0

### *Further Consideration of Memory Locations 12-22*

**(Word 12) Remote PLC - Memory Type:** Words 12–14 specify the memory type and starting address in the remote PLC from which the data is to be read. Valid values for Word 12 are listed above.

**(Words 13-14) Remote PLC - Memory Starting Address:**

Words 13 & 14 determine the starting address in the remote PLC from which the data is to be read. The value entered is the 32-bit offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 12. Word 13 contains the least significant 16 bits of the offset; word 14 contains the most significant 16 bits of the offset. This offset will be either in bits, bytes, or words depending on the mode specified. Valid values depend on the remote PLC's memory ranges.

**(Word 15) Remote PLC - Number of Memory Units:** Word 15 specifies the amount of data to be transferred. The value entered is the number of memory units to be transferred, where the size of the remote PLC memory type (bit, byte, or word) is specified in Word 12. For Read PLC Memory, the maximum length is 11200 bits, 1400 bytes, or 700 words of data, or the amount of memory available in the PLC for the selected memory type, whichever is less.

**(Word 16) Remote PLC - Network Address Type:**

Word 16 specifies the format of the remote PLC address. Word 16 must contain the value 1. This indicates a dotted-decimal IP address expressed using a separate register for each decimal digit.

**(Word 17) Remote PLC -Network Address Length:** Word 17 specifies the length in words of the remote PLC IP address in this COMMREQ Command Block. Word 17 must contain 4.

**(Words 18-21) Remote PLC - IP Address:** Words 18 – 21 specify the four integers, one integer per word, of the dotted-decimal IP address of the remote PLC to be accessed.

## Word 9 Memory Types Available

Valid values for Word 9 are listed below. The amount of data to be transferred is specified by the number of memory units of the data read from the remote PLC (Word 15).

Type	Value (Decimal)	Description
%W <sup>17</sup>	196	Word memory (word mode)
%R	8	Register memory (word mode)
%AI	10	Analog input memory (word mode)
%AQ	12	Analog output memory (word mode)
%I	16	Discrete input memory (byte mode)
	70	Discrete input memory (bit mode)
%Q	18	Discrete output memory (byte mode)
	72	Discrete output memory (bit mode)
%T	20	Discrete temporary memory (byte mode)
	74	Discrete temporary memory (bit mode)
%M	22	Discrete momentary internal memory (byte mode)
	76	Discrete momentary internal memory (bit mode)
%SA	24	Discrete system memory group A (byte mode)
	78	Discrete system memory group A (bit mode)
%SB	26	Discrete system memory group B (byte mode)
	80	Discrete system memory group B (bit mode)
%SC	28	Discrete system memory group C (byte mode)
	82	Discrete system memory group C (bit mode)
%S <sup>18</sup>	30	Discrete system memory (byte mode)
	84	Discrete system memory (bit mode)
%G	56	Discrete global data table (byte mode)
	86	Discrete global data table (bit mode)

<sup>17</sup> %W memory is supported on PACSystems clients and servers only.

<sup>18</sup> Read-only memory, cannot be written to.

## 6.6 Write PLC Memory (4001)

The Write PLC Memory command can be used to write memory locations to one remote PACSystems PLC. Use of this command does not require a configured produced/consumed exchange in the PLCs.

### 6.6.1 Write PLC Memory Command Block

Word Offset	Value	Description
Word 1	Length of command data block	Always 16
Word 2	0	Always 0 (no-wait mode request)
Word 3	(See table on page 116)	The memory type of COMMREQ Status Word
Word 4	0-based	The offset of COMMREQ Status Word
Word 5	0	Reserved
Word 6	0	Reserved
Word 7	4001 (fa1H)	Write PLC Memory command number.
Word 8	Retry time, in milliseconds	The time between retries of command transfers. Default is 1000ms.
Word 9	Local write buffer memory type	Memory type for the data that will be written, in the local PLC.
Word 10	Local write buffer reference table starting address (least significant word)	1-based offset in the local PLC.
Word 11	Local write buffer reference table starting address (most significant word)	

(Continued ...)

#### Further Consideration of Memory Locations 1-11

**(Word 7) EGD Command Number:** Word 7 a write PLC memory operation. If the command is processed successfully, it will result in PLC reference memory data being sent from the server to the client.

**(Word 8) Command Retry Time:** Word 8 specifies the time (in milliseconds) the Ethernet interface will wait between retries when transferring the command. A total of four tries will be made to send the command. If no response is received after the four tries (i.e. after four times the retry time value), an error status will be returned in the COMMREQ status word. If the command retry is specified as zero, the default value of one second is used.

**(Word 9) Local PLC - Memory Type:** Words 9-11 specify the location in the local PLC where the Ethernet interface will get the data to be written to the remote PLC. Valid values for Word 9 are listed in the description of Read PLC Memory Command. The amount of data to be transferred is specified by the number of memory units of the data written to the remote PLC (Word 15). For more information, refer to *Word 9 Memory Types Available* in this section.

**(Words 10 - 11) Local PLC - Memory Starting Address:** Words 10 and 11 determine the starting address in the local PLC from which the data is to be written to the remote PLC. The value entered is the 32-bit offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 9. Word 10 contains the least significant 16 bits of the offset; Word 11 contains the most significant 16 bits of the offset. This offset will be either in bits, bytes, or words depending on the mode specified. (For example, if Word 9=16 and Words 10,11 = 2, 0 then the starting address will be %I9.) Valid ranges of values depend on the PLC's memory ranges.

Word Offset	Value	Description
Word 12	Remote write location memory type	Memory type into which data will be written in the remote PLC(s)
Word 13	Remote reference table write location starting address (least significant word)	1-based offset in the remote PLC
Word 14	Remote reference table write location starting address (least significant word)	
Word 15	Write Length	0 to 1400 bytes, 0 to 700 words.
Word 16	Network address type	Must be 1. Indicates an IP address will be used.
Word 17	Network address length	Must be 4 for IP address. Group ID (multicast) is not permitted.
Word 18 – Word 21	IP Address of the remote PLC	Four integers, specified as one integer per word of the dotted-decimal IP address of the remote PLC. May not be a group IP address.
Word 22	Reserved	Always 0

### *Further Consideration of Memory Locations 12-22*

**(Word 12) Remote PLC - Memory Type:** Words 12–14 specify the memory type and starting address in the remote PLC where data is to be written. Valid values for Word 12 are listed above.

**(Words 13 - 14) Remote PLC - Memory Starting Address:** Words 13, 14 determine the starting address in the remote PLC where data is to be written. The value entered is the 32-bit offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 12. Word 13 contains the least significant 16 bits of the offset; word 14 contains the most significant 16 bits of the offset. This offset will be either in bits, bytes, or words depending on the mode specified (for example, if Word 12=16 and Words 13,14 =9, 0, then the starting address will be %I65). Valid ranges of values depend on the remote PLC's memory ranges.

**(Word 15) Remote PLC - Number of Memory Units:** Word 15 specifies the amount of data to be transferred. The value entered is the number of memory units to be transferred, where the size of the remote PLC memory type (bit, byte, or word) is specified in Word 12.

**(Word 16) Remote PLC - Network Address Type:** Word 16 specifies the format of the remote PLC address. Word 16 must contain the value 1. This indicates a dotted-decimal IP address expressed using a separate register for each decimal digit.

**(Word 17) Remote PLC - Network Address Length:** Word 17 specifies the length in words of the remote PLC IP address in this COMMREQ Command Block. Word 17 must contain 4.

**(Words 18 – 21) Remote PLC - IP Address:** Words 18–21 specify the four integers, one integer per word, of the dotted-decimal IP address of the remote PLC to be accessed.

## 6.7 Read EGD Exchange (4002)

The Read EGD Exchange command can be used to read full or partial data from a producer or the consumer exchange. This command identifies the data to be read using its configured Producer ID and Exchange ID. It can then read the content of the data for the exchange, directly from the producer or consumer device memory. This command can be sent to PACSystems PLCs and other EGD Class 2 devices. In a PACSystems PLC, reading an EGD exchange reads the PLC reference memory locations configured to be transferred at the specified offset in the exchange. Thus, current process data will be read, not the data that was transferred last in the exchange.

### 6.7.1 Read EGD Exchange Command Block

Word Offset	Value	Description
Word 1	Length of command data block	Always 19
Word 2	0	Always 0 (no-wait mode request)
Word 3	(See table on page 116)	The memory type of COMMREQ Status Word
Word 4	0-based	The offset of COMMREQ Status Word
Word 5	0	Reserved
Word 6	0	Reserved
Word 7	4002 (fa2H)	Read EGD Exchange command number.
Word 8	Retry time, in milliseconds	The time between retries of command transfers. Default is 1000ms.
Word 9	Local read buffer memory type	Memory type for the data, in the local PLC.
Word 10	Local read buffer reference table starting address (least significant word)	1-based offset

(Continued ...)

#### *Further Consideration of Memory Locations 1-10*

**(Word 7) EGD Command Number:** Word 7 requests that a read EGD exchange operation occur. If the command is processed successfully, it will result in data from a specified EGD exchange being read from the client to the server.

**(Word 8) Command Retry Time:** Word 8 specifies the time (in milliseconds) the Ethernet interface will wait between retries when transferring the command. A total of four tries will be made to send the command. If no response is received after the four tries (i.e. after four times the retry time value), an error status will be returned in the COMMREQ status word. If the command retry is specified as zero, the default value of one second is used.

**(Word 9) Local PLC – Memory Type:** Words 9-11 specify the location in the local PLC where the Ethernet interface will get the data to be read from the remote EGD device. Valid values for Word 9 are listed in the description of Read PLC Memory Command. The amount of data to be transferred is specified by the Exchange DataLength (Word 18).

**(Words 10 – 11) Local PLC – Memory Starting Address:** Words 10 and 11 determine the starting address in the local PLC where data is to be read from the remote EGD exchange. The value entered is the 32-bit offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 9. Word 10 contains the least significant 16 bits of the offset; Word 11 contains the most significant 16 bits of the offset. This offset will be either in bits, bytes, or words depending on the mode specified. (For example, if Word 9=16 and Words 10,11 = 2, 0 then the starting address will be %I9.) Valid ranges of values depend on the PLC's memory ranges. The user is responsible for assuring that this area is large enough to contain the requested data without overwriting other application data.

Word Offset	Value	Description
Word 11	Local read buffer reference table starting address (most significant word)	
Word 12	Remote signature	EGD Exchange signature. This should be 0 for PLCs when not using signatures. If the run-mode store of EGD will be used, the use of signatures is highly recommended.
Word 13	Remote Producer ID (least significant word)	EGD Producer ID
Word 14	Remote Producer ID (most significant word)	
Word 15	Remote Exchange ID (least significant word)	EGD Exchange ID
Word 16	Remote Exchange ID (most significant word)	
Word 17	Remote Exchange Offset	Byte offset (0-based) in the exchange that should be read.
Word 18	Read length	The number of bytes to be read in the range 0 to 1400 bytes.
Word 19	Network address type	Must be 1. Indicates that an IP address will be used.
Word 20	Network address length	Must be 4 for IP address. Group ID (multicast) is not permitted.
Word 21 to Word 24	IP Address of the remote PLC	Four integers, specified as one integer per word of the dotted-decimal IP address of the remote PLC. May not be a group IP address.
Word 25	Reserved	Always 0

### Further Consideration of Memory Locations 11-25

**(Words 10 – 11) Local PLC – Memory Starting Address:** Words 10 and 11 determine the starting address in the local PLC where data is to be read from the remote EGD exchange. The value entered is the 32-bit offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 9. Word 10 contains the least significant 16 bits of the offset; Word 11 contains the most significant 16 bits of the offset. This offset will be either in bits, bytes, or words depending on the mode specified. (For example, if Word 9=16 and Words 10,11 = 2, 0 then the starting address will be %I9.) Valid ranges of values depend on the PLC's memory ranges. The user is responsible for

**(Words 13 – 14) Remote EGD exchange – Producer ID:** Words 13 and 14 contains the 32-bit Producer ID of the desired exchange at the remote EGD device. Word 13 contains the least significant 16 bits of the Producer ID; word 14 contains the most significant 16 bits.

**(Words 15 – 16) Remote EGD exchange – Exchange ID:** Words 15 and 16 contains the 32-bit Exchange ID of the desired exchange at the remote EGD device. Word 15 contains the least significant 16 bits of the Exchange ID; word 16 contains the most significant 16 bits.

**(Word 17) Remote EGD exchange – Exchange Data Offset:** Word 17 contains the 0-based byte offset of the data to be read from the data portion of the exchange at the remote EGD device.

**(Word 18) Remote EGD exchange – Exchange Data Length:** Word 18 contains the length (in bytes) of the exchange data to be read from the remote EGD device. The exchange data length may not exceed 1400 bytes or the amount of memory available in the PLC for the selected memory type, whichever is less.

**(Word 19) Remote Server – Network Address Type:** Word 19 specifies the format of the remote PLC address. Word 19 must contain the value 1. This indicates a dotted-decimal IP address expressed using a separate register for each decimal digit.

**(Word 20) Remote Server – Network Address Length:** Word 20 specifies the length in words of the remote PLC IP address in this COMMREQ Command Block. Word 20 must contain 4.

**(Words 21 – 24) Remote Server – IP Address:** Words 21–24 specify the four integers, one integer per word, of the dotted-decimal IP address of the remote PLC to be accessed.



## Word 12 Remote Signatures

**(Word 12) Remote EGD exchange – Exchange Signature:** Words 12 contains the 16-bit exchange signature value to be compared at the remote EGD device. For remote PLCs, the exchange signature should be set to zero if signatures are not being used. However, when signatures are enabled, the signature field can be set to a non-zero value so that commands will only be executed if signatures match. In this case, mismatched signatures will cause the command to return a failure status.

An EGD signature has the format *maj.min*, where *maj* is the major value, and *min* is the minor value. The least significant byte of this word indicates the minor value and the most significant byte indicates the major value. For example, a value of 0xAABB refers to a *maj.min* value of 0xAA.0xBB.

<i>EGD Signatures Enabled (Y/N)</i>	<i>Signature Comparison Desired</i>	<i>Recommended with RMS of EGD</i>	<i>User Specified Signature</i>
No	No	No	0 (Default - no check)
Yes	No	No	0 (Default - no check)
Yes	Yes	Yes	Current EGD signature

## 6.8 Write EGD Exchange (4003)

The Write EGD Exchange command can be used to write portions of a configured Ethernet Global Data exchange in a remote producer node. EGD protocol prohibits writing to a consumed exchange. This command identifies the exchange to be written using its configured Producer ID and Exchange ID. It can then write the content of that data directly to the device memory. This command can be sent to PACSystems PLCs and other EGD Class 2 devices. In a PACSystems PLC, writing an EGD exchange modifies the PLC reference memory locations configured for transfer at the specified offset in the exchange. Thus, current process data will be updated, not the data that was transferred last in the exchange.

### 6.8.1 Write EGD Exchange Command Block

Word Offset	Value	Description
Word 1	Length of command data block	Always 19
Word 2	0	Always 0 (no-wait mode request)
Word 3	(See table on page 116)	The memory type of COMMREQ Status Word
Word 4	0-based	The offset of COMMREQ Status Word
Word 5	0	Reserved
Word 6	0	Reserved
Word 7	4003 (fa3H)	Write the EGD Exchange command number.
Word 8	Retry time, in milliseconds	The time between retries of command transfers. Default is 1000ms.
Word 9	Local write buffer memory type	Memory type for the data, in the local PLC.
Word 10	Local write buffer reference table starting address (least significant word)	1-based offset
Word 11	Local write buffer reference table starting address (most significant word)	

#### *Further Consideration of Memory Locations 1-11*

**(Word 7) EGD Command Number:** Word 7 requests that a write EGD exchange operation occur. If the command is processed successfully, it will result in data for a specified EGD exchange being written from the client to the server.

**(Word 8) Command Retry Time:** Word 8 specifies the time (in milliseconds) the Ethernet interface will wait between retries when transferring the command. A total of four tries will be made to send the command. If no response is received after the four tries (i.e. after four times the retry time value), an error status will be returned in the COMMREQ status word. If the command retry is specified as zero, the default value of one second is used.

**(Word 9) Local PLC - Memory Type:** Words 9-11 specify the location in the local PLC where the Ethernet interface will get the data to write to the remote EGD device. Valid values for Word 9 are listed in the description of Read PLC Memory Command. The amount of data to be transferred is specified by the Exchange Data Length (Word 18).

**(Words 10 - 11) Local PLC - Memory Starting Address:** Words 10 and 11 determine the starting address in the local PLC from which data is to be written to the remote EGD exchange. The value entered is the 32-bit offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 9. Word 10 contains the least significant 16 bits of the offset; Word 11 contains the most significant 16 bits of the offset. This offset will be either in bits, bytes, or words depending on the mode specified. (For example, if Word 9=16 and Words 10,11 = 2, 0 then the starting address will be %I9.) Valid ranges of values depend on the PLC's memory ranges.

Word Offset	Value	Description
Word 12	Remote signature	EGD Exchange signature. This should be 0 for PLCs when not using signatures. If the run-mode store of EGD will be used, the use of signatures is highly recommended.
Word 13	Remote Producer ID (least significant word)	EGD Producer ID
Word 14	Remote Producer ID (most significant word)	
Word 15	Remote Exchange ID (least significant word)	EGD Exchange ID
Word 16	Remote Exchange ID (most significant word)	
Word 17	Remote Exchange Offset	Byte offset (0-based) in the exchange that should be read.
Word 18	Write length	The number of bytes to be written in the range 0 to 1400 bytes.
Word 19	Network address type	Must be 1. Indicates an IP address will be used.
Word 20	Network address length	Must be 4 for IP address. Group ID (multicast) is not permitted.
Word 21 to Word 24	IP Address of the remote PLC	Four integers, specified as one integer per word of the dotted-decimal IP address of the remote PLC. May not be a group IP address.
Word 25	Reserved	Always 0

### Further Consideration of Memory Locations 12-25

**(Word 12) Remote EGD exchange – Exchange Signature:** Words 12 contains the 16-bit exchange signature value to be compared at the remote EGD device. For remote PLCs, the exchange signature should be set to zero if signatures are not being used. However, when signatures are enabled, the signature field can be set to a non-zero value so that commands will only be executed if signatures match. In this case, mismatched signatures will cause the command to return a failure status.

**(Words 13 - 14) Remote EGD exchange – Producer ID:** Words 13 and 14 contains the 32-bit Producer ID of the desired exchange at the remote EGD device. Word 13 contains the least significant 16 bits of the Producer ID; word 14 contains the most significant 16 bits.

**(Words 15 - 16) Remote EGD exchange – Exchange ID:** Words 15 and 16 contains the 32-bit Exchange ID of the desired exchange at the remote EGD device. Word 15 contains the least significant 16 bits of the Exchange ID; word 16 contains the most significant 16 bits. For the Write EGD Command, the exchange at the remote device must be a Produced exchange.

**(Word 17) Remote EGD exchange – Exchange Data Offset:** Word 17 contains the 0-based byte offset of the data to be overwritten in the data portion of the exchange at the remote EGD device.

**(Word 18) Remote EGD exchange – Exchange Data Length:** Word 18 contains the length (in bytes) of the exchange data to be written to the remote EGD device. The exchange data length may not exceed 1400 bytes or the amount of memory available in the PLC for the selected memory type, whichever is less.

**(Word 19) Remote Server - Network Address Type:** Word 19 specifies the format of the remote PLC address. Word 19 must contain the value 1. This indicates a dotted-decimal IP address expressed using a separate register for each decimal digit.

**(Word 20) Remote Server - Network Address Length:** Word 20 specifies the length in words of the remote PLC IP address in this COMMREQ Command Block. Word 20 must contain 4.

**(Words 21 – 24) Remote Server - IP Address:** Words 21–24 specify the four integers, one integer per word, of the dotted-decimal IP address of the remote PLC to be accessed.

## 6.9 Masked Write to EGD Exchange (4004)

The Masked Write to EGD Exchange command can be used to write one or more bits in a single byte of a configured Ethernet Global Data exchange in a remote producer node. EGD protocol prohibits writing to a consumed exchange. This command can be sent to PACSystems PLCs and other EGD Class 2 devices.

In a PACSystems PLC, writing an EGD exchange modifies the PLC reference memory locations configured to be transferred at the specified offset in the exchange. Thus, current process data will be updated, not the data that was transferred last in the exchange.

### 6.9.1 Masked Write EGD Exchange Command Block

Word Offset	Value	Description
Word 1	Length of command data block	Always 17
Word 2	0	Always 0 (no-wait mode request)
Word 3	(See table on page 116)	The memory type of COMMREQ Status Word
Word 4	0-based	The offset of COMMREQ Status Word
Word 5	0	Reserved
Word 6	0	Reserved
Word 7	4004 (fa4H)	Masked Write to EGD Exchange command number.
Word 8	Retry time, in milliseconds	The time between retries of command transfers. Default is 1000ms.
Word 9	Bitmask, set a bit to be written to 1, rest to 0	The bitmask selects the individual bit to be written. The most significant bytes of Word 9 and Word 10 are ignored.
Word 10	Write 0 or 1 to the selected bit.	Value to set the bit selected by the bitmask in Word 9.

(Continued ...)

#### *Further Consideration of Memory Locations 1-10*

**(Word 7) EGD Command Number:** Word 7 requests that a masked write EGD exchange operation occur. If the command is processed successfully, it will result in a data bit for a specified EGD exchange being written from the client to the server.

**(Word 8) Command Retry Time:** Word 8 specifies the time (in milliseconds) the Ethernet interface will wait between retries when transferring the command. A total of four tries will be made to send the command. If no response is received after the four tries (i.e. after four times the retry time value), an error status will be returned in the COMMREQ status word. If the command retry is specified as zero, the default value of one second is used.

**(Word 9) Bit Mask:** Words 9 – 10 specify the individual data to be written to the remote EGD exchange. The usage of the Bit Mask and Data are described in *Masked Write to EGD Exchange Bit Mask and Data Bits*, below. Word 9 contains a bit mask that identifies a bit or bits within a data byte. The mask bit corresponding to each data bit to be written is set to 1; all other bits are set to 0.

**(Word 10) Data:** Word 10 contains the data byte that contains the bit or bits to be written to the remote EGD exchange. The individual data bits to be written are in the same position as the "1" bits in the Bit Mask (Word 9).

Word Offset	Value	Description
Word 11	Remote signature	EGD Exchange signature. This should be 0 for PLCs when not using signatures. If the run-mode store of EGD will be used, the use of signatures is highly recommended.
Word 12	Remote Producer ID (least significant word)	EGD Producer ID
Word 13	Remote Producer ID (most significant word)	
Word 14	Remote Exchange ID (least significant word)	EGD Exchange ID
Word 15	Remote Exchange ID (most significant word)	
Word 16	Remote Exchange Offset	Byte offset (0-based) in the exchange that should be read.
Word 17	Network address type	Must be 1. Indicates an IP address will be used.
Word 18	Network address length	Must be 4 for IP address. Group ID (multicast) is not permitted.
Word 19 to Word 22	IP Address of the remote PLC	Four integers, specified as one integer per word of the dotted-decimal IP address of the remote PLC. May not be a group IP address.
Word 23	Reserved	Always 0.

### *Further Consideration of Memory Locations 11-23*

**(Word 11) Remote EGD exchange – Exchange Signature:** Words 11 contains the 16-bit exchange signature value to be compared at the remote EGD device. For remote PLC's, the exchange signature should be set to zero if signatures are not being used. However, when signatures are enabled, the signature field can be set to a non-zero value so that commands will only be executed if signatures match. In this case, mismatched signatures will cause the command to return a failure status.

**(Words 12 - 13) Remote EGD exchange – Producer ID:** Words 12 and 13 contains the 32-bit Producer ID of the desired exchange at the remote EGD device. Word 12 contains the least significant 16 bits of the Producer ID; word 13 contains the most significant 16 bits.

**(Words 14 - 15) Remote EGD exchange – Exchange ID:** Words 14 and 15 contains the 32-bit Exchange ID of the desired exchange at the remote EGD device. Word 14 contains the least significant 16 bits of the Exchange ID; word 15 contains the most significant 16 bits. For the Masked Write EGD Command, the exchange at the remote device must be a Produced exchange.

**(Word 16) Remote EGD exchange – Exchange Data Offset:** Word 16 contains the 0-based byte offset of the single data byte data containing the bit or bits to be overwritten in the data portion of the exchange at the remote EGD device.

**(Word 17) Remote Server - Network Address Type:** Word 17 specifies the format of the remote PLC address. Word 17 must contain the value 1. This indicates a dotted-decimal IP address expressed using a separate register for each decimal digit.

**(Word 18) Remote Server - Network Address Length:** Word 18 specifies the length in words of the remote PLC IP address in this COMMREQ Command Block. Word 18 must contain 4.

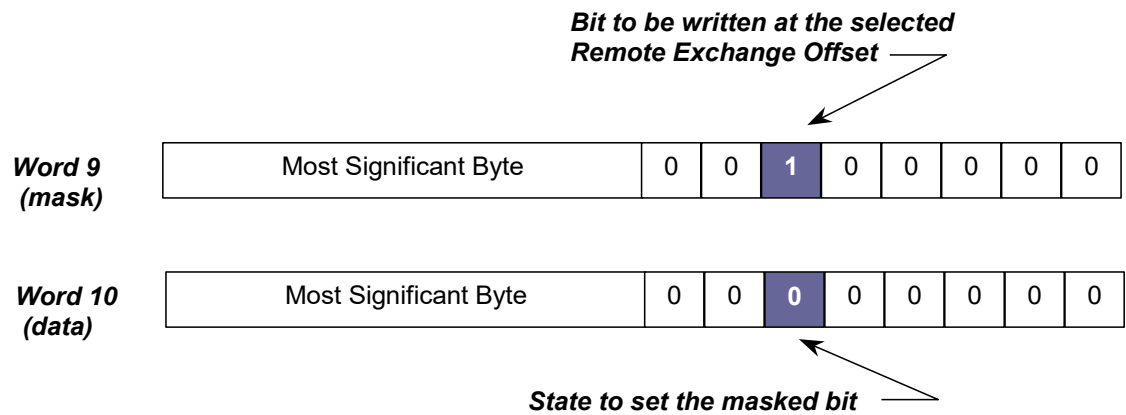
**(Words 19 – 22) Remote Server - IP Address:** Words 19–22 specify the four integers, one integer per word, of the dotted-decimal IP address of the remote PLC to be accessed.

### Masked Write to EGD Exchange Bitmask and Data Bits

Word 9 of the Masked Write command contains the bitmask. The most significant byte of Word 9 is ignored. In the least significant byte, any bits set to 1 will be written to the remote producer.

The equivalent bit of Word 10 of the Masked Write command contains the bit state to be written, 1 or 0. The most significant byte of Word 10 is also ignored.

**Figure 44: Example: Masked Write to EGD Exchange Bitmask and Data Bits**





## Section 7 SNTP Operation

This chapter describes how to sync PACSystems Ethernet interfaces to an SNTP server to synchronize computer clocks on a network. The following topics are covered:

- Normal SNTP Operation
- Using Multiple SNTP Servers
- Loss or Absence of SNTP Servers
- Time-Stamping of Ethernet Global Data Exchanges



## 7.1 Normal SNTP Operation

If SNTP is configured, the default mode of operation is Broadcast and Multicast. For Unicast mode of communication, the user can configure the necessary parameters using the *Hardware Configuration* node in PME (preferred method). Alternatively, if the hardware is compatible with AUP files, the user can configure the necessary parameters defined in *Configuring Advanced User Parameters*.

### 7.1.1 SNTP Broadcast and Multicast Operation Mode

The Ethernet interface will synchronize to a remote SNTP time server after receiving two broadcast clock values within 150 seconds. The Station Manager can be used to view server status information.

### 7.1.2 SNTP Unicast Operation Mode

In this mode, the module tries to request the time from a time server to synchronize the clock. You can configure a maximum of two time-servers: One for Primary Time Server and another for Secondary Time Server. Based on the configuration parameters, the Ethernet module first polls the Primary Time Server and synchronizes the clock. If the Primary Server does not respond to the requests, it switches to the Secondary Server and polls it for updated time. This process repeats until it synchronizes to one of the time servers. Polling rate and timing for switching from one server to another server are defined as user-configurable parameters. These parameters are defined in 7.4.2 *Steps to Synchronize the CPU TOD Clock to an SNTP Server (using PME)*. For parameter definitions in older systems, refer to *Configuring Advanced User Parameters*.

## 7.2 Multiple SNTP Servers (Applies only to SNTP Broadcast and Multicast Mode)

To guard against loss of SNTP timing messages, multiple SNTP time servers can be tracked on a network. An Ethernet interface can maintain timing information from up to four total SNTP timeservers at a time. Each server assigns a stratum number that determines its priority. The Ethernet interface uses the message from the server with the lowest stratum number until communication with that server is lost. Then the server with the next lowest stratum number becomes the server of choice and the Ethernet interface synchronizes to it if it receives two of its timing messages within 150 seconds. A server is considered "lost" if more than 150 seconds elapse between timing messages.

## 7.3 Loss or Absence of SNTP Timing Signals

If an Ethernet interface is configured for SNTP, but does not receive two subsequent timing messages from an SNTP network timeserver within 150 seconds, the following will happen:

- A fault entry will be placed in the PLC Fault Table.
- A fault entry will be placed in the Ethernet interface's exception log. This log can be read using the Station Manager.
- The Status word within a consumed exchange will indicate new data with a value of 3, instead of the normal 1 value, indicating that SNTP is selected, but the Ethernet interface is not synchronized to an SNTP server. This Status word value can be obtained from the PLC register configured for the specific exchange

**Note:** The SNTP error condition is considered the least important of all possible error codes. Therefore, if another error condition exists, its status code will appear in the Status word instead of the SNTP error code.

Upon loss or absence of synchronization, the Ethernet interface's built-in clock will operate as follows:

- If after its last power-up/restart cycle, the Ethernet interface has not received an SNTP server's timing message, it will continue to use the PLC CPU's local clock value that is received at the power-up/restart for its time base.
- If the Ethernet interface has been synchronized to an SNTP server but has lost its signal, it will use the most recently received SNTP time message as its time base.

The Ethernet interface will continue supplying time values to the PLC CPU for timestamping, while it searches for SNTP timing messages from the network. If SNTP messages are received later, the Ethernet interface will then synchronize to them.

## 7.4 Time-Stamping of Ethernet Global Data Exchanges

The CPU adds a timestamp to each Ethernet Global Data Message it produces. The timestamp indicates when the data was transferred from the producing PLC's CPU to its Ethernet interface for transmission over the network.

The timestamp is an 8-byte value representing the time elapsed since midnight, January 1, 1970. The first four bytes contain a signed integer representing seconds and the next four bytes contain a signed integer representing nanoseconds. This value can be examined to determine whether a packet received from the network has a new data sample or if it is the same data received previously.

There are three ways a PLC CPU can acquire a timestamp from a time clock source depending on the PLC configuration:

1. The PLC CPU can obtain a timestamp by synchronizing with the Ethernet interface's time clock (which is synchronized with either the clock in the CPU or an external SNTP server on the network).
2. The PLC CPU can obtain a timestamp by synchronizing with a CPU with an embedded ethernet interface. The embedded Ethernet's internal time clock can be synchronized with an external SNTP server on the network.
3. The timestamp data can be obtained from the CPU's TOD clock when the TOD clock is synchronized with an SNTP server.

Given the number of different ways that a PLC can be configured, the chart below demonstrates how the EGD Timestamps will be affected by each of the AUPs available.

	SNTP	CPU TOD Clock Sync	UTC Offset	DST	Time
<b>CPE Produced EGD Exchanges</b>	Disabled	-	Enabled	Enabled	CPU Clock Affected
	Enabled	-	Enabled	Enabled	EGD Timestamps in GMT Time
	Enabled	Enabled	Enabled	Enabled	EGD Timestamps in UTC/DST Adjusted Time
	Enabled	Enabled	Disabled	Disabled	EGD Timestamps in GMT Time
	Enabled	Disabled	Enabled	Enabled	EGD Timestamps in GMT Time
<b>ETM Produced EGD Exchanges</b>	Disabled	-	Enabled	Enabled	CPU Clock Unaffected
	Enabled	-	Enabled	Enabled	EGD Timestamps in UTC/DST Adjusted Time
	Enabled	Enabled	Enabled	Enabled	EGD Timestamps in UTC/DST Adjusted Time
	Enabled	Enabled	Disabled	Disabled	EGD Timestamps in GMT Time <sup>19</sup>
	Enabled	Disabled	Disabled	Disabled	EGD Timestamps in UTC

<sup>19</sup> CPU Local in GMT Time  
SNTP Operation

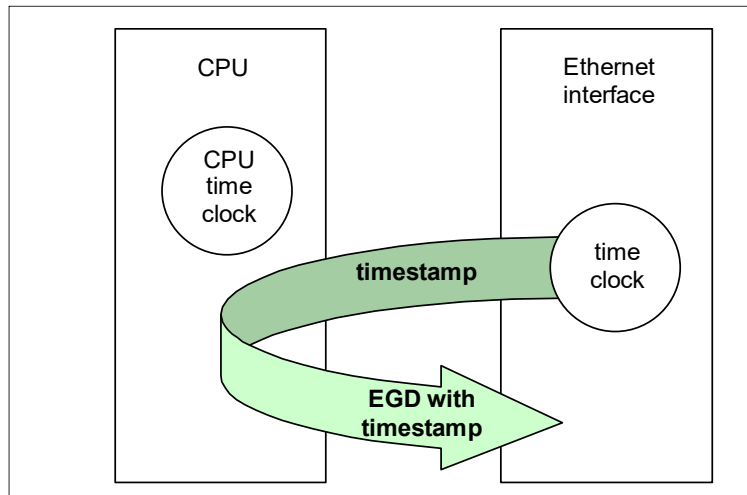
## 7.4.1 Obtaining Timestamps from the Ethernet Interface Clock

The PLC CPU obtains the timestamp data from the time clock in the Ethernet interface. The CPU only uses this timestamp for Ethernet Global Data exchanges. The timestamp from the Ethernet interface does not affect the time of the CPU's internal time clock.

### Ethernet Interface Clock

If time synchronization between the CPU and ETM is lost (for example, during power cycle or stop-to-run transition) the CPU uses its own clock for the time stamp.

**Figure 45: Obtaining Timestamps from the Ethernet interface Clock**

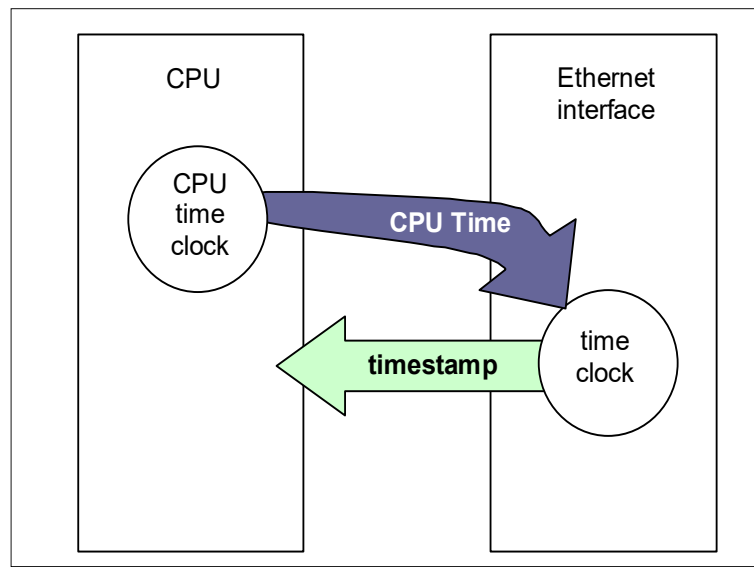


The time clock in the Ethernet interface is synchronized to either the clock in the CPU or an external SNTP server on the network. Selection of the timestamp source for Ethernet Global Data is part of the basic configuration of the Ethernet interface, as explained in Chapter 4.

## PLC's Time Clock

If this source is selected, the Ethernet interface's built-in time clock is synchronized at power-up or restart to the clock in the PLC CPU. The timestamp information produced by the PLC has a resolution of 100 microseconds. Because the time clocks in the PLCs on the network are not synchronized, EGD timestamps produced by different PLCs cannot be compared accurately.

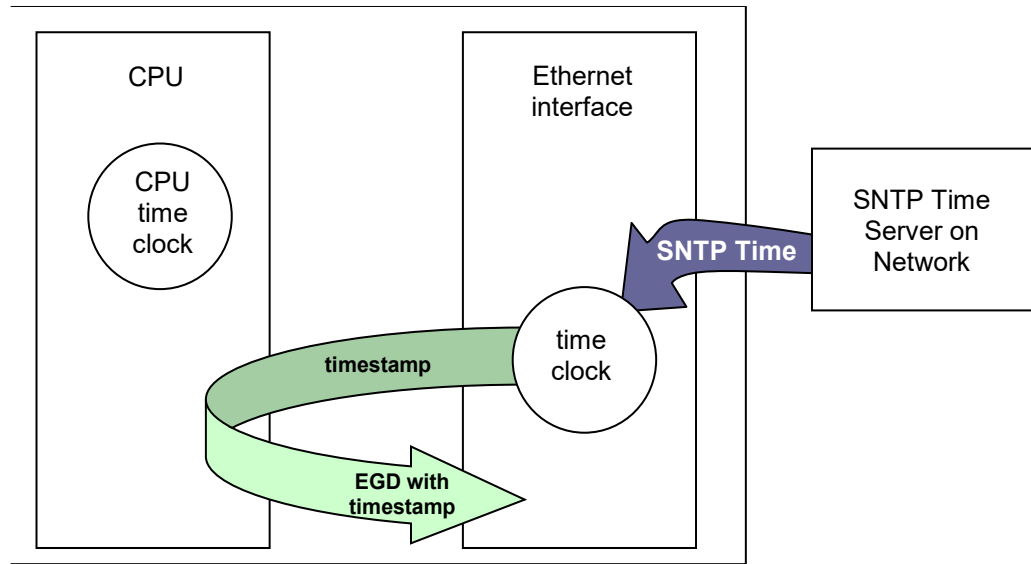
**Figure 46: Obtaining Timestamps from the PLC Time Clock**



## SNTP Server's Time Clock

If this source is selected, the Ethernet interface's built-in clock is periodically synchronized to the clock on an SNTP server on the network. All Ethernet interfaces configured to use SNTP will have updated, synchronized timestamps. Therefore, accurate timing comparisons between exchanged data can be made. If SNTP is used to perform network time synchronization, the timestamp information typically has  $\pm 10$  millisecond accuracy between PLCs on the same network.

Figure 47: Obtaining Timestamps from the SNTP Server's Time Clock



## 7.4.2 Obtaining Timestamps from the CPU TOD Clock

Synchronizing the CPU TOD clock to an SNTP server allows you to set a consistent time across multiple systems. Once the CPU TOD clock is synchronized with the SNTP time on an ETM, all produced EGD exchanges will use the CPU's TOD for the time stamp.<sup>20</sup> The CPU TOD clock is set with accuracy within  $\pm 2$ ms of the SNTP time stamp.

### Synchronizing the CPU TOD Clock to an SNTP Server

The user can synchronize the CPU TOD Clock to an SNTP server using one of two methods:

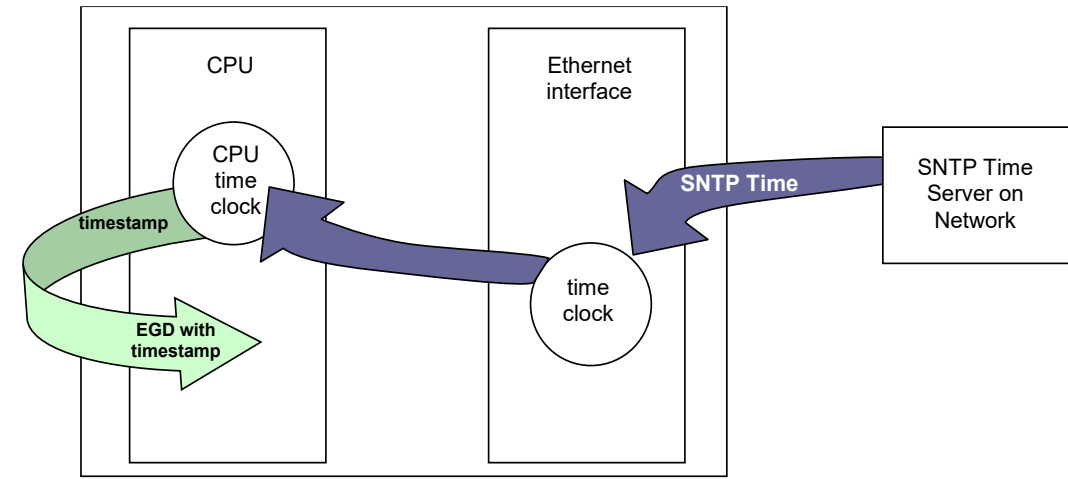
1. Configuring the *Hardware Configuration's* Ethernet settings in PME
2. Configuring and storing an AUP File in PME.

The PME software can be used to enable *SNTP* functionality. Under the ETM001-Kxxx's *Settings* tab, set the *SNTP* parameter to Enabled. This action will add the *SNTP* tab, which will include the *CPU TOD Clock* parameter. Set the *CPU TOD Clock* parameter to *Enabled*.

Alternatively, Ethernet module ETM001-Jx can enable the CPU Time Sync feature by defining the parameter *ncpu\_sync* as *1* in an AUP file. For details on configuring an AUP file, refer to Appendix A.

<sup>20</sup> Embedded EDG produced exchanges will produce GMT time, not local CPU time.  
*SNTP Operation*

Figure 48: Synchronizing CPU Time-of-Day Clock to an SNTP Server



Within a PLC, only one Ethernet interface can be selected as the time master for CPU time synchronization. If multiple Ethernet modules are configured for CPU time synchronization, the PLC application logic should issue a *Read Ethernet Clock Status and Stratum COMMREQ (5001)* to each configured module.

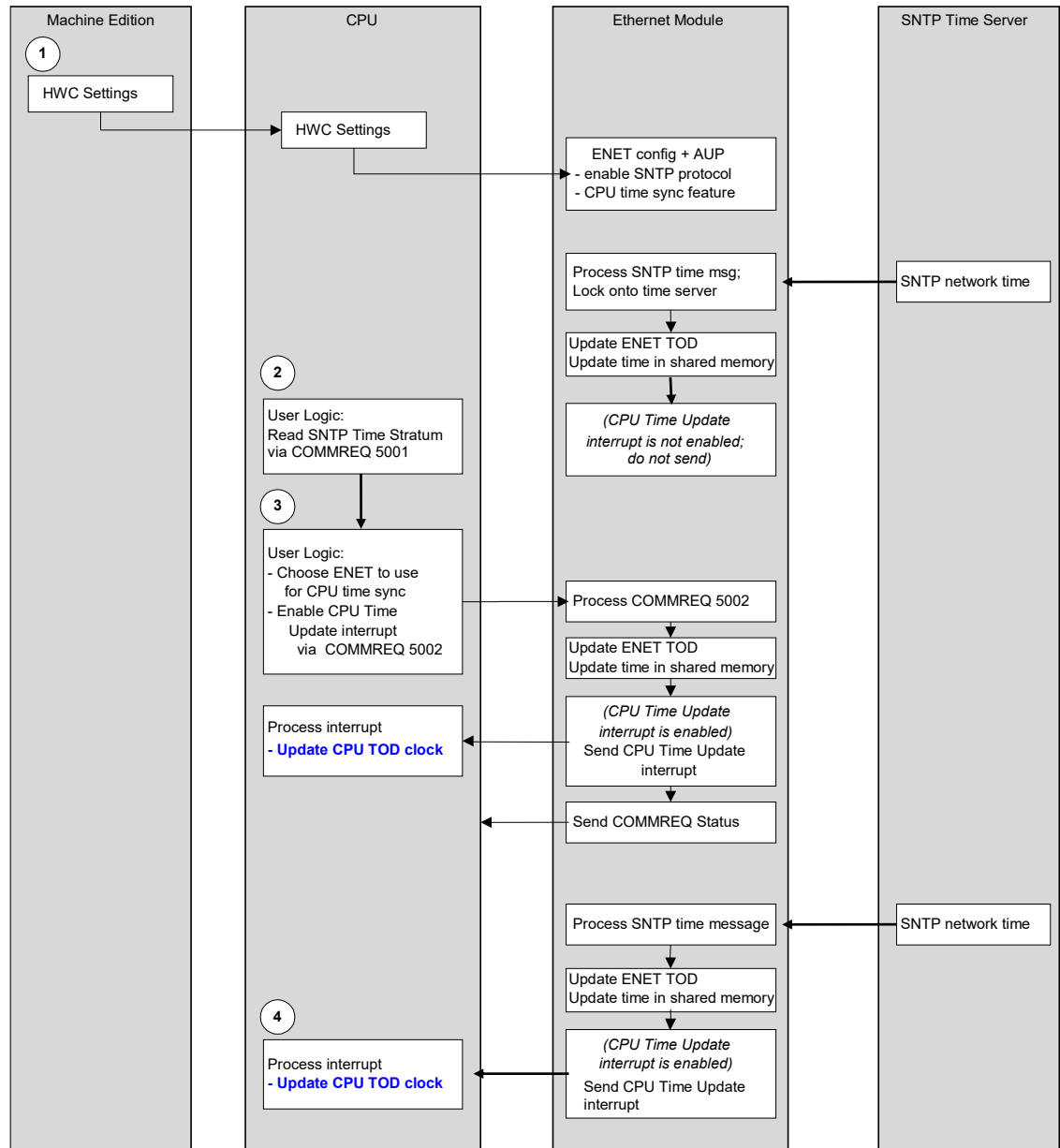
The application logic must examine the stratum number at each Ethernet module to determine which Ethernet module to select. When the application has determined which module to use as the time master, it must send an *Enable PLC Time Update COMMREQ (5002)* to that module.

When the CPU TOD is used for EGD time stamps, it continues until a STOP transition occurs. On a RUN to STOP transition, the CPU disables CPU TOD clock synchronization. The PLC application logic must enable CPU TOD clock synchronization by sending an *Enable PLC Time Update COMMREQ (5002)* on every STOP to RUN transition. For an overview of this operating sequence, see page 139.

## Operating Sequence for CPU Clock Synchronization

The following diagram illustrates the sequence of events for the setup and operation of a system that uses clock synchronization using an HWC Ethernet Settings.

Figure 49: Operating Sequence for CPU Clock Synchronization





## RX3i Model Comparison for SNTP Support

PACSystems RX3i Ethernet interfaces can synchronize their clocks by modifying their AUPs, which can be conducted one of two ways:

- Updating the parameters using an AUP File that is stored to PME.
- Updating the parameters under the interface’s settings panel in PME

The table below specifies each interface’s ability to update its AUPs.

	SNTP Support via AUP Files	SNTP Support via PME Hardware Configuration
CPE302	X	
CPE305	X	
CPE310	X	
CPE330		X
CPE400		X
CPL410		X
CPE100		
CPE115		X
ETM001-Jx	X	X
ETM001-Kxxx	X <sup>21</sup>	X

### Steps to Synchronize the CPU TOD Clock to an SNTP Server (using an AUP file)

1. The user logic program uses the *Read Ethernet Clock Status and Stratum COMMREQ (5001)* to obtain clock status and stratum for each feature-enabled Ethernet interface. The user logic program selects the Ethernet interface advertising the lowest SNTP stratum value to use for CPU time synchronization.
2. The application logic program enables CPU time update for the selected Ethernet interface via the *Enable PLC Time Update COMMREQ (5002)*. If the Ethernet interface is already locked to an SNTP time server on the network, the CPU immediately updates its TOD clock.
3. At every subsequent periodic network time message from the locked SNTP time server, the CPU receives the network time and immediately updates its TOD clock.

**Note:** In a PLC with only one Ethernet interface specified, the logic program may skip step 1. There is no need to select between multiple Ethernet interfaces.

### Steps to Synchronize the CPU TOD Clock to an SNTP Server (using PME)

To synchronize the CPU TOD Clock to an SNTP Server using PME, the user enables *SNTP* in the *Network Time Sync* parameter in the Settings tab of the module. This synchronization method is available when storing an ETM001-Jx HWC that includes an AUP file or either an ETM001-Jx or

---

<sup>21</sup> The ET001-Kxxx only supports the use of AUP files when configured as a ETM001-Jx (such as when used as a drop-in replacement.)

ETM001-Kxxx. Configuration with an AUP file will no longer be supported. To begin, select the ETM module you would like to use to synchronize to the SNTP server:

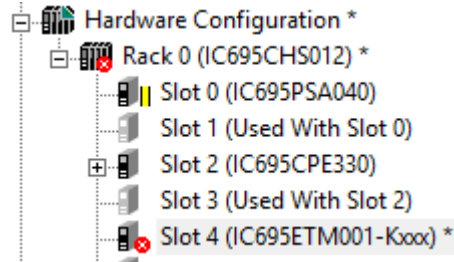


Figure 50 Hardware Configuration Module in PME

When *Network Time Sync* is set to *SNTP*, a new *SNTP* tab becomes available. The newly available *SNTP* tab will allow the user to configure the following parameters:

SNTP Parameters Available in PME	Values
CPU TOD Clock Sync	Enabled or Disabled.
SNTP Mode	<b>Multicast/Broadcast:</b> a one-to-many connection that periodically sends a request (a) local address(es). <b>Unicast:</b> a one-to-one connection that periodically sends a request to a dedicated SNTP server.
Primary Address	(Only available in <b>unicast</b> mode.) The primary address identifies hosts on a single network link. The primary address should specify the primary server. Use of <b>BOOTP</b> must be set to True for IP address of 0.0.0.0
Secondary Address	(Only available in <b>unicast</b> mode.) If the primary server is down, the secondary address can be polled.
Poll Interval <sup>22</sup>	<b>Poll Interval</b> (Interval for unicast, in seconds, at which new time requests are sent to the server): <i>Low Limit =16, High Limit=1024, modulus 2</i>
Poll Count	<b>Poll Count</b> (Number of retransmissions that will be sent when no timely response is received from the server): <i>Low Limit =1, High Limit=100</i>
Poll Timeout	<b>Poll Timeout</b> (The time, in seconds, to wait for a response from the server): <i>Low Limit =2, High Limit=100</i>

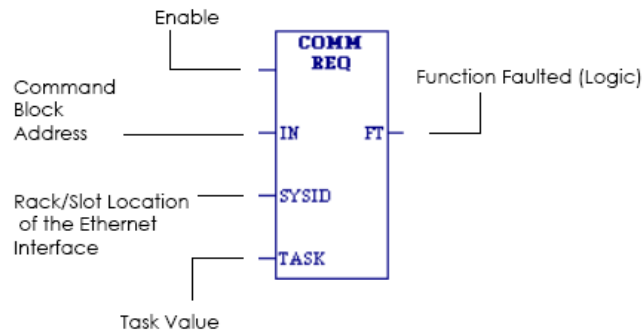
For more information on these SNTP parameters, please refer to Section 7, *SNTP Operation*.

<sup>22</sup> Poll Interval is only available when Unicast is selected.  
*SNTP Operation*

## SNTP Time Transfer COMMREQs

The PLC application logic uses the following Communication Requests (COMMREQ) functions to control CPU TOD clock synchronization. The Communications Request is triggered when the logic program passes power to the COMMREQ Function Block.

**Figure 51: COMMREQ to Control the CPU Time-of-Day Clock**



### COMMREQ Parameters

Command	Description															
ENABLE	Control logic for activating the COMMREQ Function Block.															
IN	The location of the Command Block. It can be any valid address within a word-oriented area of (%R, %AI, %AQ, %P, %L, or %W).															
SYSID	A hexadecimal word value that gives the rack (high byte) and slot (low byte) location of the Ethernet interface. For the PACSystems CPU embedded Ethernet interface, enter the rack/slot location of the CPU module. <table border="1" data-bbox="581 1178 1081 1371"> <thead> <tr> <th>Rack</th> <th>Slot</th> <th>Hex Word Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>4</td> <td>0004H</td> </tr> <tr> <td>3</td> <td>4</td> <td>0304H</td> </tr> <tr> <td>2</td> <td>9</td> <td>0209H</td> </tr> <tr> <td>4</td> <td>2</td> <td>0402H</td> </tr> </tbody> </table>	Rack	Slot	Hex Word Value	0	4	0004H	3	4	0304H	2	9	0209H	4	2	0402H
Rack	Slot	Hex Word Value														
0	4	0004H														
3	4	0304H														
2	9	0209H														
4	2	0402H														
TASK	For the PACSystems Ethernet module, the Task must be set to 98 (62H). For the PACSystems CPU embedded Ethernet interface, Task must be set to the value 65634 (10062H) to address the CPU's Ethernet daughterboard.															
FT OUTPUT	The FT output is set if the PLC CPU (rather than the Ethernet interface) detects that the COMMREQ fails. In this case, the other status indicators are not updated for this COMMREQ.															

### CAUTION

Entering an incorrect TASK value may cause unexpected results.

## Read Ethernet Clock Status and Stratum COMMREQ (5001)

This COMMREQ is used to read the clock status and stratum from the specified Ethernet interface.

If multiple Ethernet modules are enabled for TOD Clock Synchronization, the application logic must examine the stratum at each Ethernet module to determine which Ethernet module to select. The Ethernet clock status and stratum values from the locked time server (if any) are returned as two consecutive words.

### Command Block for Reading Ethernet Clock Status and Stratum COMMREQ

Word Offset	Value	Description
Word 1	Length of a command data block.	Always 3.
Word 2	0	Always 0 (Wait/No Wait mode request).
Word 3	For a list of memory type codes, see Section 6.4, <i>COMMREQ Status for the EGD Commands</i> .	The memory type of the COMMREQ status word.
Word 4	0-based	The offset of COMMREQ status word. For CRS word values, refer to page 146.
Word 5	0	Always 0.
Word 6	0	Always 0.
Word 7	5001	Read Clock Status and Stratum command number.
Word 8	For a list of memory type codes, see Section 6.4, <i>COMMREQ Status for the EGD Commands</i> .	The memory type of the storage location for the clock status and stratum values retrieved from the Ethernet interface.
Word 9	Any valid offset within memory type specified in Word 8. This is a 1-based number.	Ethernet Clock Status and Stratum reference address offset

## Clock Status and Stratum Format

Clock Status and Stratum PLC memory address	Clock Status
Clock Status and Stratum PLC memory address + 1	Clock Stratum

An Ethernet interface can maintain timing information from up to four SNTP servers at a time. Each server assigns a stratum number that determines its priority.

When locked to a network time server, the Ethernet clock stratum value indicates the accuracy of the time value provided by the server. A stratum value of one indicates the highest accuracy time; a value of 15 indicates the lowest accuracy. A stratum value of 255 indicates that the Ethernet clock is not locked to any timeserver. Before using this stratum value, always check that the corresponding clock status indicates that the Ethernet clock is locked to a network timeserver.

The **Status** word indicates whether the Ethernet clock is locked to a network timeserver.

## Clock Status Word Values

Value	Description
0	Ethernet interface is not configured for SNTP operation
1	Ethernet clock is currently locked to network timer server
2	Ethernet clock is not locked to network timer server

**Note:** Bit 5 in the LAN interface Status (LIS) block indicates whether the Ethernet module is currently locked to an SNTP time server on the network. The logic application can periodically examine this bit to determine when an Ethernet module has lost its lock with a network time server. For details of the LIS block, refer to Section 12.6, *Monitoring the Ethernet Interface Status Bits*.

## Enable or Disable PLC Time Update COMMREQ (5002)

This COMMREQ is used to enable or disable a specific Ethernet interface to update the CPU's TOD clock. When enabled, the Ethernet interface updates the TOD clock each time that a time update message is received from an SNTP server on the network. If the Ethernet interface is locked to a timer server when this COMMREQ command is issued, the Ethernet interface immediately updates the TOD clock with the current synchronized clock value.

### Command Block for Enable/Disable PLC Time Update COMMREQ

Word Offset	Value	Description
Word 1	Length of the command data block.	Always 2
Word 2	0	Always 0 (Wait/No Wait mode request).
Word 3	For a list of memory type codes, see Section 6.4, <i>COMMREQ Status for the EGD Commands</i> .	The memory type of the COMMREQ status word.
Word 4	0-based	The offset of COMMREQ status word. For CRS word values, refer to page 146.
Word 5	0	Always 0.
Word 6	0	Always 0.
Word 7	5002	Enable/Disable Time Update command number
Word 8	1 = Enable PLC time update 0 = Disable PLC time update	This word contains the value to enable or disable this Ethernet interface to update the PLC clock. This word must be set to 0 to disable PLC clock updates, and set to 1 to enable PLC clock updates. All other values will cause COMMREQ to return a failure status.

## COMMREQ Status Word Values

The following table lists the CRS values returned by the SNTP Time Transfer commands. For a discussion of CRS major and minor codes, refer to the section on *Communication Request* in the *PACSystems RX7i, RX3i, and RSTi-EP CPU Reference Manual*, GFK-2222.

Before executing a COMMREQ, the application logic should set the CRS word to 0. After executing a COMMREQ, the application logic should monitor the CRS word to determine the completion and success of that command.

Minor (Hex)	Major (Hex)	Description
00	01	Successful completion.
04	01	Successful completion. The Ethernet interface is not locked to an SNTP server at this time, so the CPU clock was not updated.
05	01	Successful completion. The CPU clock was already synchronized to the SNTP server via this Ethernet interface, so the CPU clock was not updated again.
11	0C	Internal error reading clock status or stratum value from this Ethernet interface. The clock status/stratum values were not returned.
12	0C	Internal error enabling CPU time synchronization. The CPU clock will not be synchronized to an SNTP server at this Ethernet interface.
13	0C	Internal error disabling CPU time synchronization.
07	0D	COMMREQ data block length (COMMREQ word 1) is too short.
08	0D	COMMREQ command code (COMMREQ word 7) is not recognized.
10	0D	CPU and/or ENET firmware version does not support the SNTP Time Transfer feature.
12	0D	Attempted to enable CPU time sync on this Ethernet interface while already enabled on another Ethernet interface. The logic application must first disable CPU time sync on the original Ethernet interface before enabling it on another Ethernet interface.
13	0D	Attempted to disable CPU time sync that was not previously enabled at this Ethernet interface.
14	0D	Invalid COMMREQ command data.
15	0D	COMMREQ is not allowed because the SNTP Time Transfer feature was not configured.
16	0D	COMMREQ data block length (COMMREQ word 1) is too long.

## Local Time and Daylight Savings Time Corrections

Ethernet interfaces support the ability to specify an offset to the Coordinated Universal Time (UTC) to correct for local time zone and daylight savings time (DST). You can specify the DST start/stop times and offset from local standard time, as well as the local time offset from the UTC in the *Hardware Configuration* node. The specified correction is applied to all modes of SNTP communications (Broadcast, Multicast, and Unicast).

To begin, right-click the Ethernet module you wish to configure and select *Configure*. In the opened *Settings* tab, select *Enable* for the *Enable UTC Offset* parameter. In tab *Time*, the user can select the offset value from the dropdown based on the closest time zone.

### Daylight Savings

If a Day Light Savings Time (DST) offset is required, the parameter can also be enabled from the *Settings* tab. After it is enabled, the new DST parameter will become available in the *Time*

The default SNTP operation features *no correction* for local time or DST. Local time and DST corrections must be enabled using the instructions above. Alternatively, previous versions of Ethernet modules can be configured using an AUP file. For local time correction and DST parameters, refer to *Configuring Advanced User Parameters*.



# Section 8 Programming SRTP Channel Commands

This chapter describes how to implement PLC to PLC communications over the Ethernet network using SRTP Channel commands:

- SRTP Channel Commands
  - Channel Operations
  - Aborting and Re-tasking a Channel
  - SRTP Channel Commands in a Redundant System
  - Executing a Channel Command
- COMMREQ Format for Programming Channel Commands
  - Establish Read Channel
  - Establish Write Channel
  - Send Information Report
  - Abort Channel
  - Retrieve Detailed Channel Status
- Programming for Channel Commands
  - COMMREQ Example
  - Sequencing Communications Requests
  - Managing Channels and TCP Connections
  - Use Channel Re-Tasking to Avoid using up TCP Connections
  - Client Channels TCP Resource Management
  - SRTP Application Timeouts
- Monitoring Channel Status

## 8.1 Model Comparison for SRTP Server Capabilities

PACSystems Ethernet interfaces enable client PLCs to communicate with a server PLC through the PAC Machine Edition programming software using SRTP protocol. The following table compares the SRTP capabilities of each RX3i Ethernet interface.

SRTP Capabilities for RX3i Ethernet Interfaces										
	CPE302	CPE305	CPE310	CPE330	CPE400	CPL410	CPE100	CPE115	ETM001-jx	ETM001-kxxx
Simultaneous SRTP Server Connections	32	32	32	48	48	48	16	16	48	48
SRTP Channels	16	16	16	32	32	32	16	16	32	32

## 8.2 SRTP Channel Commands

The SRTP Channel commands are a set of client PLC commands that can be used to communicate with a server PLC.

A Channel command can establish a channel to execute multiple *periodic* reads or writes with a single initiation of a COMMREQ function. A Channel command can also be used to execute a single read or write.

There are five Channel commands:

- Establish Read Channel
- Establish Write Channel
- Send Information Report
- Abort Channel
- Retrieve Detailed Channel Status

### 8.2.1 Channel Operations

Channel commands are based on the concept of periodic data transfers. The client (local) PLC uses a single COMMREQ function to establish a channel (connection) to a server (remote) PLC and to request that specific data be periodically transferred between the PLCs.

The Ethernet interface automatically manages the establishment of communications and the periodic data transfer. Parameters in the Command Block specify the frequency and direction of the transfer, and the memory locations in the client and server to be used in the transfer.

## 8.2.2 Aborting and Re-tasking a Channel

There are four ways a channel can be aborted:

1. When the PLC CPU is stopped, all channels in use are aborted automatically.
2. A channel (or all channels) can be aborted by issuing an Abort Channel command.
3. A channel in use can be re-tasked by issuing an established command for its channel number. This aborts the previous channel operation and then performs the new channel operation.
4. A channel is also automatically aborted if a fatal error occurs.

## 8.2.3 Monitoring the Channel Status

The Ethernet interface status bits occupy a single block of memory, which is specified during the configuration of the Ethernet interface. The status bits include Channel Status bits, which provide runtime status information for each communication channel. Each channel has two status bits; the meaning of the channel status bits depends upon the type of communication performed on that channel.

SRTP channels operation provides two Channels Status bits for each SRTP channel, a Data Transfer bit and a Channel Error bit.

For details of the status bits and their operation, refer to Section 12.6, *Monitoring the Ethernet Interface Status Bits*.

## 8.2.4 SRTP Channel Commands in a Redundant System

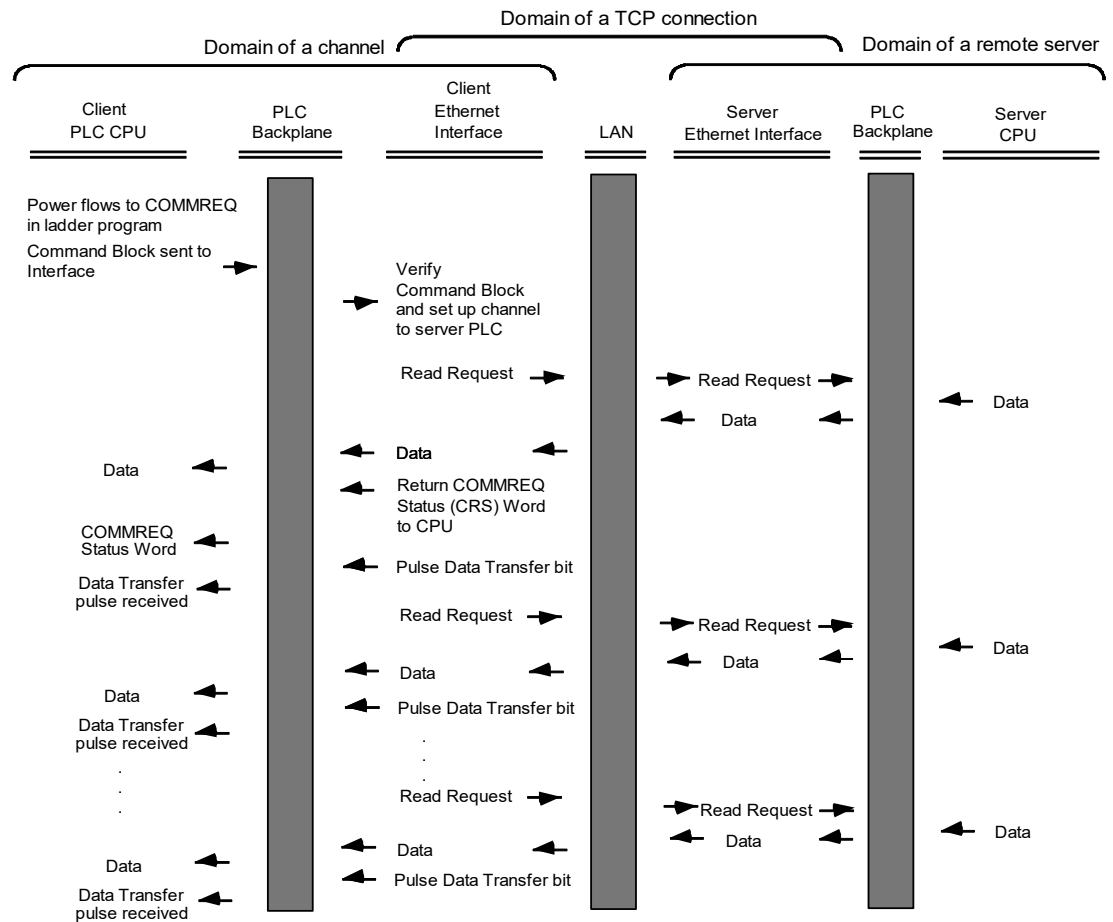
When configured for Redundant IP operation (see Section 1 for more information), only the active unit establishes and maintains the SRTP Client connections used for the Channel commands. The backup unit does not perform any SRTP Client operations. If SRTP Client operation is attempted, a COMMREQ error status is returned to the local logic program. When the Ethernet interface changes from active to backup state, it takes down all SRTP Client connections and their underlying TCP connections.

Because it can take some time to take down a TCP connection, the Redundant system should reserve a spare SRTP Client connection for each connection using the Redundant IP address. That will prevent temporary resource problems when establishing new SRTP Client connections to the new active unit while the previous connections to the old active unit are being taken down.

## 8.2.5 Executing a Channel Command

The figure below shows how a Communications Request carries out a Channel command, in this case, Establish Read Channel.

**Figure 52: COMMREQ Sequence for Establish Read Channel**



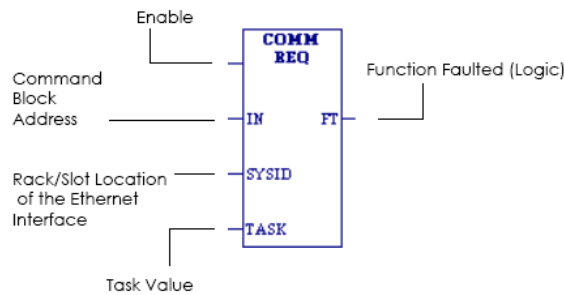
1. The command starts when there is power flow to a COMMREQ function in the client PLC. At this time, the Command Block data is sent from the PLC CPU to the Ethernet interface.
2. For the Establish Read Channel command, the COMMREQ status word is returned immediately if the Command Block is invalid. If the syntax is correct, the COMMREQ status word is returned after the first significant event: upon failure to establish a channel correctly and promptly or upon the first successful transfer of data.
3. After the channel is successfully set up to the server PLC, the Ethernet interface performs the periodic reads as specified in the Command Block.

## 8.3 COMMREQ Format for Programming Channel Commands

The Channel commands described in this chapter are sent using the Communications Request (COMMREQ) function.

The Communications Request is triggered when the logic program passes power to the COMMREQ Function Block.

**Figure 53: COMMREQ for Programming Channel Commands**



For the Channel Commands, the parameters of the COMMREQ are:

Command	Description																		
<b>ENABLE</b>	Control logic for activating the COMMREQ Function Block.																		
<b>IN</b>	The location of the Command Block. It can be any valid address within a word-oriented area of (%R, %AI, %AQ, %P, %L, or %W).																		
<b>SYSID</b>	A hexadecimal word value that gives the rack (high byte) and slot (low byte) location of the Ethernet interface. For the PACSystems CPU embedded Ethernet interface, enter the rack/slot location of the CPU module. The adapter name (module position) can be pointed by the hex word value in the table below: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Rack</th> <th>Slot</th> <th>Hex Word Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0000H CPE400/CPL410 and CPE100/CPE115)</td> </tr> <tr> <td>0</td> <td>4</td> <td>0004H</td> </tr> <tr> <td>3</td> <td>4</td> <td>0304H</td> </tr> <tr> <td>2</td> <td>9</td> <td>0209H</td> </tr> <tr> <td>4</td> <td>2</td> <td>0402H</td> </tr> </tbody> </table>	Rack	Slot	Hex Word Value	0	0	0000H CPE400/CPL410 and CPE100/CPE115)	0	4	0004H	3	4	0304H	2	9	0209H	4	2	0402H
Rack	Slot	Hex Word Value																	
0	0	0000H CPE400/CPL410 and CPE100/CPE115)																	
0	4	0004H																	
3	4	0304H																	
2	9	0209H																	
4	2	0402H																	
<b>TASK</b>	For the PACSystems Ethernet module, the Task must be set to zero. For the PACSystems CPU embedded Ethernet interface, Task must be set to the value 65536 (10000H)																		
<b>FT OUTPUT</b>	The FT output is set if the PLC CPU (rather than the Ethernet interface) detects that the COMMREQ fails. In this case, the other status indicators are not updated for this COMMREQ.																		

## 8.3.1 The COMMREQ Command Block: General Description

When the COMMREQ function is initiated, the Command Block is sent from the PLC CPU to the Ethernet interface. The Command Block contains the details of a Channel command to be performed by the interface.

- The address in CPU memory of the Command Block is specified by the IN input of the COMMREQ Function Block.
- The address can be any valid address within a word-oriented area of memory (%R, %AI, %AQ, %P, %L, or %W).
- The Command Block is set up using an appropriate programming instruction, such as a BLOCK MOVE or DATA\_INIT\_COMM). The Command Block has the following structure:

Word Offset	Description
Word 1	Data Block Length (words)
Word 2	WAIT/NOWAIT Flag
Word 3	COMMREQ status word Memory Type
Word 4	COMMREQ status word Address Offset
Word 5	Reserved
Word 6	Reserved
Words 7 and up	Data Block (Channel Command Details)

### ***Further Considerations for Memory Locations 1-7***

**(Word 1) Data Block Length:** This is the length in words of the Data Block portion of the Command Block. The Data Block portion starts at Word 7 of the Command Block. The length is measured from the beginning of the Data Block at Word 7, not from the beginning of the Command Block. The correct value for each command, and the associated length of each command, is specified in the next section.

**(Word 2) WAIT/NOWAIT Flag:** Must be set to zero for TCP/IP Ethernet Communications.

**COMMREQ Status Word:** The Ethernet interface updates the COMMREQ status word to show success or failure of the command. Command words 3 and 4 specify the PLC memory location of the COMMREQ status word. For more information, refer to COMMREQ Status Word Memory Type in this section.

**(Word 3) COMMREQ Status Word Memory Type:** This word specifies the memory type for the COMMREQ status word.

**(Word 4) COMMREQ Status Word Address Offset:** This word contains the offset within the memory type selected. *The status word address offset is a zero-based number.* For example, if you want %R1 as the location of the COMMREQ status word, you must specify a zero for the offset. The offset for %R100 would be 99 decimal. Note, however, that this is the only zero-based field in the Channel commands. (When using %W memory, the maximum offset value that can be entered is 65535, signifying %W65536.)

**(Word 5):** Reserved. Set to zero.

**(Word 6):** Reserved. Set to zero.

**(Words 7 and up) Data Block:** The Data Block defines the Channel command to be performed.

## COMMREQ Status Word Memory Type

This word specifies the memory type for the COMMREQ status word. The memory types are listed in the table below:

Type	Value (Decimal)	Value (Hex.)	Description
%R	8	08H	Register memory (word mode)
%AI	10	0AH	Analog input memory (word mode)
%AQ	12	0CH	Analog output memory (word mode)
%I	16	10H	Discrete input memory (byte mode)
	70	46H	Discrete input memory (bit mode)
%Q	18	12H	Discrete output memory (byte mode)
	72	48H	Discrete output memory (bit mode)
%T	20	14H	Discrete temporary memory (byte mode)
	74	4AH	Discrete temporary memory (bit mode)
%M	22	16H	Discrete momentary internal memory (byte mode)
	76	4CH	Discrete momentary internal memory (bit mode)
%G	56	38H	Discrete global data table (byte mode)
	86	56H	Discrete global data table (bit mode)
%W	196	C4H	Word memory (word mode; limited to %W1 through %W65536)

## Using COMMREQs for Channel Commands

- Be sure to use unique COMMREQ Status (CRS) memory locations for each COMMREQ.
- Always initialize the COMMREQ Status Word to zero before initiating a Channel command COMMREQ to a given channel. Wait for the COMMREQ Status Word to go to a non-zero value (which signals the COMMREQ is complete) before issuing another Channel command to that channel. The COMMREQ Status Word is updated once per COMMREQ execution: a non-zero value in the status word completes the COMMREQ.
- Always use a one-shot to initiate a Channel command COMMREQ. That prevents the channel COMMREQ from being executed each CPU scan, which would overrun the capability of the Ethernet interface.
- The COMMREQ Status is issued only once for each COMMREQ; for more information, refer to
- *Using COMMREQs for Channel Commands*. The pending transfer can still complete after the period error occurs. You can determine when the pending transfer completes by monitoring the Channel Error and Data Transfer bits. For Channel commands set up to issue multiple reads, the next read transfer will be issued only after the pending read transfer completes.

## 8.3.2 Establish Read Channel (2003)

The Establish Read Channel command requests that a channel be associated with a remote PLC and that data from the remote PLC be transferred (periodically) to the local PLC. The Command Block specifies the period, the number of reads from the remote PLC to perform, and the timeout allowed in waiting for each transfer to complete. The first read is performed immediately, regardless of the period specified.

### Example Command Block

Establish a channel (Channel 5) to a remote PLC at IP address 10.0.0.1. Return the COMMREQ Status word to %R10. Read remote PLC registers %R50–%R57 to local PLC registers %R100–%R107. Repeat the read ten times, once every 7 seconds, with a timeout of 500ms for each read.

Word Offset	Dec	(Hex)	Description
Word 1	00017	(0011)	Length of Channel Command Data Block (17–25 words)
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	The memory type of COMMREQ status word (%R)
Word 4	00009	(0009)	COMMREQ status word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	02003	(07D3)	Establish Read Channel command number
Word 8	00005	(0005)	Channel number (5)
Word 9	00010	(000A)	Number of read repetitions (read 10 times)
Word 10	00003	(0003)	Time unit for read period (3=seconds)

### Further Considerations for Memory Locations 1-10

**(Word 7) Channel Command Number:** Word 7 requests that a read channel be set up. If the command is processed successfully, it will attempt the specified number of transfers from the server to the client.

**(Word 8) Channel Number:** Word 8 specifies the channel to be used for the read. This value must be in the range of 1–32. If the channel number is out of range, a command error indication will be placed in the COMMREQ Status word. If the channel number is the same as a channel already in use, the channel will be re-tasked to perform this new command.

**(Word 9) Number of Read Repetitions:** Word 9 specifies the number of reads to be performed before automatically completing the communications request and closing the channel. If this value is set to 1, only a single read will be issued. If this value is set to 0, reads will be issued continuously on the requested period until the channel is aborted.

**(Word 10) Time Unit for Read Period:** Words 10–11 together define how often the read is to be performed (*read period*). Word 10 specifies the time unit such as seconds or minutes for the read period. Word 11 specifies the number of those units. The choices for the time units are shown below.

Value	Meaning
1	hundredths of seconds (10ms)
2	tenths of seconds (100ms)
3	seconds
4	minutes
5	hours



Word Offset	Dec	(Hex)	Description
Word 11	00007	(0007)	Number of time units for read period (every 7 seconds)
Word 12	00050	(0032)	Timeout for each read (500ms)
Word 13	00008	(0008)	Local PLC - Memory type at which to store data (%R)
Word 14	00100	(0064)	Local PLC - Starting address at which to store data (%R100)
Word 15	00008	(0008)	Remote PLC - Memory type from which to read data (%R)

### Further Considerations for Memory Locations 11-15

**(Word 11) Number of Time Units for Read Period:** Word 11 specifies the number of time units for the read period. The read period is in effect even when the Channel command is setup to issue a single read.

**Example Read Period Calculation:** If Word 10 contains a value of 3 specifying seconds as the time unit and Word 11 contains a value of 20, then the read period is 20 seconds.

A Channel command set up to issue a single read can have only one *pending* read transfer. A read will normally be issued at the start of each read period. If the *pending* read transfer has not completed during the read period, the Channel Error bit and Detailed Channel Status words will be set to indicate a non-fatal period error. If the period error occurs on the first transfer, the COMMREQ Status will also indicate a non-fatal period error. If the Number of Time Units is zero, a subsequent transfer will be issued as soon as the previous transfer completes. In this case, no period errors can occur.

**(Word 12) Timeout for Each Read:** Word 12 specifies the time (in hundredths of a second) the Ethernet interface will wait for a read transfer to complete before setting the Channel Error bit and Detailed Channel Status words to indicate a non-fatal timeout error. If the timeout error occurs on the first transfer, the COMMREQ Status will also indicate a non-fatal timeout error.

For most applications, a timeout is not needed because the read period acts as a timeout. (Word 12 should be zero for no timeout). However, there are two circumstances in which specifying a timeout is recommended:

- When the number of time units (Word 11) is zero, so that a subsequent transfer will be issued as soon as the previous transfer completes and no period errors are reported. In this case, a timeout value can be specified so that the Channel Error bit will report timeout errors.
- When the read period is very long (minutes or hours). In this case, a shorter timeout value can be specified so the application doesn't have to wait for the read period to expire before acting.

**(Word 13) Local PLC - Memory Type:** Words 13–14 specify the location in the local PLC where the Ethernet interface will store data received from the remote PLC. Please see the table in the following section. The amount of data to be transferred is specified by the number of memory units of the data read from the remote PLC (Word 17). For more information, refer to *Read Channel Memory Types Available* in this section.

**(Word 14) Local PLC - Memory Starting Address:** Word 14 determines the starting address in the local PLC in which the data from the remote PLC is to be stored. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 13. This offset will be either in bits, bytes, or words depending on the mode specified (for example, if Word 13=16 and Word 14=2, then the starting address will be %I9). Valid ranges of values depend on the PLC's memory ranges. The user is responsible for assuring that this area is large enough to contain the requested data without overwriting other application data.

**(Word 15) Remote PLC - Memory Type:** Words 15–16 specify the memory type and starting address in the remote PLC from which the data is to be read. Valid values for Word 15 are listed above. If %P memory is used, you must specify a Program name in Words 24–27. If %L memory is used, you must specify a Program name in Words 24–27 and a Program Block name in Words 28–31.

Word Offset	Dec	(Hex)	Description
Word 16	00050	(0032)	Remote PLC - Starting address from which to read data (%R50)
Word 17	00008	(0008)	Remote PLC - Number of memory units (8 registers)
Word 18	00001	(0001)	Remote PLC - Network Address type (IP Address)
Word 19	00004	(0004)	Remote PLC - Network Address length in words (4)
Word 20	00010	(000A)	Remote PLC - Register 1 of IP address (10)
Word 21	00000	(0000)	Remote PLC - Register 2 of IP address (0)
Word 22	00000	(0000)	Remote PLC - Register 3 of IP address (0)
Word 23	00001	(0001)	Remote PLC - Register 4 of IP address (1)
Word 24–27			Remote PLC - Program Name (needed for access to remote %P or %L) (zero-terminated and padded)
Word 28–31			Remote PLC - Program Block (needed for access to remote %L) (zero-terminated and padded)

### ***Further Considerations for Memory Locations 16-31***

**(Word 16) Remote PLC - Memory Starting Address:** Word 16 determines the starting address in the remote PLC from which the data is to be read. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 15. This offset will be either in bits, bytes, or words depending on the mode specified (for example, if Word 15=16 and Word 16=9, then the starting address will be %I65). Valid ranges of values depend on the remote PLC's memory ranges.

**(Word 17) Remote PLC - Number of Memory Units:** Word 17 specifies the amount of data to be transferred. The value entered is the number of memory units to be transferred, where the size of a memory unit is a bit, byte, or word as specified in Word 15. For example, if Word 15=16 and Word 17=4, then 4 bytes (32 bits) of %I memory will be transferred. A maximum of 8192bits, 1024 bytes, or 512 words of data can be specified.

**(Word 18) Remote PLC - Network Address Type:** Word 18 specifies the format of the remote PLC address. Word 18 must contain the value 1. This indicates a dotted-decimal IP address expressed using a separate register for each decimal digit.

**(Word 19) Remote PLC - Network Address Length:** Word 19 specifies the length in words of the remote PLC IP address. Word 19 must contain 4.

**(Words 20–23) Remote PLC - IP Address:** Words 20–23 specify the four integers, one integer per word, of the dotted-decimal IP address of the remote PLC to be accessed.

**(Words 24–27) Remote PLC - Program Name:** Words 24–27 specify the case-sensitive, zero-terminated and padded program name (also called task name, which can be found through the PROG Station Manager command on the server Ethernet interface) to be used with access to remote %P or %L memory. These words are required only for access to such memory and will be ignored if the Memory Type field is not %P or %L. See Note below.

**(Words 28–31) Remote PLC - Program Block Name:** Words 28–31 specify the case-sensitive, zero-terminated and padded program block name (which can be found in the program block declaration in the server ladder program) to be used with access to remote %L memory. These words are required only for access to such memory and will be ignored if the Memory Type field is not %P or %L.

## Read Channel Memory Types Available

Words 13–14 specify the location in the local PLC where the Ethernet interface will store data received from the remote PLC. Valid values for Word 13 are listed below. The amount of data to be transferred is specified by the number of memory units of the data read from the remote PLC (Word 17).

Type	Value (Decimal)	Description
%L <sup>23</sup>	0	Program Block Local register memory (word mode)
%P	4	Program register memory (word mode)
%W <sup>24</sup>	196	Word memory (word mode; max address %W65535)
%R	8	Register memory (word mode)
%AI	10	Analog input memory (word mode)
%AQ	12	Analog output memory (word mode)
%I	16	Discrete input memory (byte mode)
	70	Discrete input memory (bit mode)
%Q	18	Discrete output memory (byte mode)
	72	Discrete output memory (bit mode)
%T	20	Discrete temporary memory (byte mode)
	74	Discrete temporary memory (bit mode)
%M	22	Discrete momentary internal memory (byte mode)
	76	Discrete momentary internal memory (bit mode)
%SA	24	Discrete system memory group A (byte mode)
	78	Discrete system memory group A (bit mode)
%SB	26	Discrete system memory group B (byte mode)
	80	Discrete system memory group B (bit mode)
%SC	28	Discrete system memory group C (byte mode)
	82	Discrete system memory group C (bit mode)
%S <sup>18</sup>	30	Discrete system memory (byte mode)
	84	Discrete system memory (bit mode)
%G	56	Discrete global data table (byte mode)
	86	Discrete global data table (bit mode)

**Note:** The Program Name (Words 24–27) and Program Block Name (Words 28–31) must have each pair of ASCII characters reversed within the PLC memory. For example, the name “MARY” (“M” = 4DH, “A” = 41H, “R” = 52H, “Y” = 59H) would have 414DH in the first word and 5952H in the second word.

<sup>23</sup> Can only be accessed in the Remote PLC

<sup>24</sup> %W memory is supported by PACSystems clients and servers only.

### 8.3.3 Establish Write Channel (2004)

The Establish Write Channel command requests that a channel be connected to a remote PLC and that data from the local PLC be transferred (periodically) to the remote PLC. The Command Block specifies the period, the number of writes to the server (remote PLC) to perform, and the timeout allowed in waiting for each transfer to complete. The first write is performed immediately, regardless of the period specified.

#### Example Command Block

Establish a write channel (Channel 6) to a remote PLC at IP address 10.0.0.1. Return the COMMREQ Status word to %R10. Write local PLC registers %R50–%R57 to remote PLC registers %R100–%R107. Repeat the write indefinitely, once every 7 seconds, with a timeout of 500ms for each write.

Word Offset	Dec	(Hex)	Description
Word 1	00017	(0011)	Length of Channel Command Data Block (17–25 words)
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of COMMREQ status word (%R)
Word 4	00009	(0009)	COMMREQ status word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	02004	(07D4)	Establish Write Channel command number
Word 8	00006	(0006)	Channel number (6)
Word 9	00000	(0000)	Number of write repetitions (write indefinitely)
Word 10	00003	(0003)	Time unit for write period (3=seconds)

#### Further Considerations for Memory Locations 1-10

**(Word 7) Channel Command Number:** Word 7 requests that a write channel be set up. If the command is processed successfully, it will result in attempting the specified number of transfers from the client to the server.

**(Word 8) Channel Number:** Word 8 specifies the channel to be used for the write. This value must be in the range of 1–32. If the channel number is out of range, a command error indication will be placed in the COMMREQ Status word. If the channel number is the same as a channel already in use, the channel will be re-tasked to perform this new command.

**(Word 9) Number of Write Repetitions:** Word 9 specifies the number of writes to be performed before automatically completing the communications request and closing the channel. If this value is set to 1, only a single write will be issued. If this value is set to 0, writes will be issued on the requested period until the channel is aborted.

**(Word 10) Time Units for Write Period:** Words 10–11 together define how often the write is to be performed (*write period*). Word 10 specifies the time unit such as seconds or minutes for the write period. Word 11 specifies the number of those units. The choices for the time units are:

Value	Meaning
1	hundredths of seconds (10ms)
2	tenths of seconds (100ms)
3	Seconds
4	minutes
5	hours

Word Offset	Dec	(Hex)	Description
Word 11	00007	(0007)	Number of time units for write period (every 7 seconds)
Word 12	00050	(0032)	Timeout for each write (500ms)
Word 13	00008	(0008)	Local PLC - Memory type from which to write data (%R)
Word 14	00050	(0032)	Local PLC - Starting address from which to write data (%R50)
Word 15	00008		Remote PLC - Memory type at which to store data (%R)

### Further Considerations for Memory Locations 11-15

**(Word 11) Number of Time Units for Write Period:** Word 11 specifies the number of time units for the write period. The write period is in effect even when the Channel command is setup to issue a single write.

**Example Write Period Calculation:** If Word 10 contains a value of 3 specifying seconds as the time unit and Word 11 contains a value of 20, then the write period is 20 seconds.

A Channel command set up to issue a single write can have only one **pending** write transfer. A write will normally be issued at the start of each write period. If the **pending** write transfer has not completed during the write period, the Channel Error bit and Detailed Channel Status words will be set to indicate a non-fatal period error. If the period error occurs on the first transfer, the COMMREQ Status will also indicate a non-fatal period error.

If the Number of Time Units is zero, a subsequent transfer will be issued as soon as the previous transfer completes. In this case, no period errors are reported by the Channel Error bit.

**(Word 12) Timeout for Each Write:** Word 12 specifies the time (in hundredths of a second) the Ethernet interface will wait for a write transfer to complete before setting the Channel Error bit and Detailed Channel Status bits to indicate a non-fatal timeout error. If the timeout error occurs on the first transfer, the COMMREQ Status (will also indicate a non-fatal timeout error.

For most applications, a timeout is not needed because the write period acts as a timeout. (Word 12 should be zero for no timeout.) However, there are two special circumstances in which specifying a timeout is recommended:

- When the number of time units (Word 11) is zero, so that a subsequent transfer will be issued as soon as the previous transfer completes and no period errors are reported. In this case, a timeout value can be specified so that the Channel Error bit will report timeout errors.
- When the write period is very long (minutes or hours). In this case, a shorter timeout value can be specified so the application doesn't have to wait for the write period to expire before taking action.

**(Word 13) Local PLC - Memory Type:** Words 13–14 specify the location in the local PLC where the Ethernet interface will get the data to be written to the remote PLC. Valid values for Word 13 are listed in the description of Establish Read Channel. The amount of data to be transferred is specified by the number of memory units of the data written to the remote PLC (Word 17).

**(Word 14) Local PLC - Memory Starting Address:** Word 14 determines the starting address in the local PLC from which the data is to be written. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 13. This offset will be in bits, bytes, or words depending on the mode specified (for example, if Word 13=16 and Word 14=2, then the starting address will be %I9). Valid ranges of values depend on the PLC's memory ranges.

**(Word 15) Remote PLC - Memory Type:** Words 15–16 specify the memory type and starting address in the remote PLC where the data is to be written. Valid values for Word 15 are listed under Establish Read Channel. If %P memory is used, you must specify a Program name in Words 24–27. If %L memory is used, you must specify a Program name in Words 24–27 and a Program Block name in Words 28–31.

Word Offset	Dec	(Hex)	Description
Word 16	00100	(0064)	Remote PLC - Starting address at which to store data (%R100)
Word 17	00008	(0008)	Remote PLC - Number of memory units (8 registers)
Word 18	00001	(0001)	Remote PLC - Network Address type (IP address)
Word 19	00004	(0004)	Remote PLC - Network Address length in words (4)
Word 20	00010	(000A)	Remote PLC - Register 1 of IP address (10)
Word 21	00000	(0000)	Remote PLC - Register 2 of IP address (0)
Word 22	00000	(0000)	Remote PLC - Register 3 of IP address (0)
Word 23	00001	(0001)	Remote PLC - Register 4 of IP address (1)
Word 24–27			Remote PLC - Program Name (needed for access to remote %P or %L) (zero-terminated and padded)
Word 28–31			Remote PLC - Program Block (needed for access to remote %L) (zero-terminated and padded)

### Further Considerations for Memory Locations 16-31

**(Word 16) Remote PLC - Memory Starting Address:** Word 16 determines the starting address in the remote PLC where the data is to be written. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 15. This offset will be either in bits, bytes, or words depending on the mode specified (for example, if Word 15=16 and Word 16=9, then the starting address will be %I65). Valid ranges of values depend on the remote PLC's memory ranges.

**(Word 17) Remote PLC - Number of Memory Units:** Word 17 specifies the amount of data to be transferred. The value entered is the number of memory units to be transferred, where the size of a memory unit is a bit, byte, or word as specified in Word 15. For example, if Word 15=16 and Word 17=4, then 4 bytes (32 bits) of %I memory will be transferred. The user is responsible for assuring that this area is large enough to contain the requested data without overwriting other application data. A maximum of 8192 bits, 1024 bytes, or 512 words of data can be specified.

**(Word 18) Remote PLC - Network Address Type:** Word 18 specifies the format of the remote PLC address. Word 18 must contain the value 1, indicates a dotted-decimal IP address expressed using a separate register for each decimal digit.

**(Word 19) Remote PLC - Network Address Length:** Word 19 specifies the length in words of the remote PLC IP address. Word 19 must contain 4.

**(Words 20–23) Remote PLC - IP Address:** Words 20–23 specify the four integers, one integer per word, of the dotted-decimal IP address of the remote PLC to be accessed.

**(Words 24–27) Remote PLC - Program Name:** Words 24–27 specify the case-sensitive, zero-terminated and padded program name (also called task name, which can be found through the PROG Station Manager command on the server Ethernet interface) to be used with access to remote %P or %L memory. These words are required only for access to such memory and will be ignored if the Memory Type field is not %P or %L.

**(Words 28–31) Remote PLC - Program Block Name:** Words 28–31 specify the case-sensitive, zero-terminated and padded program block name (which can be found in the program block declaration in the server ladder program) to be used with access to remote %L memory. These words are required only for access to such memory and will be ignored if the Memory Type field is not %P or %L.

The Program Name (Words 24–27) and Program Block Name (Words 28–31) must have each pair of ASCII characters reversed within the PLC memory. For example, the name "MARY" ("M" = 4DH, "A" = 41H, "R" = 52H, "Y" = 59H) would have 414DH in the first word and 5952H in the second word.

### 8.3.4 Send Information Report (2010)

The Send Information Report COMMREQ requests that a particular block of memory within the PLC CPU reference tables be transferred periodically from a third-party Ethernet interface to a host application SRTP server. Currently, PACS Ethernet devices do not support this server capability. The Command Block specifies the repetition period, the number of transfers to the server to perform, and the timeout allowed in waiting for each transfer to complete. The first send is performed immediately, regardless of the period specified.

#### Example Command Block

Establish a channel (Channel 7) to a remote Host application server at IP address 10.0.0.1. Return the COMMREQ Status word to %R10. Send local PLC registers %R50-%R57 to a remote host. Repeat the send 10 times, once every 7 seconds, with a timeout of 500ms for each transfer.

Word Offset	Dec	(Hex)	Description
Word 1	00017	(0011)	Length of Send Information Report Data Block (17 words)
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	The memory type of COMMREQ status word (%R)
Word 4	00009	(0009)	COMMREQ status word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	02010	(07DA)	Send Information Report Channel command number
Word 8	00007	(0007)	Channel number (7)
Word 9	00010	(000A)	Number of repetitions (send 10 times)
Word 10	00003	(003)	Time unit for send period (3=seconds)

#### Further Considerations for Memory Locations 1-10

**(Word 7) Channel Command Number:** Word 7 requests that a Send Information Report channel be set up. If the command is processed successfully, it will result in attempting the specified number of transfers from the client to the server.

**(Word 8) Channel Number:** Word 8 specifies the channel to be used for the send. This value must be in the range of 1–32. If the channel number is out of range, a command error indication is placed in the COMMREQ status word. If the channel number is the same as a channel already in use, the channel is re-tasked to perform this new command.

**(Word 9) Number of Send Repetitions:** Word 9 specifies the number of transfers to be performed before automatically completing the communications request and closing the channel. If this value is set to 1, only a single transfer will be issued. If this value is set to 0, transfers will be issued on the requested period until the channel is aborted.

**(Word 10) Time Unit for Send Period:** Words 10-11 together define how often the transfer is to be performed (*transfer period*). Word 10 specifies the time unit such as seconds or minutes for the send period. Word 11 specifies the number of those units. The choices for the time units are shown below.

Value	Meaning
1	hundredths of seconds (10ms)
2	tenths of seconds (100ms)
3	seconds
4	minutes
5	hours

Word Offset	Dec	(Hex)	Description
Word 11	00007	(0007)	The minimum interval between host accesses (every 7 seconds)
Word 12	00050	(0032)	Timeout on each transfer response (500ms)
Word 13	00008	(0008)	Local PLC - Memory type from which to send data (%R)
Word 14	00050	(0032)	Local PLC - Starting address from which to send data (%R50)

### Further Considerations for Memory Locations 11-14

**(Word 11) Number of Time Units for Send Period:** Word 11 specifies the number of time units for the send period. The send period is in effect even when the Channel command is set up to issue a single send. *A Channel command set up to issue a single send can have only one pending send transfer.*

**Example Send Period Calculation:** If Word 10 contains a value of 3 specifying seconds as the time unit and Word 11 contains a value of 20, the send period is 20 seconds.

A send is normally issued at the start of each send period. If the *pending* transfer has not completed during the send period, the Channel Error bit and Detailed Channel Status words are set to indicate a non-fatal period error. The pending transfer can still complete after the period error occurs. For Channel commands set up to issue multiple sends, the next transfer is issued only after the pending transfer completes.

If the Number of Time Units is zero, a subsequent transfer is issued as soon as the previous transfer completes. In this case, no period errors are reported by the Channel Error bit.

**(Word 12) Timeout for Each Send:** Word 12 specifies the time (in hundredths of a second) the Ethernet interface will wait for a send transfer to complete before setting the Channel Error bit and Detailed Channel Status bits to indicate a non-fatal timeout error. The transfer can still complete even after a timeout occurs. As a result, an application can choose what to do if one occurs. If the timeout value is specified as zero, no timeout errors will be reported.

For most applications, a timeout is not needed because the send period acts as a timeout. (Word 12 should be zero for no timeout.) However, there are two circumstances where a timeout is recommended:

- If number of time units (Word 11) is zero, so that a subsequent transfer is issued as soon as the previous transfer completes and no period errors are reported. In this case, a timeout value can be specified so that the Channel Error bit will report timeout errors.
- If the send period is very long (minutes or hours). In this case, a shorter timeout value can be specified so the application doesn't have to wait for the send period to expire before acting.

**(Word 13) Local PLC - Memory Type:** Words 13–14 specify the location in the local PLC where the Ethernet interface will get the data to be written to the remote SRTP server. Valid values for Word 13 are listed for Establish Read Channel.

**(Word 14) Local PLC - Memory Starting Address:** Word 14 determines the starting address in the local PLC from which the data is to be sent. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 13. This offset can be in bits, bytes, or words depending on the mode specified (for



Word Offset	Dec	(Hex)	Description
Word 15	00008	(0008)	Local PLC - Number of memory units (8 registers)
Word 16	00000	(0000)	Reserved
Word 17	00000	(0000)	Reserved
Word 18	00001	(0001)	Remote Network Address type (IP Address)
Word 19	00004	(0004)	Remote Network Address length in words (4)
Word 20	00010	(000A)	Remote Host - Register 1 of IP address (10)
Word 21	00000	(0000)	Remote Host - Register 2 of IP address (0)
Word 22	00000	(0000)	Remote Host - Register 3 of IP address (0)
Word 23	00001	(0001)	Remote Host - Register 4 of IP address (1)

### ***Further Considerations for Memory Locations 15-23***

**(Word 15) Local PLC - Number of Memry Units:** Word 15 specifies the amount of data to be transferred. The value entered is the number of memory units to be transferred, where the size of a memory unit is a bit, byte, or word as specified in Word 13. For example, if Word 13=16 and Word 15=4, then 4 bytes (32 bits) of %I memory will be transferred. A maximum of 16384 bits, 2048 bytes, or 1024 words of data can be specified.

**(Word 16) Reserved:** Word 16 is reserved and should contain the value zero.

**(Word 17) Reserved:** Word 17 is reserved and should contain the value zero.

**(Word 18) Remote Host - Network Address Type:** Word 18 specifies the format of the remote host's address. Word 18 must contain the value 1, which indicates a dotted-decimal IP address expressed using a separate register for each decimal digit.

**(Word 19) Remote Host - Network Address Length:** Word 19 specifies the length in words of the remote host's IP address. Word 19 must contain 4.

**(Words 20–23) Remote Host - IP Address:** Words 20–23 specify the four integers, one integer per word, of the dotted-decimal IP address of the remote host to be accessed.

### 8.3.5 Abort Channel (2001)

The Abort Channel command immediately disconnects an active channel from its remote PLC and closes the channel. The Channel Transfer bit, the Channel Error bit, and the Detailed Channel Status words for the channel are set to zero.

#### Example Command Block

Abort Channel 5. Return the COMMREQ Status word to %R10.

Word Offset	Dec	(Hex)	Description
Word 1	00002	(0002)	Length of Channel Command Data Block (2 words)
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	The memory type of COMMREQ status word (%R)
Word 4	00009	(0009)	COMMREQ status word address minus 1 (%R10) (0-based)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	02001	(07D1)	Abort Channel command number
Word 8	00005	(0005)	Channel number 5

#### *Further Considerations for Memory Locations 1-8*

**(Word 7) Channel Command Number:** This command parameter requests that a channel be aborted. If the command is processed successfully, it terminates processing on the channel by the time success is indicated in the COMMREQ status word.

**(Word 8) Channel Number:** The channel number specifies the channel to be disconnected (1–32). As a convenient way to abort all channels, if the channel number parameter is –1 (FFFFH), all channels in use are aborted. It is *not* an error to abort all channels if there are none in use. Neither is it an error to abort an idle channel.

**Note:** For the Abort Channel and Retrieve Detailed Channel Status commands, no actual data is transmitted on the network. Communication occurs between the client PLC CPU and the local Ethernet interface only. For these commands, the actual function is performed locally within the Ethernet interface and then the COMMREQ Status word is sent immediately to the CPU.

## 8.3.6 Retrieve Detailed Channel Status (2002)

The Retrieve Detailed Channel Status command requests that the *current* Detailed Channel Status words are returned for a channel. The Detailed Channel Status words contain an active/inactive channel indicator and the last channel error codes seen. These two words of detailed status supplement the information available in the COMMREQ Status word and the Channel Status bits. The command does not affect the value of the Channel Status bits.

The Detailed Channel Status words are updated every time the status of the channel changes. If the channel is operating with a fast repetition period, the status words may change faster than the ladder executes the COMMREQ to retrieve them. If that happens, some status values could be missed by the application program.

### Example Command Block

Retrieve detailed channel status for Channel 5. Store the Detailed Channel Status words to Registers %R100-%R101. Return the COMMREQ status word to %R10.

Word Offset	Dec	(Hex)	Description
Word 1	00004	(0004)	Length of Channel Command Data Block (4 words)
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory Type of COMMREQ status word (%R)
Word 4	00009	(0009)	COMMREQ status word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	02002	(07D2)	Retrieve Detailed Channel Status Command number
Word 8	00005	(0005)	Channel number 5
Word 9	00008	(0008)	Local PLC - Memory type to store Detailed Chan. Stat. (%R)
Word 10	00100	(0064)	Local PLC - Starting address (%R100)

#### *Further Considerations for Memory Locations 1-10*

**(Word 7) Channel Command Number:** Requests that Detailed Channel Status words be returned. The Detailed Channel Status words are written to the location specified in Words 9 and 10. The COMMREQ status word indicates successful completion of the command. If the specified channel is not currently in use, the latest status is returned.

**(Word 8) Channel Number:** Specifies the channel (1 – 32) whose status is to be read.

**(Word 9) Local PLC - Memory Type:** Words 9 and 10 specify the starting point in the client CPU memory where the Detailed Channel Status words are to be written. The length of the transfer is always 2 words.

**(Word 10) Local PLC - Memory Starting Address:** Determines the starting address to store the Detailed Channel Status data. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 9. This offset is in bits, bytes, or words depending on the mode specified (for example, if Word 9=16 and Word 10=2, then the starting address will be %I9). Valid ranges of values depend on the PLC's memory ranges. Make sure this area can contain the 2 words of data without overwriting other application data.

**Note:** For the Abort Channel and Retrieve Detailed Channel Status commands, no actual data is transmitted on the network. Communication occurs between the client CPU and the local Ethernet interface only. For these commands, known as “local” commands, the function is performed locally within the Ethernet interface and then the COMMREQ Status word is sent immediately to the CPU.

## Monitoring the Detailed Channel Status Words

The Detailed Channel Status words (DCS words) are returned from the Ethernet interface to the CPU in response to a Retrieve Detailed Channel Status command from the application program. The first two Detailed Channel Status bytes report status and errors in the same format as the COMMREQ Status word. See the list of error codes in Section 12, *Diagnostics*.

The second word of the DCS words indicates when the channel is active.

If a channel error is indicated (by the Channel Error bit) after the channel is established, the first word of the DCS words contains an error code indicating the cause of the error. The second word of the DCS words indicates whether the channel is active or idle.

The Detailed Channel Status words are updated in the Ethernet interface every time the status of the channel changes. If the channel is operating with a fast repetition period, the status words may change faster than the ladder executes the COMMREQ to retrieve them. Therefore, some status values may be missed by the program logic.

The DCS word location is specified in the Retrieve Detailed Channel Status Command. The contents of these status words are defined below.

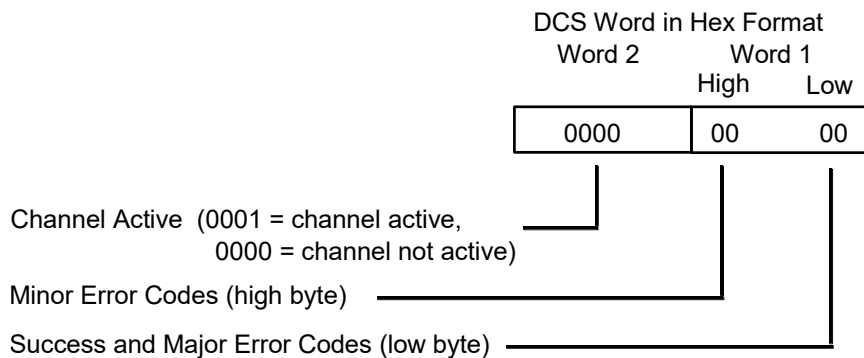
The initial value of the Detailed Channel Status words is all zeroes. DCS words are reset to zero when:

- The Ethernet interface is powered up or restarted
- The CPU transitions from STOP to RUN
- A channel abort COMMREQ aborts the channel

### Format of the Detailed Channel Status Words (DCS Words)

Display the DCS status words in the hexadecimal form to differentiate the high and low bytes.

**Figure 54: Interpreting Detailed Channel Status Words**



## 8.4 Programming for Channel Commands

The COMMREQ function for a Channel command must be initiated by a one-shot. That will prevent the COMMREQ from being executed each CPU scan, which would overrun the capability of the Ethernet interface and possibly require a manual restart. Checking certain status bits before initiating a COMMREQ function is also important. In particular, the LAN interface OK bit should be used as an interlock to prevent the execution of the COMMREQ when the Ethernet interface is not operational. After initiating a COMMREQ on a channel, no further COMMREQs should be issued to that channel until a non-zero COMMREQ status word has been returned to the program from the Ethernet interface.

Every ladder program should do the following before initiating a COMMREQ function.

1. Initiate the COMMREQ function with a one-shot. This prevents sending the same COMMREQ Command Block more than once.
2. Include at least the LAN interface OK bit in the LAN interface Status Word as an interlocking contact for the COMMREQ function.
3. Zero the word location you specify for the COMMREQ status word and FT Outputs of the COMMREQ function block before the COMMREQ function is initiated.
4. Move the command code and parameters for the Channel command into the memory location specified in the IN input of the COMMREQ Function Block before the COMMREQ function is initiated.

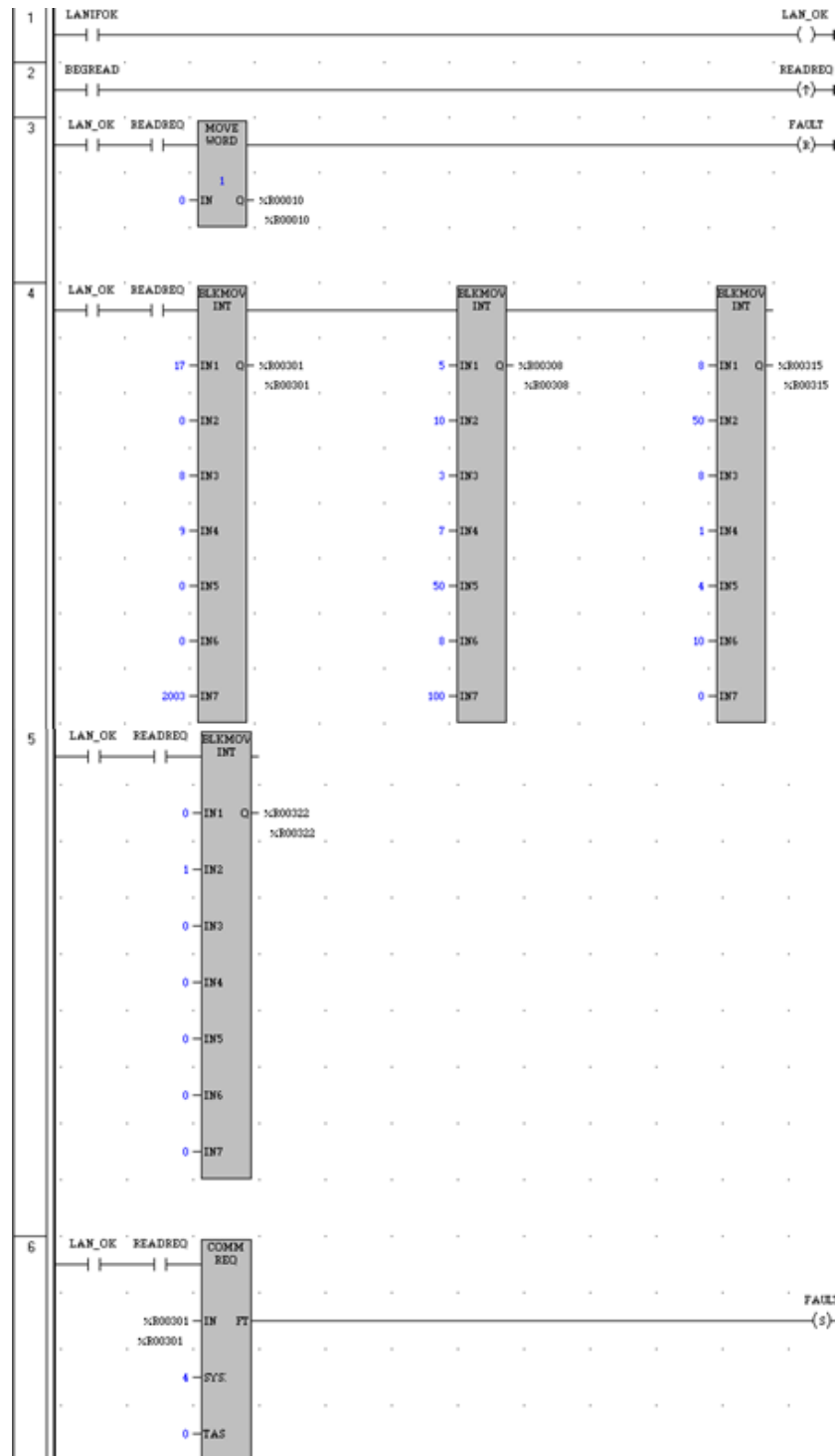
An example ladder program segment on the next page illustrates these points.

### 8.4.1 COMMREQ Sample Logic

In the sample logic that follows, the input values for the Block Move Functions are taken from the command *Example Command Block* in this chapter.

Nicknames are used in this example to make the ladder program easier to follow. LANIFOK is bit 16 of the LAN interface Status bits. All other nicknames can be assigned as needed.

Figure 55: Sample Ladder Logic for COMMREQ



**Rung # 1:** Input LANIFOK (bit 16 of the LAN interface Status bits) monitors the health of the Ethernet interface. If it is OK to send a COMMREQ, the LAN\_OK coil is ON. LAN\_OK is used as an interlock for Rungs 3–6.

**Rung # 2:** Input BEGREAD triggers READREQ, which enables execution of the MOVE and COMMREQ functions. READREQ is a one-shot (Positive Transition) coil, activating once when BEGREAD transitions from OFF to ON.

**Rung # 3:** The MOVE WORD function moves a zero to the COMMREQ status word referenced in the Command Block (see rung #4). This clears the COMMREQ status word. This rung also resets the FT output coil of the COMMREQ Function Block in rung #6.

It is vital that the COMMREQ status word be cleared and the COMMREQ fault output coil be cleared each time before initiating a COMMREQ function.

**Rungs # 4–5:** The BLKMOV INT functions to set up the COMMREQ Command Block contents. When these rungs are activated, the constant operands are moved into the memory beginning at the address indicated in the instruction. The constant operands in this example are defined in the Establish Read Channel Example in this chapter.

**Rung # 6:** The COMMREQ Function Block.

- The IN field points to the starting location of the Command Block parameters (%R00301 in this example).
- The SYSID field of the COMMREQ function block defines the rack and slot of the Ethernet interface to receive the command data. This is a hexadecimal word value that gives the rack (high byte) and slot (low byte) location of the Ethernet interface module. In the example ladder diagram shown, the first three number places (from left to right) are zeroes and are not displayed; only the last number, 4, appears. This indicates rack 0, slot 4.
- The TASK field of the COMMREQ function block indicates which mailbox task ID to use for the specified rack and slot. For a PACSystems RX3i Ethernet module, the Task must be set to 0. For a PACSystems CPU embedded Ethernet interface, the Task must be set to 65536 (10000H).
- The FT output (energizes the FAULT coil in this example) is turned ON (set to 1) if problems were preventing the delivery of the Command Block to the Ethernet interface. In this case, the other status indicators are not updated for this COMMREQ

## 8.4.2 Sequencing Communications Requests

If the Ethernet interface receives Command Blocks from the PLC CPU faster than the interface can process them, the interface will log an exception event 08, Entry 2=0024H and will log the PLC Fault Table entry:

“Backplane Communications with PLC Fault; Lost Request”

Only one COMMREQ function per channel can be pending at one time. A COMMREQ function is pending from the time it is initiated in the ladder program until its COMMREQ status word has been updated to a non-zero value by the Ethernet interface.

If the PLC CPU attempts to send COMMREQs to the Ethernet interface faster than the Ethernet interface can receive them, the FT output of the COMMREQ function block will be set and the CPU will generate the following entry in the PLC Fault Table:

“Mailbox queue full – COMMREQ aborted”

The PLC logic program should send retry the COMMREQ after a short delay.

## 8.4.3 Managing Channels and TCP Connections

When you issue a COMMREQ to establish a read or write channel, a TCP connection is created, the transfer(s) are made, then upon completion of all the transfers, the TCP connection is terminated. It takes time to create and to terminate these connections. If an application is constructed so that it rapidly and repeatedly establishes a channel with only one repetition (one transfer), the available TCP connections for the Ethernet interface may be totally consumed. A “snapshot” of the state of the TCP connections would show some of them being created, some being terminated, and some active, but none available.

### CAUTION

In Certain Conditions TCP Connections Can Be Totally Consumed

If the logic for issuing COMMREQs is constructed so it does the following, all available TCP connections can quickly be used up:

- The number of repetitions (Word 9 in an Establish Read or Write Channel COMMREQ) is set to 1, *and*
- A new COMMREQ is issued repeatedly and immediately upon completion of the prior one.

## 8.4.4 Use “Channel Re-Tasking” To Avoid Using Up TCP Connections

TCP connections can be used up if each successive COMMREQ is directed to the same target device (same IP address). In this case, it is better to establish a channel with the target device once, leave it active, then re-task the channel, even if data transfers take place infrequently. This method will use only one TCP connection.

An additional advantage of re-tasking is that the time and network traffic required to create a channel and its associated TCP connection are not incurred each time a data transfer is required.

The disadvantages of re-tasking are:

- While the TCP connection is open, it is unavailable to the rest of your application, and
- The active TCP connection uses up network bandwidth because the active TCP connection generates a small amount of ongoing periodic network traffic.

### How to Re-Task a Channel

1. For Establish Read/Write Channel Commands, set the number of repetitions (COMMREQ Word 9) to 2 and set the read/write period (COMMREQ Words 10 and 11) to be longer than the expected time between transfers. For example, if you expect to transfer data about once per minute, set the read/write period to about two minutes. This will cause a TCP connection to be created and held open for two minutes.
2. Set up the ladder program to:
  - a. Issue the first COMMREQ and wait for the first transfer to complete, which will be indicated when the COMMREQ Status (CRS) word is changed to 1.
  - b. Then before the read/write period expires (at which time the second and final transfer is sent and the TCP connection is dropped), issue the next COMMREQ with the same parameters as specified in step 1. This will “re-task” the channel to use the existing



TCP connection instead of opening a new one, and will send another data transfer restarting the timer for the read/write period. Repeat step 2b for each successive data transfer desired.

## 8.4.5 Client Channels TCP Resource Management

There is a period of time that the OS Network stack hangs on to the TCP resources associated with a connection after it is closed. It applies to the initiator of the close, which is almost always the client-side. This time is referred to as the “TCP Linger Period”. Once the TCP Linger Period expires (60 seconds in the current OS implementation), the TCP resources are released. Application developers using client channels need to be aware of this behavior when designing their logic. There are a finite number of TCP resources allocated to client channels, and if channel connections are brought up and down so fast that these resources are depleted, then the application may have to wait until a TCP resource frees up to establish another client channel (a COMMREQ Status of 0xA890 is returned if no TCP resources are currently available; the application should wait and retry again).

SRTP Client Channels provides features that help the user preserve TCP connections. These include a period time where one can establish an SRTP Channel and specify the channel to run at a given interval, or run as fast as possible. One can also specify the number of iterations, or run forever. Additionally, SRTP Channels allow channel re-tasking of an active channel to the same remote device, where the parameters of an active channel, such as changing the channel command type (Read/Write), number of repetitions, periods, local memory address, remote memory address, etc. can be changed. SRTP Channels also allows channel re-tasking of an active channel to a different remote device (changing the remote device’s IP address, etc.). However, re-tasking to a different remote device will neither conserve TCP connections nor save on the time it takes to create a channel.

## 8.4.6 SRTP Application Timeouts

The application timeouts within SRTP Channels also include the time needed to establish and maintain the underlying network and SRTP connection. Examples are establishing the TCP connection for a new channel, establishing communication with the remote device, and TCP retransmissions during Channel operations. If the time needed for TCP connection establishment or maintenance exceeds the user-specified channel application timeout values, an application timeout will occur. Channel application timeouts are temporary errors; the channel continues to run when the expected response is received.

If the application is seeing timeouts during channel startup, there are a few different options:

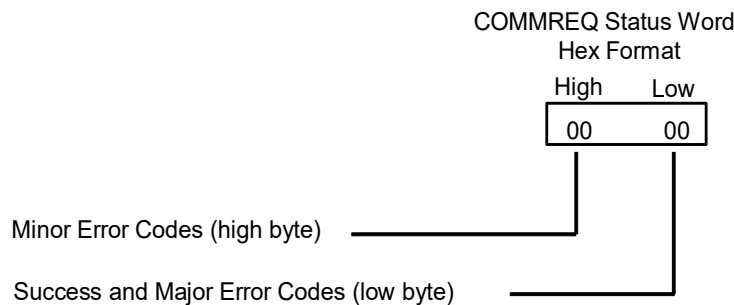
1. Increase timeout value to account for Channel connection overhead
2. Ignore the timeout error on the first transfer
3. Use a two-step setup approach where the first COMMREQ has a timeout large enough to account for the connection overhead and then Re-Task the channel to the normal operating timeouts.

## 8.5 Monitoring Channel Status

The COMMREQ Status word is returned from the Ethernet interface to the PLC CPU immediately if the Command Block contains a syntax error or if the command is local. For remote commands with no syntax error, it is returned either after the channel is established successfully and the first transfer has been completed or if there is an error establishing the channel. The location of the COMMREQ status word is defined in the Command Block for the COMMREQ function.

### 8.5.1 Format of the COMMREQ Status Word

**Figure 56: Interpreting COMMREQ Status Word**



It is critical to monitor the COMMREQ status word for each COMMREQ function. Zero the associated COMMREQ status word before executing the COMMREQ function. When the COMMREQ status word becomes non-zero, the Ethernet interface has updated it.

If after executing a COMMREQ function, the COMMREQ status word is zero (0) and the FT Output is OFF, the Command Block has been sent to the Ethernet interface, but no status has been returned. If this condition persists, check the PLC Fault Table for information.

If the COMMREQ status word is updated to 1, the Command Block was processed successfully by the Ethernet interface.

If the COMMREQ status word is updated to a value other than 1, an error has occurred in processing the Command Block. The cause may be:

- Errors in the Command Block (the Channel command code or parameters), or
- For an establish command (Establish Read Channel, Establish Write Channel, or Send Information Report), the command parameters were valid but there was an error in establishing a channel

Section 12 lists the Major and Minor error codes that may be returned in the COMMREQ status words. Do not use data received from a server until the COMMREQ status word for that channel is 1 or the Data Transfer bit goes to 1.

## Features and Benefits of the PACSystems SRTP Channels

This section lists differences between the Series 90 implementation of SRTP Channels and the PACSystems implementation:

	PACSystems
TCP Connection Timeout	<ul style="list-style-type: none"> <li>- Default 75 Seconds.</li> <li>- Override with AUP setting: “SRTP Channel TCP Connect Timeout”</li> <li>- (Minimum value 10 ms)</li> </ul>
TCP Linger Period	60 Seconds
SRTP Channel Implementation	<ul style="list-style-type: none"> <li>- Run-to-Stop transition causes an Abrupt Shutdown</li> <li>- Avoid Linger Period and reduces the change of exhausting TCP resources when transitioning between Run-&gt;Stop and Stop-&gt;Run.</li> </ul>
Abort/Abort All Channel COMMREQ Issuance	If COMMREQ Status Word is not updated before the A/AA COMMREQ command is issued, Status Word is set to failure value (A990) to indicate “discard while aborting in progress.”
COMMREQ Status Codes	New COMMREQ Status Codes Defined
Re-tasking	Implementation support re-tasking to a different remote device (different IP address)
Total Number of TCP Connections Shared	<ul style="list-style-type: none"> <li>- 48 Max Server TCP Connection</li> <li>- 32 Max Client Channel TCP Connections</li> </ul>



## Section 9 Modbus/TCP Server

This section describes the implementation of the Modbus/TCP Server feature for the PACSystems family of products.

- Modbus/TCP Server
- Reference Mapping
- Modbus Function Codes

## 9.1 Model Comparison for Modbus/TCP Server Capabilities

PACSystems Ethernet interfaces are enabled with Modbus/TCP server/client capabilities to communicate with MODBUS devices. The following table shows which RX3i Ethernet Interfaces support Modbus/TCP server capabilities.

RX3i Embedded Ethernet Interfaces	RSTi-EP Ethernet Interfaces
CPE302	CPE100
CPE305	CPE115
CPE310	<b>RX3i Ethernet Module</b>
CPE330	ETM001-Jx
CPE400	ETM-Kxxx
CPL410	

The values in the following table specify the maximum number of simultaneous Modbus/TCP Server connections. These connections are not shared with any other applications. Other TCP-based application protocols such as SRTP Server use a different set of TCP connections.

Modbus/TCP Server Capabilities for RX3i Ethernet Capabilities										
	CPE302	CPE305	CPE310	CPE330	CPE400	CPL410	CPE100	CPE115	ETM001 -Jx	ETM001 -Kxxx
<b>Simultaneous Modbus/TCP Server Connections</b>	16	16	16	16	16	16	8	8	16	16

## 9.2 Modbus Conformance Classes

PACSystems Modbus/TCP Server supports Modbus Conformance classes 0, 1, and 2.

The RX3i Ethernet module has been certified by the Modbus/TCP Conformance Test Laboratory to be in conformance with *Conformance Test Policy* Version 2.1.

### 9.2.1 Server Protocol Services

The Modbus/TCP Server responds to incoming Request Connection, Terminate Connection, and Request Service messages. The client with which the server is interacting should wait for the server's response before issuing the next Request Service, otherwise, requests could be lost.

There is no inactivity timeout in the server. If a client opens a connection, that connection stays open until the connection is terminated.

### 9.2.2 Station Manager Support

The Modbus/TCP Server supports the standard Station Manager commands: STAT and TALLY. The Modbus/TCP Server task letter is "o".

## 9.3 Reference Mapping

The Modbus protocol's reference table definition is different from the internal structure of the PACSystems reference tables. Modbus refers to Holding Register, Input Register, Input Discrete, and Coil tables; PACSystems uses Discrete Input (%I), Discrete Output (%Q), Analog Input (%AI), Register (%R), and Word (%W) reference tables for Modbus data. The following table shows how each Modbus table is mapped to the PACSystems reference tables.

### 9.3.1 Modbus Reference Tables

Modbus File Access (6xxxx)	Modbus Holding Register Table (4xxxx)	Modbus Input Register Table (3xxxx)	Modbus Input Discrete Table (1xxxx)	Modbus Coil Table (0xxxx)	PACSystems Reference Tables
---	---	---	1 – 32768 (bits) <sup>25</sup>	---	%I1 – 32768 (bits)
---	---	1 – 32640 (16-bit words)	---	---	%AI1 – 32640 (16-bit words)
---	---	---	---	1 – 32768 (bits) <sup>26</sup>	%Q1 – 32768 (bits)
---	1 – 32640 (16-bit words)	---	---	---	%R1 – 32640 (16-bit words)
F1,R1 – F525,R2880 (16-bit words)	---	---	---	---	%W1 – 5,242,880 (16-bit words)

<sup>25</sup> CPE100/115 supports 1-2048 Modbus input discrete bits.

<sup>26</sup> CPE100/115 supports 1-2048 Modbus coil bits.

## Modbus File Access Table

The Modbus File Access table is mapped exclusively to PACSystems %W memory.

### Applicable Functions

- Read File Record
- Write File Record

### Translating %W Reference Addresses

To find the PACSystems %W memory address equivalent of a Modbus File and Record:

$$\%W = 10,000 (F - 1) + R$$

To find the Modbus File and Record equivalent of a PACSystems %W memory address:

$$File = \left( \frac{W - 1}{10,000} \right) + 1$$

$$Record = W - (10,000 (F - 1))$$

*Discard any fractional portion; round the result down to the next integer value.*

Figure 57: Calculations for Modbus File and Record %W Memory Address

### CAUTION

If you use the Modbus function Write File Record and specify multiple record sections, the first N-1 sections will be written to the server's PLC reference memory, even if an error prevents the writing of the last section.



## Modbus Holding Register Table

The Modbus Holding Register table is mapped exclusively to the CPU Register (%R) table.

### Applicable Functions

- Read Multiple Registers
- Write Multiple Registers
- Write Single Register
- Mask Write Register
- Read/Write Multiple Registers

## Modbus Input Register Table

- The Modbus Input Register table is mapped exclusively to the CPU Analog Input (%AI) table.

### Applicable Functions

- Read Input Registers

## Modbus Input Discrete Table

- The Modbus Input Discrete table is mapped exclusively to the CPU Discrete Input (%I) table.

### Applicable Functions

- Read Input Discretes

## Modbus Coil Table

- The Modbus Coil table is mapped exclusively to the CPU Discrete Output (%Q) table.

### Applicable Functions

- Read Coils
- Write Coils
- Write Single Coil

## 9.3.2 Address Configuration

- Address mapping is done in the *PME Hardware Configuration* of the CPU. All Ethernet modules and daughter-boards in the PLC use Modbus-to-PLC address mapping based on this one map. The Modbus/TCP Server does not use COMMREQs to configure address mapping.
- Each PLC memory area is mapped to an appropriate Modbus address space. On the Settings tab, Modbus Address Space Mapping can be set to Standard Modbus Addressing or Disabled. If Modbus Address Space Mapping is set to Standard, the Modbus/TCP Address Map tab displays the standard reference assignments.

Number	Modbus Register	Start Address	End Address	PLC Memory Address	Length
1	0xxx – Coil Table	1	32768	%Q00001	32768
2	1xxx - Discrete Table	1	32768	%I00001	32768
3	3xxx - Input Registers	1	64	%AI00001	64
4	4xxx – Register Table	1	1024	%R00001	1024
5	6yxxx – Internal Table	0	0	%W0001	0

**Note:** The maximum bit length for the CPE100/115 is 2048.

When Modbus Address Space Mapping is set to Disabled on the Settings tab, the Modbus/TCP Address Map tab does not appear.

If the CPU module does not receive an address map from PME, Ethernet interfaces within the PLC will respond to Modbus/TCP clients with Exception Code 4, Slave Device Failure. This same exception code will also be returned when the PLC's *Hardware Configuration* is cleared.

## 9.4 Modbus Function Codes

This section summarizes the mapping of PACSystems reference tables to Modbus addresses by the Modbus function codes supported by the Modbus/TCP Server. The mapping shown in this table assumes that the PLC is configured to use its default reference table sizes.

Modbus Function Code	Modbus			PLC	
	Table	Start Address	Length	Start Address	Length
1 Read Coils	0xxx	1	32768	%Q00001	32768
5 Write Single Coil					
15 Write Multiple Coils					
2 Read Discrete Inputs	1xxx	1	32768	%I00001	32768
3 Read Holding Registers	4xxx	1	1024	%R00001	1024
6 Write Single Register					
16 Write Multiple Registers					
22 Mask Write Register					
23 Read/Write Multiple Registers					
4 Read Input Registers	3xxx	1	64	%AI00001	64
7 Read Exception Status	N/A	N/A	N/A	N/A	N/A
8 Diagnostics					
20 Read File Record	6yxxx	1	0	%W00001	0
21 Write File Record					

**Note:** The maximum bit length for the CPE100/115 is 2048.



# Section 10 Modbus/TCP Client

This chapter explains how to program communications over the Ethernet network using Modbus/TCP Channel commands. This chapter applies only to PLCs being used as client PLCs to initiate Modbus/TCP communications.

- The Communications Request
- The COMMREQ Function Block and Command Block
- Modbus/TCP Channel Commands
- Status Data
- Controlling Communications in the Ladder Program
- PACSystems Modbus/TCP Channels

## 10.1 Model Comparison for Modbus/TCP Client Capabilities

PACSystems Ethernet interfaces can enable PLCs to be used as client PLCs to initiate Modbus/TCP communications. The values in the table below specify the maximum number of clients permitted. Each may be SRTP or Modbus/TCP.

	CPE302	CPE305	CPE310	CPE330	CPE400	CPL410	CPE100	CPE115	ETM001-jx	ETM001-Kxxx
Clients permitted (SRTP Client or Modbus/TCP Client)	16	16	16	32	32	32	8	8	32	32

## 10.2 Communications Request

“Communications Request” is a term used to describe all the user elements required for correctly initiating Channel commands in the client. No programming of Communications Requests is required for devices acting as servers.

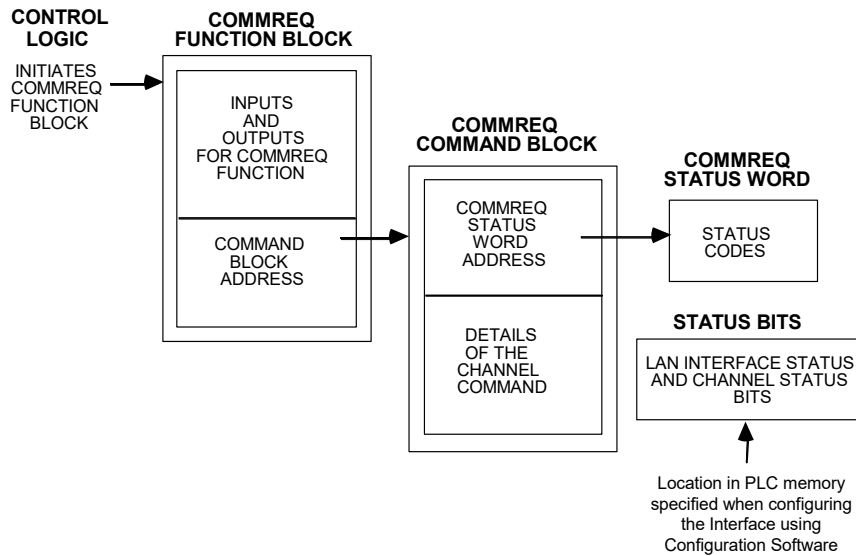
### 10.2.1 Structure of the Communications Request

The Communications Request is made up of the following elements:

- The COMMREQ Function Block (ladder instruction)
- The COMMREQ Command Block
- The Channel Command
- Status Data (COMMREQ Status word, LAN interface Status and Channel Status bits)
- The logic program controlling the execution of the COMMREQ Function Block

The figure below illustrates the relationship of these elements:

Figure 58: Phases of a COMMREQ Execution



## 10.2.2 COMMREQ Function Block

The COMMREQ Function Block is the ladder instruction that triggers the execution of the Channel command. In the COMMREQ Function Block, you specify the rack and slot location of the Ethernet interface, a task value, and the address of a location in memory that contains the Command Block. There is also a fault output on the COMMREQ Function Block that indicates certain programming errors.

## 10.2.3 COMMREQ Command Block

The COMMREQ Command Block is a structure that contains information about the Channel command to be executed. The Command Block consists of two parts:

**Common Area** - includes the address of the COMMREQ Status word (CRS word).

**Data Block Area** - describes the Channel command to be executed.

When the COMMREQ function is initiated, the Command Block is transferred to the Ethernet interface for action.

## 10.2.4 Modbus/TCP Channel Commands

The Channel commands are a set of client commands used to communicate with a server. Up to 32 channels can be established. The channel number is specified in the Command Block for the Channel command. The channel can be monitored using the Channel Status bits. The 32 Client connections of an Ethernet interface are shared between all Client protocols. For example, if 16 Client connections are used for SRTP Channels, there are 16 Client connections available for Modbus/TCP Channels. Any given channel can be assigned to only one protocol at a time.

## 10.2.5 Status Data

There are several types of status available to the client application program.

**LAN Interface Status Bits (LIS Bits):** The LIS bits comprise bits 1–16 of the 80-bit status area. The location of this 80-bit status area is assigned using the configuration software. The LIS bits contain information on the status of the Local Area Network (LAN) and the Ethernet interface.

**Channel Status Bits:** The Channel Status bits comprise bits 17–80 (64 bits) of the 80-bit status area. When used for Modbus/TCP channels, these bits consist of a *connection open* bit and an unused bit, reserved for future use, for each of the 16 channels that can be established. Status bits for unused channels are always set to zero.

**COMMREQ Status Word (CRS Word):** The 16-bit CRS word will receive the initial status of the communication request. The location of the CRS word is assigned for each COMMREQ function in the COMMREQ Command Block.

**FT Output of the COMMREQ Function Block:** This output indicates that the PLC CPU detected errors in the COMMREQ Function Block and/or Command Block and did not pass the Command Block to the Ethernet interface.

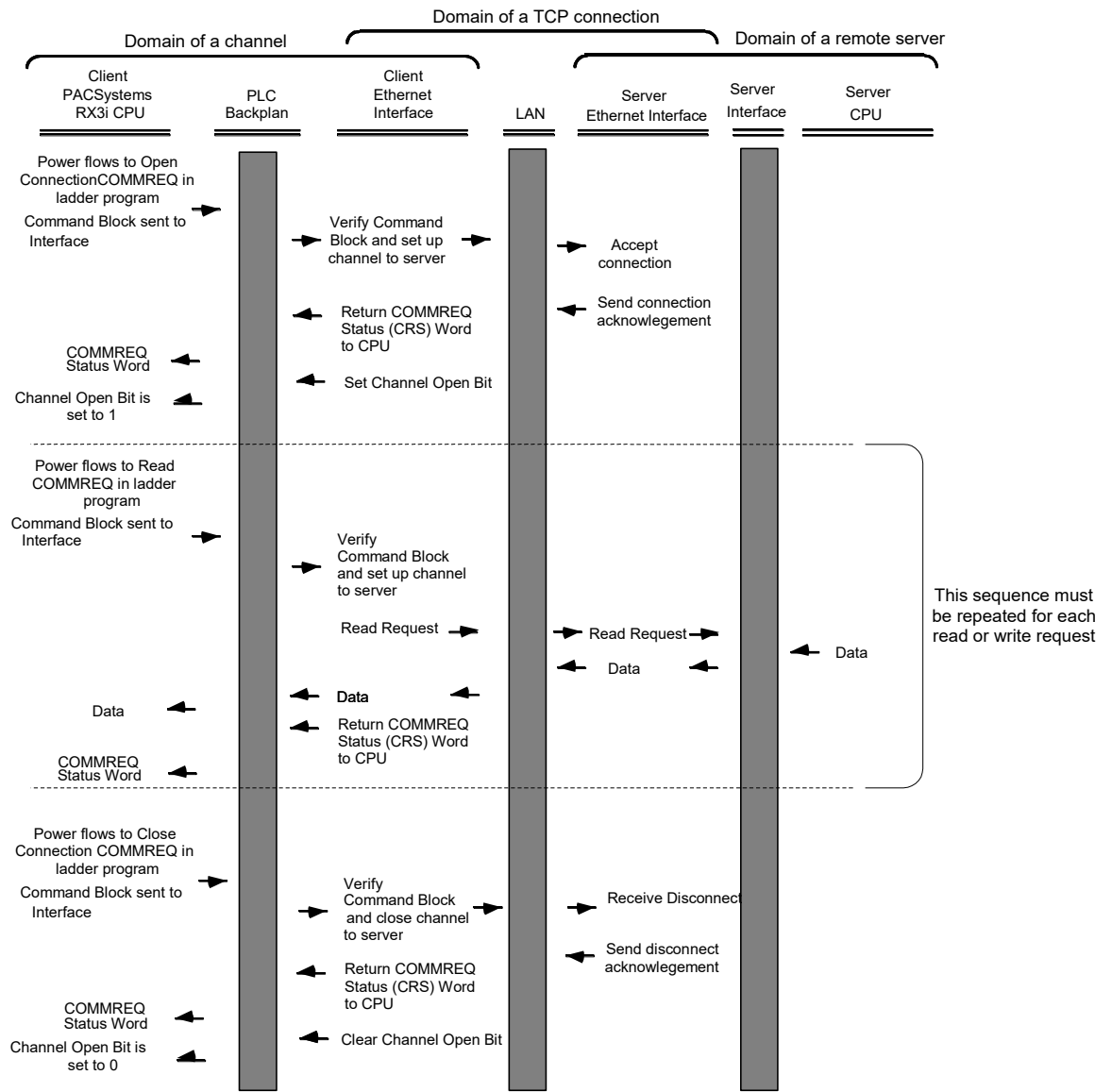
### The Logic Program Controlling Execution of the COMMREQ Function Block

The COMMREQ must be initiated by a one-shot to prevent the COMMREQ from being executed repeatedly each CPU scan, which would overrun the capability of the Ethernet interface and possibly require a manual restart. Checking certain status bits before initiating a COMMREQ function is also important. In particular, the LAN interface OK bit should be used as an interlock to prevent the execution of the COMMREQ function when the Ethernet interface is not operational. Following the initiation of a COMMREQ on a channel, no further COMMREQs should be issued to that channel until a non-zero CRS word has been returned to the program from the Ethernet interface.

## 10.2.6 Operation of the Communications Request

Figure 59 shows how Communications Requests are executed to complete a data read from the remote Modbus/TCP device. The figure specifically illustrates the successful operation of a data read.

**Figure 59: Illustration of Phased Operation of a COMMREQ**



1. A Communications Request begins when there is power flow to a COMMREQ function in the client. The Command Block data is sent from the CPU to the Ethernet interface.
2. The COMMREQ Status word (CRS word) is returned immediately if the Command Block is invalid. If the syntax is correct, then the CRS word is returned after the transfer of data.



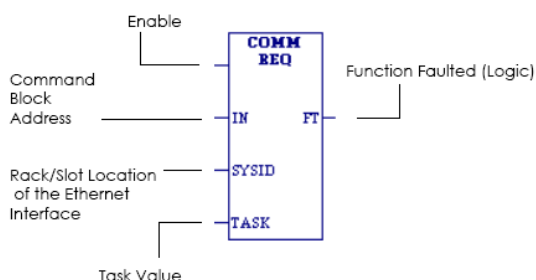
## 10.3 COMMREQ Function Block and Command Block

This section describes the programming structures common to all Communications Requests: the COMMREQ Function Block and the Command Block.

### 10.3.1 The COMMREQ Function Block

The Communications Request is triggered when the logic program passes power to the COMMREQ Function Block.

**Figure 60: The COMMREQ Function Block**



Each of the inputs and outputs is discussed in detail below. It is important to understand that the Command Block address points to the location in memory you have set up as the Command Block.

**Enable:** Control logic for activating the COMMREQ Function Block. See Section 5 for tips on developing your program.

**IN:** The location of the Command Block. It can be any valid address within a word-oriented area of memory (%R, %AI, %AQ, %P, %L, or %W for the Ethernet interface).

**SYSID:** A hexadecimal word value that gives the rack (high byte) and slot (low byte) location of the Ethernet interface. For the PACSystems embedded Ethernet interface, enter the rack/slot location of the module.

Rack	Slot	Hex Word Value	Notes
0	0	16#0000	RXi, CPE400/CPL410 and CPE100/CPE115
0	1	16#0001	
0	4	16#0004	
3	4	16#0304	
2	10	16#020A	
4	2	16#0402	

**TASK:** For the RX3i ETM001-Jx and ETM001-Kxxx Ethernet interfaces TASK must always be set to zero. For PACSystems CPU embedded Ethernet interface, TASK must be set to 65536 (0x10000) to address the CPU's Ethernet embedded ethernet interfaces.

**FT Output:** The FT output is set if the PLC CPU (rather than the Ethernet interface) detects that the COMMREQ fails. In this case, the other status indicators are not updated for this COMMREQ.

## 10.3.2 The COMMREQ Command Block

When the COMMREQ function is initiated, the Command Block is sent from the PLC CPU to the Ethernet interface. The Command Block contains the details of a command to be performed by the interface.

The address in CPU memory of the Command Block is specified by the IN input of the COMMREQ Function Block. This address can be any valid address within a word-oriented area of memory. The Command Block is usually set up using either the BLOCK MOVE or the DATA INIT COMM programming instruction. The Command Block has the following structure:

Word Offset	Description
Word 1	Data Block Length (words)
Word 2	WAIT/NOWAIT Flag
Word 3	CRS Word Memory Type
Word 4	CRS Word Address Offset
Word 5	Reserved
Word 6	Reserved
Words 7 and up	Data Block (Channel Command Details)

### ***Further Considerations for Memory Locations 1-7***

**(Word 1) Data Block Length:** This is the length in words of the Data Block portion of the Command Block. The Data Block portion starts at Word 7 of the Command Block. The length is measured from the beginning of the Data Block at Word 7, not from the beginning of the Command Block. The correct value for each command, and the associated length of each command, is specified in the next section.

**(Word 2) WAIT/NOWAIT Flag:** This flag must be set to zero for TCP/IP Ethernet Communications.

*COMMREQ Status Word:* The Ethernet interface updates the CRS word to show success or failure of the command. Command words 3 and 4 specify the PLC memory location of the CRSW word.

**(Word 3) COMMREQ Status Word Memory Type:** This word specifies the memory type for the CRS word. For more information, refer to *COMMREQ Memory Types Available* in this section.

**(Word 4) COMMREQ Status Word Address Offset:** This word contains the offset within the memory type selected. *The status word address offset is a zero-based number.* For example, if you want %R1 as the location of the CRS word, you must specify a zero for the offset. The offset for %R100 would be 99 decimal. Note that this is the only zero-based field in the Channel commands.

**(Word 5):** Reserved. Set to zero.

**(Word 6):** Reserved. Set to zero.

**(Words 7 and up) Data Block:** The Data Block defines the Channel command to be performed. For information on how to fill in the Channel command information, see the next section.

## COMMREQ Memory Types Available

The memory types are listed in the table below:

Type	Value (Decimal)	Value (Hex.)	Description
%R	8	08H	Register memory (word mode)
%AI	10	0AH	Analog input memory (word mode)
%AQ	12	0CH	Analog output memory (word mode)
%I	16	10H	Discrete input memory (byte mode)
	70	46H	Discrete input memory (bit mode)
%Q	18	12H	Discrete output memory (byte mode)
	72	48H	Discrete output memory (bit mode)
%T	20	14H	Discrete temporary memory (byte mode)
	74	4AH	Discrete temporary memory (bit mode)
%M	22	16H	Discrete momentary internal memory (byte mode)
	76	4CH	Discrete momentary internal memory (bit mode)
%G	56	38H	Discrete global data table (byte mode)
	86	56H	Discrete global data table (bit mode)
%W	196	C4H	Word memory (word mode limited to %W1 - %W65536)

## 10.4 Modbus/TCP Channel Commands

This section describes the operation of the Channel commands. A detailed description and example of each Channel command are included. There are four Channel commands:

- Open a Modbus/TCP Connection
- Close a Modbus/TCP Connection
- Read Data from a Modbus Server Device to the PLC
- Write Data from the PLC to a Modbus Server Device
- Mask Write Register Request to a Modbus Server Device
- Read/Write Multiple Registers between PLC memory and a Modbus Server Device

Please note that Modbus/TCP channel COMMREQs (unlike SRTP channel COMMREQs) do not contain a parameter to configure a timeout value. Enforcing a timeout for a Modbus channel command is at the discretion of the user and must be implemented in the user application.

### 10.4.1 Open a Modbus/TCP Client Connection (3000)

The Modbus/TCP Ethernet interface transfers data to or from another Modbus/TCP device using a *channel*. Up to 32 channels are available for Modbus/TCP client communications. However, these 32 channels are shared with SRTP Channels so that the combination of SRTP Channels and Modbus/TCP Channels cannot exceed 32.

The Open Modbus/TCP COMMREQ requests the communication subsystem to associate a channel with a remote Modbus/TCP device. Using the COMMREQs defined later in this document the PLC may transfer data to and from a remote device.

Once a channel is allocated for Modbus/TCP Client communications, the channel remains allocated (i.e. another protocol such as SRTP Channels cannot use the channel). The channel connection is released only when: the application program closes the channel, the channel is automatically closed when the PLC transitions to STOP when the Ethernet interface uses a Redundant IP and the CPU transitions from the Active to the Backup unit, the Ethernet interface is reset or the underlying TCP connection is terminated.

The IP address of the remote Modbus/TCP device is specified in the Open Modbus/TCP COMMREQ using the standard dotted-decimal format. No other IP address format is accepted.

The COMMREQ Status Word (CRS) indicates the success or failure of the Open Modbus/TCP Client Connection COMMREQ. If the COMMREQ requests an invalid channel number or an already allocated channel the COMMREQ fails and the CRS is set to a non-zero value to identify the failure. See 10.5, *Status Data* for detailed CRS failure codes.

## Command 3000 Example

Establish a channel (Channel 5) to a remote Modbus/TCP device at IP address 10.0.0.1. Return the COMMREQ Status word to %R10.

Word Offset	Dec	(Hex)	Description
Word 1	00008	(0008)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03000	(0BB8)	Open Modbus/TCP Client Connection
Word 8	00005	(0005)	Channel number (5)
Word 9	00001	(0001)	Remote Device Address Type
Word 10	00004	(0004)	Length of Remote Device Address
Word 11	00010	(0010)	Numeric value of 1 <sup>st</sup> Octet
Word 12	00000	(0000)	Numeric value of 2 <sup>nd</sup> Octet
Word 13	00000	(0000)	Numeric value of 3 <sup>rd</sup> Octet
Word 14	00001	(0001)	Numeric value of 4 <sup>th</sup> Octet

### *Further Considerations for Memory Locations 1-14*

**(Word 7) Channel Command Number:** Word 7 is the command id for an Open Modbus/TCP Client Connection COMMREQ. If successful a TCP connection with the specified device is allocated.

**(Word 8) Channel Number:** Word 8 specifies the channel number to allocate for the Modbus/TCP Client connection. Channels 1-32 can be used for Client communications.

**(Word 9) Address Type:** Word 9 specifies the type of IP Address specified for the remote device. A value of one (1) is required in this word.

**(Word 10) Length of IP Address:** Word 10 specifies the length of the IP Address. A value of four (4) is required in this word.

**(Word 11) IP Address 1<sup>st</sup> Octet:** Word 11 specifies the value of the first octet of the IP Address.

**(Word 12) IP Address 2<sup>nd</sup> Octet:** Word 12 specifies the value of the second octet of the IP Address.

**(Word 13) IP Address 3<sup>rd</sup> Octet:** Word 13 specifies the value of the third octet of the IP Address.

**(Word 14) IP Address 4<sup>th</sup> Octet:** Word 14 specifies the value of the fourth octet of the IP Address.

## 10.4.2 Close a Modbus/TCP Client Connection (3001)

The application program closes a Modbus/TCP Client Connection by issuing the Close Modbus/TCP Client Connection COMMREQ. The Close COMMREQ closes the underlying TCP connection and frees the channel for other communication tasks.

An error response is returned if the channel number in the COMMREQ identifies a non-Modbus/TCP Client connection or an inactive channel.

### Command 3001 Example

Terminate the Modbus/TCP Client connection established on Channel 5. Return the COMMREQ Status word to %R10.

Word Offset	Dec	(Hex)	Description
Word 1	00002	(0002)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03001	(0BB9)	Close Modbus/TCP Client Connection
Word 8	00005	(0005)	Channel number (5)

### *Further Considerations for Memory Locations 1-8*

**(Word 7) Channel Command Number:** Word 7 requests the Close channel service.

**(Word 8) Channel Command Number:** Word 8 identifies a channel previously opened with an Open Modbus/TCP Client Connection request. If a Close Modbus/TCP Client Connection is sent to a channel that is already closed, a success CRS value of 1 will be returned.

## 10.4.3 Read Data from a Modbus/TCP Device (3003)

The Read Data from a Modbus/TCP Device COMMREQ requests a data transfer from a Modbus/TCP device to the PLC. The Read Data COMMREQ must reference an active Modbus/TCP channel previously established with the Open Modbus/TCP Client Connection COMMREQ.

Registers, Coils, or Exception Status data may be read from the remote Modbus/TCP device. The Modbus Function Code specifies the data type. Valid Function Codes for the Read Data COMMREQ are presented in the following table.

Function Code	Description	Modbus Server Memory Region Accessed	Data Unit Size	Maximum Data Units
1	Read Coils	Internal Bits or Physical coils	Bit	2000
2	Read Input Discretes	Physical Discrete Inputs	Bit	2000
3	Read Multiple Registers	Internal Registers or Physical Output Registers	Register (16-bit Word)	125
4	Read Input Registers	Physical Input Registers	Register (16-bit Word)	125
7	Read Exception Status	Server Exception Memory	Byte	Not Applicable
24	Read FIFO Queue	Internal Registers or Physical Output Registers	Register (16-bit Word)	32

The table above describes the general Modbus server memory areas. The actual memory accessed is dependent on how the server maps the Modbus memory regions to the server's local memory.

An Address and Length specify the location of the data in the remote device and the number of data units to transfer. The Length is the number of Registers or Coils to transfer. Modbus Function Code 7, Read Exception Status does not require the address as the remote device retrieves the exception status from an internal location.

When transferring data between server bit or coil memory to PLC bit memory, only the number of bits specified is transferred. For example, if the COMMREQ requests to read 9 coils from the Remote Device and requests to put the data at %M00001 in the Local PLC (using a bit type memory type), %M00001 through %M00009 will be updated with the data from the Remote Device and %M00010 through %M00016 will be unaffected. However, if server bit or coil memory is transferred to PLC byte or word memory, the following rules apply:

1. Transferring discrete data from the Remote Device to Local PLC Word (16-bit) memory: If the number of requested coils is not a multiple of 16, the data is padded with 0s to a 16-bit boundary. For example, if the COMMREQ requests reading 17 coils from the Remote Device and requests to place this data at %R00010, %R00010 (all 16 bits) and bit 0 of %R00011 will be updated with values from the Remote Device and bits 1 through 15 of %R00011 will be set to 0.
2. Transferring discrete data from the Remote Device to Local PLC byte memory (using byte type memory type): If the number of requested coils is not on an 8-bit boundary, the data is padded with 0s to an 8-bit boundary. For example, if the COMMREQ requests 9 coils from the Remote Device and requests to place this data at %M00001, %M00001 through %M00009 will be updated with values from the Remote Device, and %M00010 through %M00016 will be set to 0.

Data returned from the remote device is stored in the PLC data area specified in the Read Modbus/TCP Device COMMREQ. Data can be stored in any of the PLC data areas. for the list of data areas and identification codes for the PLC. Note: the first item referred to in each data area is indexed at item 1 (not item 0).

The COMMREQ Status Word (CRS) indicates the success or failure of the Read Data COMMREQ. If the COMMREQ requests an invalid channel number or any other field is invalid the COMMREQ fails and the CRS is set to a non-zero value to identify the failure.

## Command 3003, Example 1

Read four Input Registers from Input Registers in the remote Modbus/TCP device. Store the registers at location %R20. Return the COMMREQ Status word to %R10.

Word Offset	Dec	(Hex)	Description
Word 1	00008	(0008)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03003	(0BBB)	Read from a Modbus/TCP Device
Word 8	00006	(0006)	Channel number (6)
Word 9	00004	(0004)	Modbus Function Code (Read Input Registers)
Word 10	00008	(0008)	Local PLC Memory Type
Word 11	00020	(0014)	Local PLC Starting Address
Word 12	00200	(00C8)	Address in the Remote Server
Word 13	00004	(0004)	Number of Registers in the Remote Device
Word 14	00001	(0001)	Unit Identifier

### Further Considerations for Memory Locations 1-14

**(Word 7) Channel Command Number:** Word 7 identifies the COMMREQ as a Read Data from Modbus/TCP Device command block.

**(Word 8) Channel Number:** Word 8 identifies the channel number previously allocated for communication with the remote Modbus/TCP server.

**(Word 9) Modbus Function Code:** Word 9 specifies Modbus Function Code 4, Read Input Registers.

**(Word 10) Local PLC Memory Type:** Words 10-11 specify the location in the local PLC where the Ethernet interface will store data received from the remote device. For more information, refer to *Local Memory Type*.

**(Word 11) Local PLC Memory Address:** Word 11 determines the starting address in the local PLC in which the data from the remote device is to be stored. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 10. This offset will be either in bits, bytes, or words depending on the mode specified. Valid ranges of values depend on the PLC's memory ranges. Be sure this area is large enough to contain the requested data without overwriting other application data.

**(Word 12) Remote Device Address:** Word 12 specifies the address in the remote Modbus/TCP device.

**(Word 13) Number Registers in Remote Device:** Word 13 specifies the quantity of registers (16bit words) to read from the remote device.

**(Word 14) Unit Identifier:** This field is typically used by Ethernet to Serial bridges to specify the address of a Modbus Slave on a multi-drop link. The Modbus/TCP Unit Identifier is a special control code used in a Modbus/TCP message block.



## Local PLC Memory Type

Words 10-11 specify the location in the local PLC where the Ethernet interface will store data received from the remote device. Valid values for Word 10 are listed below.

Type	Value (Decimal)	Description
%W	196	Word memory (word mode)
%R	8	Register memory (word mode)
%AI	10	Analog input memory (word mode)
%AQ	12	Analog output memory (word mode)
%I	16	Discrete input memory (byte mode)
	70	Discrete input memory (bit mode)
%Q	18	Discrete output memory (byte mode)
	72	Discrete output memory (bit mode)
%T	20	Discrete temporary memory (byte mode)
	74	Discrete temporary memory (bit mode)
%M	22	Discrete momentary internal memory (byte mode)
	76	Discrete momentary internal memory (bit mode)
%SA	24	Discrete system memory group A (byte mode)
	78	Discrete system memory group A (bit mode)
%SB	26	Discrete system memory group B (byte mode)
	80	Discrete system memory group B (bit mode)
%SC	28	Discrete system memory group C (byte mode)
	82	Discrete system memory group C (bit mode)
%S <sup>18</sup>	30	Discrete system memory (byte mode)
	84	Discrete system memory (bit mode)
%G	56	Discrete global data table (byte mode)
	86	Discrete global data table (bit mode)

## Command 3003, Example 2

Read nine (9) Input Discretes starting from Discrete input address 5 in the remote Modbus/TCP server. Store the registers at location %T3(bit mode). Return the COMMREQ Status word to %R10.

Word Offset	Dec	(Hex)	Description
Word 1	00008	(0008)	Length of Channel Command Data Block (8–14 words)
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03003	(0BBB)	Read from a Modbus/TCP Device
Word 8	00006	(0006)	Channel number (6)
Word 9	00002	(0002)	Modbus Function Code (Read Input Discretes)
Word 10	00074	(004A)	Local PLC Memory Type
Word 11	00003	(0003)	Local PLC Starting Address
Word 12	00005	(0005)	Address in the Remote Device
Word 13	00009	(0009)	Number of Input Discretes to Read from the Remote Device
Word 14	00001	(0001)	Unit Identifier

### *Further Considerations for Memory Locations 1-14*

**(Word 7) Channel Command Number:** Word 7 identifies the COMMREQ as a Read Data from Modbus/TCP Device command block.

**(Word 8) Channel Number:** Word 8 identifies the channel number previously allocated for communication with the remote Modbus/TCP server.

**(Word 9) Modbus Function Code:** Word 9 specifies Modbus Function Code 2, Read Input Discretes.

**(Word 10) Local PLC Memory Type:** Words 10-11 specify the location in the local PLC where the Ethernet interface will store data received from the remote device. Valid values for Word 10 are listed on page

**(Word 11) Local PLC Memory Address:** Word 11 determines the starting address in the local PLC in which the data from the remote device is to be stored. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 10. This offset will be either in bits, bytes, or words depending on the mode specified. Valid ranges of values depend on the PLC's memory ranges. Be sure this area is large enough to contain the requested data without overwriting other application data.

**(Word 12) Remote Device Address:** Word 12 specifies the address in the remote Modbus/TCP device

**(Word 13) Number Registers in Remote Device:** Words 13 specifies the quantity of input discretes to read from the remote device.

**(Word 14) Unit Identifier:** This field is typically used by Ethernet to Serial bridges to specify the address of a Modbus Slave on a multi-drop link. The Modbus/TCP Unit Identifier is a special control code used in a Modbus/TCP message block.

## Command 3003, Example 3 – Read Exception Status

Read the Exception Status from the remote Modbus/TCP server. Store the Exception Data at location %Q4 (bit mode). Return the COMMREQ Status word to %R10.

Word Offset	Dec	(Hex)	Description
Word 1	00008	(0008)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03003	(0BBB)	Read from a Modbus/TCP Device
Word 8	00006	(0006)	Channel number (6)
Word 9	00007	(0007)	Modbus Function Code (Read Exception Status)
Word 10	00072	(0048)	Local PLC Memory Type
Word 11	00004	(0004)	Local PLC Starting Address
Word 12	00000	(0000)	Reserved
Word 13	00001	(0001)	Data Size
Word 14	00001	(0001)	Unit Identifier

### *Further Considerations for Memory Locations 1-14*

**(Word 7) Channel Command Number:** Word 7 identifies the COMMREQ as a Read Exception Status from the Modbus/TCP device.

**(Word 8) Channel Number:** Word 8 identifies the channel number previously allocated for communication with the remote Modbus/TCP server.

**(Word 9) Modbus Function Code:** Word 9 specifies Modbus Function Code 7, Read Exception Status.

**(Word 10) Local PLC Memory Type:** Words 10-11 specify the location in the local PLC where the Ethernet interface will store data received from the remote device. Valid values for Word 10 are listed on page 174.

**(Word 11) Local PLC Memory Address:** Word 11 determines the starting address in the local PLC in which the data from the remote device is to be stored. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 10. This offset will be either in bits, bytes, or words depending on the mode specified. Valid ranges of values depend on the PLC's memory ranges. Be sure this area is large enough to contain the requested data without overwriting other application data.

**(Word 12) Reserved:** Word 12 is reserved and must be set to zero.

**(Word 13) Data Size:** Word 13 is the data size and must be set to 1.

**(Word 14) Unit Identifier:** This field is typically used by Ethernet to Serial bridges to specify the address of a Modbus Slave on a multi-drop link. The Modbus/TCP Unit Identifier is a special control code used in a Modbus/TCP message block.

## Command 3003, Example 4 – Read FIFO Queue

Read the FIFO Queue from the remote Modbus/TCP server. Store the FIFO Queue Data at location %W1. Return the COMMREQ Status word to %R10.

Word Offset	Dec	(Hex)	Description
Word 1	00008	(0008)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03003	(0BBB)	Read from a Modbus/TCP Device
Word 8	00006	(0BBB)	Channel number (6)
Word 9	00024	(0BBB)	Modbus Function Code (Read FIFO Queue)
Word 10	00196	(00C4)	Local PLC Memory Type
Word 11	00001	(0001)	Local PLC Starting Address
Word 12	00048	(0030)	FIFO Pointer Address
Word 13	00001	(0001)	Data Size (Unused)
Word 14	00001	(0001)	Unit Identifier

### *Further Considerations for Memory Locations 1-14*

**(Word 7) Channel Command Number:** Word 7 identifies the COMMREQ as a Read Exception Status from the Modbus/TCP device.

**(Word 8) Channel Number:** Word 8 identifies the channel number previously allocated for communication with the remote Modbus/TCP server.

**(Word 9) Modbus Function Code:** Word 9 specifies Modbus Function Code 24, Read FIFO Queue.

**(Word 10) Local PLC Memory Type:** Words 10-11 specify the location in the local PLC where the Ethernet interface will store data received from the remote device. Valid values for Word 10 are listed on page 174.

**(Word 11) Local PLC Memory Address:** Word 11 determines the starting address in the local PLC in which the data from the remote device is to be stored. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 10.

**(Word 12) FIFO Pointer Address:** Word 12 is the FIFO pointer address in the Remote Device.

**(Word 13) Data Size:** Word 13 is unused because the return data size is dependent on the number of items in the server's FIFO queue when the command is received. Zero (0) through 32 registers can be returned as a result of this function code.

**(Word 14) Unit Identifier:** This field is typically used by Ethernet to Serial bridges to specify the address of a Modbus Slave on a multi-drop link. The Modbus/TCP Unit Identifier is a special control code used in a Modbus/TCP message block.

## 10.4.4 Write Data to a Modbus/TCP Device (3004)

The Write Data to a Modbus/TCP Device COMMREQ requests a data transfer from the PLC to a Modbus/TCP server. The Write Data COMMREQ must reference an active Modbus/TCP channel previously established with the Open Modbus/TCP Client Connection COMMREQ.

Registers or Coils may be written to the remote Modbus/TCP device. The Modbus Function Code specifies the data type. Valid Function Codes for the Write Data COMMREQ are presented in the following table:

Function Code	Description	Modbus Server Memory Region Accessed	Data Unit Size	Maximum Data Units
5	Write Single Coil	Internal Bits or Physical coils	Bit	1
6	Write Single Register	Internal Registers or Physical Output Registers	Register	1
15	Write Multiple Coils	Internal Bits or Physical coils	Bit	1968
16	Write Multiple Registers	Internal Registers or Physical Output Registers	Register	123

An Address Offset and Length specify the location in the Modbus/TCP device and the number of data units to transfer. The Address Offset is the offset from the Base Address for that memory region in the server. The Length is the number of Registers or Coils to transfer.

A PLC data area is the source for the data written to the Modbus/TCP device.

**Function Code 5**, Write Single Coil, forces a Coil On or Off. To force a coil off, the value zero (0) is used as the COMMREQ data value. If the PLC memory type is a bit type, the remote device coil is set to the same state as the specified PLC memory location. If the PLC memory type is a byte or word type, a value of zero (0) is used to force a coil off and a value of one (1) is used to force a coil on.

**Function Code 15**, Write Multiple Coils, forces multiple Coils On or Off. If the PLC memory type is a bit type, remote device coils are set to the same state as the corresponding bits in the specified PLC memory location. If the PLC memory type is a byte or word type, the remote device coils follow the state of the packed bits contained in the byte or word memory. For example, if 16 coils are written to a PACSystems Modbus server starting at %Q1 from the client PLC memory at %R1 containing a value of 0x1111, the following remote server coils will be set %Q1, %Q5, %Q9, and %Q13 and the following remote server bits will be cleared: %Q2, %Q3, %Q4, %Q6, %Q7, %Q8, %Q10, %Q11, %Q12, %Q14, %Q15, %Q16.

The COMMREQ Status Word (CRS) indicates the success or failure of the Write Data COMMREQ. If the COMMREQ specifies an invalid channel number or any other invalid field the COMMREQ fails and the CRS is set to a non-zero value to identify the failure.

## Command 3004, Example 1 – Set Single Register

Write one register from %AI10 to register address 200 in the remote Modbus/TCP server. Return the COMMREQ Status word to %R10. Use channel 6, a channel previously opened with the Open Modbus/TCP Client Connection COMMREQ.

Word Offset	Dec	(Hex)	Description
Word 1	00008	(0008)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03004	(0BBC)	Write to a Modbus/TCP Device
Word 8	00006	(0006)	Channel number (6)
Word 9	00006	(0006)	Modbus Function Code – Write Single Register
Word 10	00010	(000A)	Local PLC Memory Type
Word 11	00010	(000A)	Local PLC Starting Address
Word 12	00200	(00C8)	Address in the Remote Device
Word 13	00001	(0001)	Number of Registers in the Remote Device
Word 14	00001	(0001)	Unit Identifier

### *Further Considerations for Memory Locations 1-14*

**(Word 7) Channel Command Number:** Word 7 identifies the COMMREQ as a Write Data to remote Modbus/TCP device.

**(Word 8) Channel Number:** Word 8 identifies the channel number previously allocated for communication with the remote Modbus/TCP server.

**(Word 9) Modbus Function Code:** Word 9 specifies Function Code 6, Write Single Register.

**(Word 10) Local PLC Memory Type:** Words 10–11 specify the location in the local PLC from where the Ethernet interface will get the data to be written to the remote PLC. Valid values for Word 10 are listed on page 174.

**(Word 11) Local PLC Starting Address:** Word 11 determines the starting address in the local PLC from which the data is to be written. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 10.

**(Word 12) Remote Device Address:** specifies the destination register in the remote device.

**(Word 13) Number Registers in Remote Device:** Word 13 specifies the quantity of registers to write to the remote device. For Function Code 6, Write Single Register this must be set to 1.

**(Word 14) Unit Identifier:** This field is typically used by Ethernet to Serial bridges to specify the address of a Modbus Slave on a multi-drop link. The Modbus/TCP Unit Identifier is a special control code used in a Modbus/TCP message block.

## Command 3004, Example 2 – Write Single Coil

Set coil 501 ON in the remote Modbus/TCP device using the value at %Q4. Return the COMMREQ Status word to %R10. Use channel 6, a channel previously opened with the Open Modbus/TCP Client Connection COMMREQ.

Word Offset	Dec	(Hex)	Description
Word 1	00008	(0008)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03004	(0BBC)	Write to a Modbus/TCP Device
Word 8	00006	(0006)	Channel number (6)
Word 9	00005	(0005)	Modbus Function Code – Write Single Coil
Word 10	00072	(0048)	Local PLC Memory Type
Word 11	00004	(0004)	Local PLC Starting Address
Word 12	00501	(01F5)	Address in the Remote Device
Word 13	00001	(0001)	Number of Coils in the Remote Device.
Word 14	00001	(0001)	Unit Identifier

### Further Considerations for Memory Locations 1-14

**(Word 7) Channel Command Number:** Word 7 identifies the COMMREQ as a Write Data to Modbus/TCP device.

**(Word 8) Channel Number:** Word 8 identifies the channel number previously allocated for communication with the remote Modbus/TCP server.

**(Word 9) Modbus Function Code:** Word 9 specifies Modbus Function Code 5 Write Single Coil.

**(Word 10) Local PLC Memory Type:** Words 10–11 specify the location in the local PLC from where the Ethernet interface will get the data to be written to the remote PLC. Valid values for Word 10 are listed on page 174.

**(Word 11) Local PLC Starting Address:** Word 11 determines the starting address in the local PLC from which the data is to be written. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 10. This offset will be either in bits, bytes, or words depending on the mode specified. Valid ranges of values depend on the PLC's memory ranges.

**(Word 12) Remote Device Address:** Word 12 specifies the destination coil address in the Modbus/TCP device.

**(Word 13). Number Coils in Remote Device:** Words 13 specifies the quantity of coils to write to the remote device. For Modbus Function Code 5, Write Single Coil, this must be set to 1.

**(Word 14) Unit Identifier:** This field is typically used by Ethernet to Serial bridges to specify the address of a Modbus Slave on a multi-drop link. The Modbus/TCP Unit Identifier is a special control code used in a Modbus/TCP message block.

## Command 3004, Example 3 – Set Multiple Registers

Write the four registers from Discrete Input Memory (%I40 to) address 200 in the remote Modbus/TCP server. Return the COMMREQ Status word to %R10. Use channel 6, a channel previously opened with the Open Modbus/TCP Client Connection COMMREQ.

Word Offset	Dec	(Hex)	Description
Word 1	00008	(0008)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03004	(0BBC)	Write to a Modbus/TCP Device
Word 8	00006	(0006)	Channel number (6)
Word 9	00016	(0010)	Modbus Function Code – Write Multiple Registers
Word 10	00016	(0010)	PLC Memory Type
Word 11	00040	(0028)	PLC Starting Address
Word 12	00200	(00C8)	Address in the Remote Device
Word 13	00004	(0004)	Number of Registers in the Remote Device
Word 14	00001	(0001)	Unit Identifier



## 10.4.5 Mask Write Register Request to a Modbus Server Device (3009)

The Mask Write Register Request to a Modbus Server Device COMMREQ is used to modify the contents of a specified remote device register using a combination of an AND mask, OR mask, and the current register's value. This function is used to set or clear individual bits in a register. The register is modified per the following algorithm:

Register value = ((Current register value) AND (And Mask Value)) OR  
((OR Mask Value) AND (NOT(And Mask Value)))

### Command 3009, Example – Mask Write Register

Modify register at address 200 in the remote Modbus/TCP server and clear all bits except bit 0. Return the COMMREQ Status word to %R10. Use channel 6, a channel previously opened with the Open Modbus/TCP Client Connection COMMREQ.

Word Offset	Dec	(Hex)	Description
Word 1	00008	(0008)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03009	(0BC1)	Mask Write Register to a Modbus/TCP Server Device
Word 8	00006	(0006)	Channel number (6)
Word 9	00022	(0016)	Modbus Function Code – Write Mask Register
Word 10	00200	(00C8)	Address in the Remote Device
Word 11	00001	(0001)	AND Mask
Word 12	00000	(0000)	OR Mask
Word 13	00001	(0001)	Unit Identifier

**(Word 7) Channel Command Number:** Word 7 identifies the COMMREQ as a Mask Write Register operation on remote Modbus/TCP device.

**(Word 8) Channel Number:** Word 8 identifies the channel number previously allocated for communication with the remote Modbus/TCP server.

**(Word 9) Modbus Function Code:** Word 9 specifies Function Code 22, Mask Write Register.

**(Word 10) Remote Device Address:** specifies the destination register in the remote device.

**(Word 11) AND Mask:** Word 11 specifies the AND mask to be used in the Mask Write operation. For this example, all bits are cleared except bit 0.

**(Word 12) OR Mask:** Word 12 specifies the OR mask to be used in the Mask Write operation. In this example, no bits are to be set.

**(Word 13) Unit Identifier:** This field is typically used by Ethernet to Serial bridges to specify the address of a Modbus Slave on a multi-drop link. The Modbus/TCP Unit Identifier is a special control code used in a Modbus/TCP message block.

## 10.4.6 Read/Write Multiple Registers to/from a Modbus Server Device (3005)

The Read/Write Multiple Registers to/from a Modbus Server Device COMMREQ is used to read and write data between the remote server and the PLC with one COMMREQ operation. The write operation occurs first and the data exchange does not occur coherently (i.e. data can change in the server between the write and read operations).

### Command 3005, Example – Read/Write Multiple Register

Write 10 values starting at %R100 in the Local PLC to register address 200 in the remote Modbus/TCP server and read 20 values starting from register 300 in the remote Modbus/TCP server and write this value to %R300 in the Local PLC. Return the COMMREQ Status word to %R10. Use channel 6, a channel previously opened with the Open Modbus/TCP Client Connection COMMREQ.

Word Offset	Dec	(Hex)	Description
Word 1	00014	(000E)	Length of Channel Command Data Block
Word 2	00000	(0000)	Always 0 (no-wait mode request)
Word 3	00008	(0008)	Memory type of CRS word (%R)
Word 4	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03005	(0BB5)	Read/Write Multiple Registers to/from a Modbus/TCP Device
Word 8	00006	(0006)	Channel number (6)
Word 9	00023	(0017)	Modbus Function Code – Read/Write Multiple Registers
Word 10	00008	(0008)	Local PLC Memory Type of memory to write with data read from Remote Device
Word 11	00300	(012C)	Local PLC Starting Address (LSW) of memory to write with data read from Remote Device

#### *Further Considerations for Memory Locations 1-11*

**(Word 7) Channel Command Number:** Word 7 identifies the COMMREQ as a Read/Write Multiple Register operation on remote Modbus/TCP device.

**(Word 8) Channel Number:** Word 8 identifies the channel number previously allocated for communication with the remote Modbus/TCP server.

**(Word 9) Modbus Function Code:** Word 9 specifies Function Code 23, Read/Write Multiple Register.

**(Word 10) Local PLC Memory Type (Write With Data Read From Server):** Words 10–12 specify the location in the local PLC where the Ethernet interface will write data received from the remote server. The value 8 specifies Register Memory %R.

**(Word 11) Local PLC Starting Address LSW (Write With Data Read From Server):** Word 11 determines the least significant word (LSW) of the starting address in the local PLC from which the data is to be written. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 10. This offset will be either in bits, bytes, or words depending on the mode specified. Valid ranges of values depend on the PLC's memory ranges.

Word Offset	Dec	(Hex)	Description
Word 12	00000	(0000)	Local PLC Starting Address (MSW) of memory to write with data read from Remote Device (normally 0 unless %W is used)
Word 13	00300	(012C)	Address to Read From on Remote Server
Word 14	00020	(0014)	Number of Memory Units to Read from Remote Device (1 to 125)
Word 15	00008	(0008)	Local PLC Memory Type of memory to use for writing to the Remote Device
Word 16	00100	(0064)	Local PLC Starting Address (LSW) of memory to use for writing to the Remote Device
Word 17	00000	(0000)	Local PLC Starting Address (MSW) of memory to use for writing to the Remote Device (normally 0 unless %W is used)
Word 18	00200	(00C8)	Address to Write to on the Remote Server
Word 19	00010	(000A)	Number of Memory Units to Write to the Remote Device (1 to 121)
Word 20	00001	(0001)	Unit Identifier

### *Further Considerations for Memory Locations 12-20*

**(Word 12) Local PLC Starting Address MSW (Write With Data Read From Server):** Word 12 determines the most significant word (MSW) of the starting address in the local PLC from which the data is to be written. This value will typically be 0 unless the address is above 65535 for %W memory.

**(Word 13) Remote Device Read Address:** Word 13 specifies the register(s) to read from the remote Modbus/TCP device.

**(Word 14) Number Registers to Read From Remote Device:** Words 14 specifies the quantity of registers to read from the remote device.

**(Word 15) Local PLC Memory Type (Read Data to Write to Server):** Words 15–17 specify the location in the local PLC where the Ethernet interface will read data to use for writing to the remote server. Values for Word 15 are listed on page 178. The value 8 specifies Register Memory %R.

**(Word 16) Local PLC Starting Address LSW (Read Data to Write to Server):** Word 16 determines the least significant word (LSW) of the starting address in the local PLC from which the data is to be read. The value entered is the offset (1-based) from the beginning of PLC memory for the memory type and mode specified in Word 15. This offset will be either in bits, bytes, or words depending on the mode specified. Valid ranges of values depend on the PLC's memory ranges.

**(Word 17) Local PLC Starting Address MSW (Read Data to Write to Server):** Word 17 determines the most significant word (MSW) of the starting address in the local PLC from which the data is to be read. This value will typically be 0 unless the address is above 65535 for %W memory.

**(Word 18) Remote Device Write Address:** Word 18 specifies the register(s) to be written on the remote Modbus/TCP device.

**(Word 19) Number Registers to Write To Remote Device:** Words 19 specifies the quantity of registers to write to the remote device.

**(Word 20) Unit Identifier:** This field is typically used by Ethernet to Serial bridges to specify the address of a Modbus Slave on a multi-drop link. The Modbus/TCP Unit Identifier is a special control code used in a Modbus/TCP message block.

## 10.5 Status Data

This section describes all the status data that is available to the ladder program to determine the state of the Ethernet interface and its Modbus/TCP channels.

### 10.5.1 Types of Status Data

There are three main types of status data available to the application program:

- Ethernet interface status bits,
- FT Output of the COMMREQ function block
- COMMREQ Status Word

#### Ethernet interface Status Bits

The status bits are updated in the CPU once each PLC scan by the Ethernet interface. These bits are generally used to prevent the initiation of a COMMREQ function when certain errors occur or to signal a problem on an established channel. The status bits include the LAN interface Status bits and the Channel Status bits. The starting location of these bits is set up when the module is configured.

The LAN interface Status bits monitor the health of the Ethernet interface itself, such as the LAN interface OK bit. The Channel Status bits monitor the health of a channel. Each Modbus channel has a dedicated status bit.

For details of the status bits and their operation, refer to Section 12.6, Monitoring the Ethernet Interface Status Bits.

#### FT Output of the COMMREQ Function Block

This output is set if there is a programming error in the COMMREQ Function Block itself, if the rack and slot specified in the COMMREQ SYSID parameter are not configured to contain an Ethernet interface, or if the data block length specified in the Command Block is out of range. This output also may indicate that no more COMMREQ functions can be initiated in the ladder program until the Ethernet interface has time to process some of the pending COMMREQ functions.

If the FT Output is set, the CPU does not transfer the Command Block to the Ethernet interface. In this case, the other status indicators are not updated for this COMMREQ.

The FT Output passes power upon the following errors:

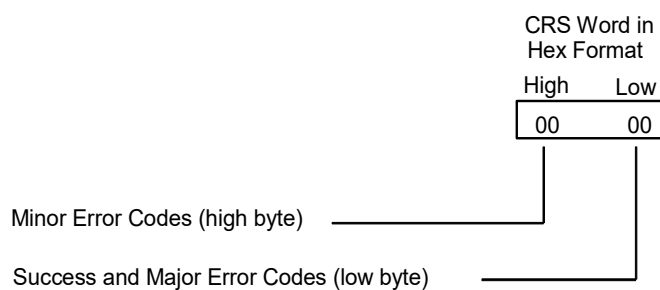
- Invalid rack/slot specified. The module at this rack/slot is unable to receive a COMMREQ.
- Invalid Task ID.
- Invalid Data Block length (zero or greater than 128).
- Too many simultaneous active COMMREQs (overloading either the PLC CPU or the Ethernet interface).

## COMMREQ Status Word

The COMMREQ Status word (CRS word) provides detailed information on the status of the COMMREQ request. The communications status word is not updated in the CPU each scan as are the status bits. They are generally used to determine the *cause* of a communication error after the COMMREQ function is initiated. The cause is reported in the form of an error code described later in this section. The COMMREQ Status word (CRS word) is returned from the Ethernet interface to the PLC CPU immediately if the Command Block contains a syntax error or if the command is local. The location of the CRS word is defined in the Command Block for the COMMREQ function.

The COMMREQ Status word (CRS word) reports status in the format shown below. The CRS word location is specified in Words 3 and 4 of the Command Block.

**Figure 61: Interpreting the COMMREQ Status Word**



The Ethernet interface reports the status of the COMMREQ back to the status location. See Section 12, *Diagnostics*, for COMMREQ major and minor error codes that may be reported in the CRS words for Modbus/TCP commands.

## 10.6 Controlling Communications in the Ladder Program

This section provides tips on how to control communications in your ladder program. Only segments of actual ladder logic are included. Topics discussed are:

- Essential Elements of the Ladder Program
- Troubleshooting Your Ladder Program
- Monitoring the Communications Channel

### 10.6.1 Essential Elements of the Ladder Program

Every ladder program, whether in the developmental phase or the operational phase, should do the following before initiating a COMMREQ function.

1. Initiate the COMMREQ function with a one-shot transitional coil. This prevents sending the same COMMREQ Command Block more than once.
2. Include at least the LAN interface OK bit in the LAN interface Status Word as an interlocking contact for the COMMREQ function. You may choose to add more interlocks.
3. Zero the word location you specify for the COMMREQ Status (CRS) word and the FT Outputs of the COMMREQ Function Block before the COMMREQ function is initiated.

4. Move the command code and parameters for the Channel command into the memory location specified by the IN input of the COMMREQ Function Block before the COMMREQ function is initiated.

The example ladder program segment starting on the next page illustrates how to incorporate these important points in your program.

**Note:** When using a Write Data or Read/Write COMMREQ, data is not read from the local PLC synchronously with the execution of the COMMREQ. A number of CPU sweeps may occur before the data is read. It is recommended that the data not be changed until after the COMMREQ Status Word indicates completion of the command.

## 10.6.2 Managing Channels and TCP Connections

When you issue a COMMREQ to open a Modbus channel, a TCP connection is created, the transfer then can be made, then upon “closing” of the channel, the TCP connection is terminated. It takes time to create and to terminate these connections. If an application is constructed so that it rapidly and repeatedly open and close a channel(s), the available TCP connections for the Ethernet interface may be totally consumed. A “snapshot” of the state of the TCP connections would show some of them being created, some being terminated, and some active, but none available.

### CAUTION

In Certain Conditions TCP Connections Can Be Totally Consumed

## 10.6.3 Client Channels TCP Resource Management

There is a period of time that the OS Network stack hangs on to the TCP resources associated with a connection after it is closed. It applies to the initiator of the close, which is almost always the client-side. This time is referred to as the “TCP Linger Period”. Once the TCP Linger Period expires (60 seconds in the current OS implementation), the TCP resources are released. Application developers using client channels need to be aware of this behavior when designing their logic. There are a finite number of TCP resources allocated to client channels, and if channel connections are brought up and down so fast that these resources are depleted, then the application may have to wait until a TCP resource frees up to establish another client channel (a COMMREQ Status of 0xA890 is returned if no TCP resources are currently available; the application should wait and retry again).

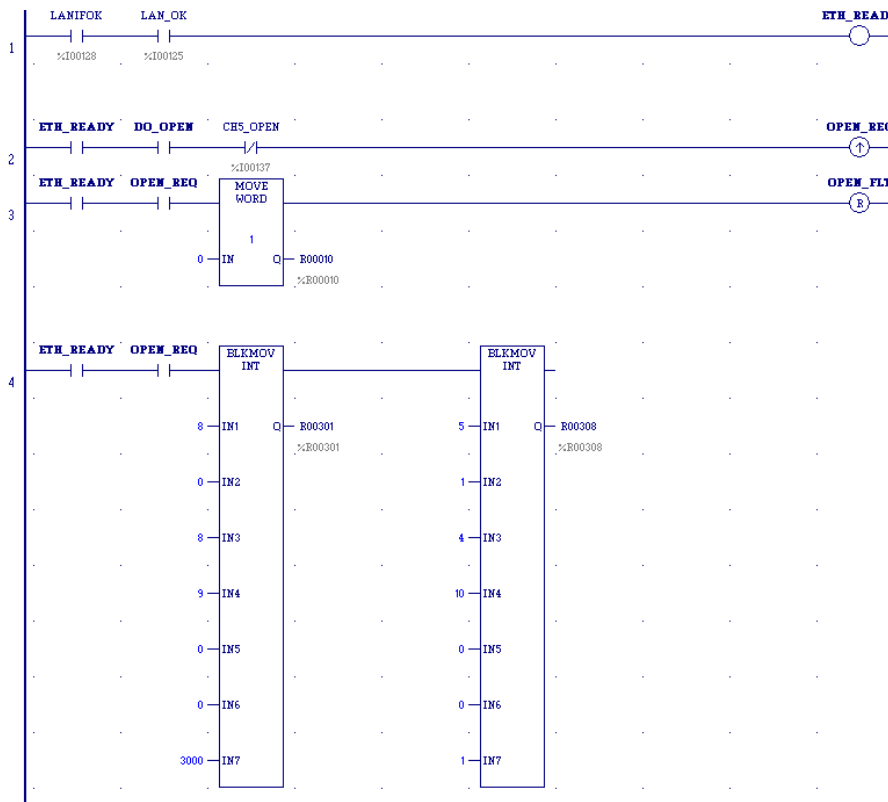
Modbus Client Channels provides features that help the user preserve TCP connections. This includes opening a channel where one can re-task an active channel to the same remote device, where the parameters of an active channel, such as changing the channel command type (Read/Write), local memory address, remote memory address, etc. can be changed.

## 10.6.4 COMMREQ Ladder Logic Example

The input values for the Block Move Functions in the example below are taken from the Open Modbus/TCP Connection (3000), Modbus/TCP Read (3003), and Close Modbus/TCP Connection (3001) Examples in this chapter.

Named variables are used in this example to make the ladder program easier to follow. LANIFOK is bit 16 of the LAN interface Status bits. LAN\_OK is bit 13 of the LAN interface Status bits. CH5\_OPEN is bit 25 of the LAN interface Status bits. All other nicknames may be assigned as you choose.

**Figure 62: COMMREQ Ladder Logic Segment**



**Rung # 1:** Input LANIFOK (bit 16 of the LAN interface Status bits) monitors the health of the Ethernet interface. Input LAN\_OK (bit 13 of the LAN interface Status bits) monitors the online/offline status of the Ethernet interface. If both bits are set, the network connection is available and healthy. CH5\_OPEN (bit 25 of the LAN interface Status bits) monitors channel 5 Connection Open status: it needs to be 0 (off). If these conditions are met, it is okay to send a COMMREQ, so the ETH\_READY coil is turned ON. ETH\_READY is used as an interlock for Rungs 2-16 in this example.

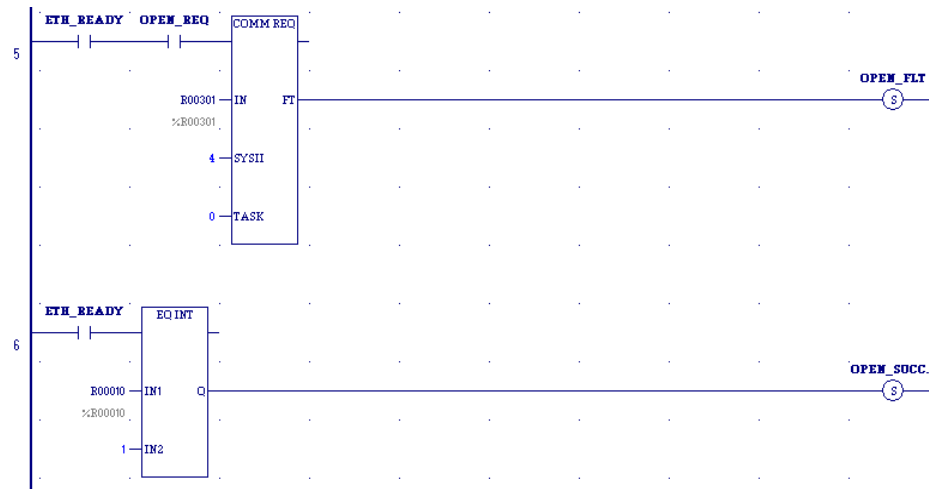
**Rung # 2:** When ETH\_READY is set and CH5\_OPEN is not set, Input DO\_OPEN triggers OPEN\_REQ, which enables execution of the MOVE and COMMREQ functions for the Open Modbus/TCP Connection COMMREQ. OPEN\_REQ is a one-shot (Positive Transition) coil, activating once when both ETH\_READY and DO\_OPEN have transitioned from OFF to ON.

**Rung # 3:** The MOVE WORD function moves a zero to the CRS word referenced in the Command Block (see rung #4). This clears the CRS word. This rung also resets the OPEN\_FLT output coil of the COMMREQ Function Block in rung #5.

It is vital that the CRS Status Word is cleared and the COMMREQ fault output coil is cleared each time before initiating a COMMREQ function.

**Rung # 4:** The BLKMOV INT functions to set up the COMMREQ Command Block contents. When this rung is activated, the constant operands are moved into the memory beginning at the address indicated in the instruction. The constant operands in this example are defined in the Open Modbus/TCP Connection Example in this chapter.

**Figure 63: COMMREQ Ladder Logic Segment (continued)**



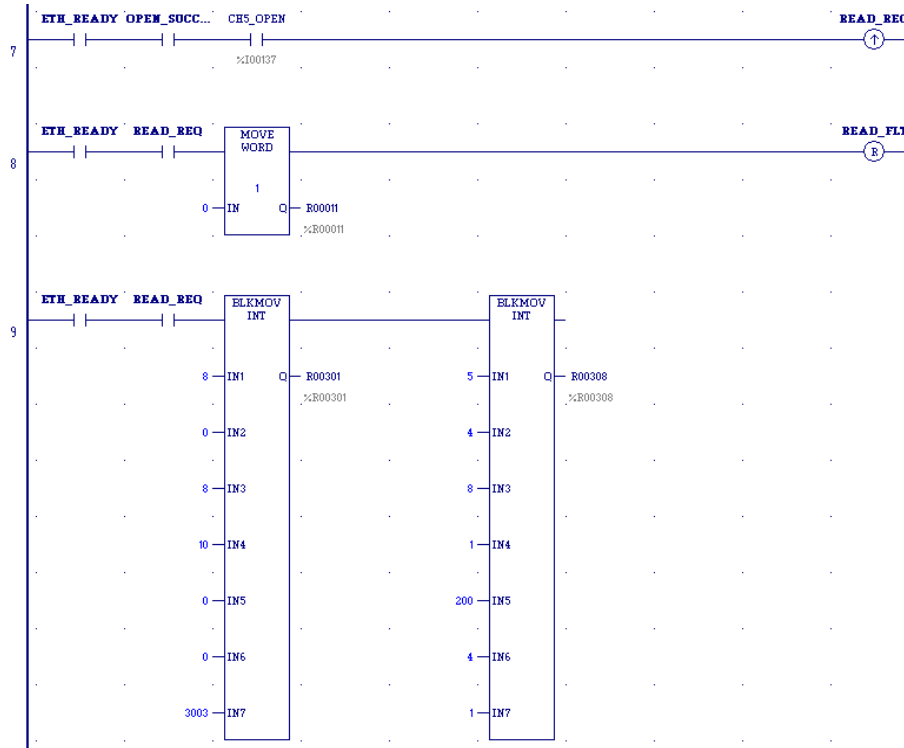
**Rung # 5:** The COMMREQ Function Block has three input parameters and one output parameter.

- The IN field points to the starting location of the Command Block parameters (%R00301 in this example).
- The SYSID field of the COMMREQ Function Block defines the target rack and slot of the Ethernet interface to receive the command data. This is a hexadecimal word value that gives the rack (high byte) and slot (low byte) location of the Ethernet interface module. In the example, the first three number places (from left to right) are zeroes and are not displayed; only the last number, 4, appears. This indicates rack 0, slot 4.
- The TASK field of the COMMREQ Function Block indicates which mailbox task ID to use for the specified rack and slot. For the RX3i ETM001-jx Ethernet interfaces TASK must always be set to zero. For PACSystems CPU embedded Ethernet interface, TASK must be set to 65536 (0x10000) to address the CPU’s Ethernet daughterboard.
- The FT output (energizes the OPEN\_FLT coil in this example) is turned ON (set to 1) if problems were preventing the delivery of the Command Block to the Ethernet interface. In this case, the other status indicators are not updated for this COMMREQ.

**Rung # 6:** When ETH\_READY is set the CRS word for the Open Modbus/TCP Connection COMMREQ is monitored for a status of 1, indicating that the Open COMMREQ completed successfully. In addition, the Ethernet Module will set the CH5\_OPEN status. The CRS word change to 1 set coil OPEN\_SUCCESS.



Figure 64: COMMREQ Ladder Logic Segment (continued)

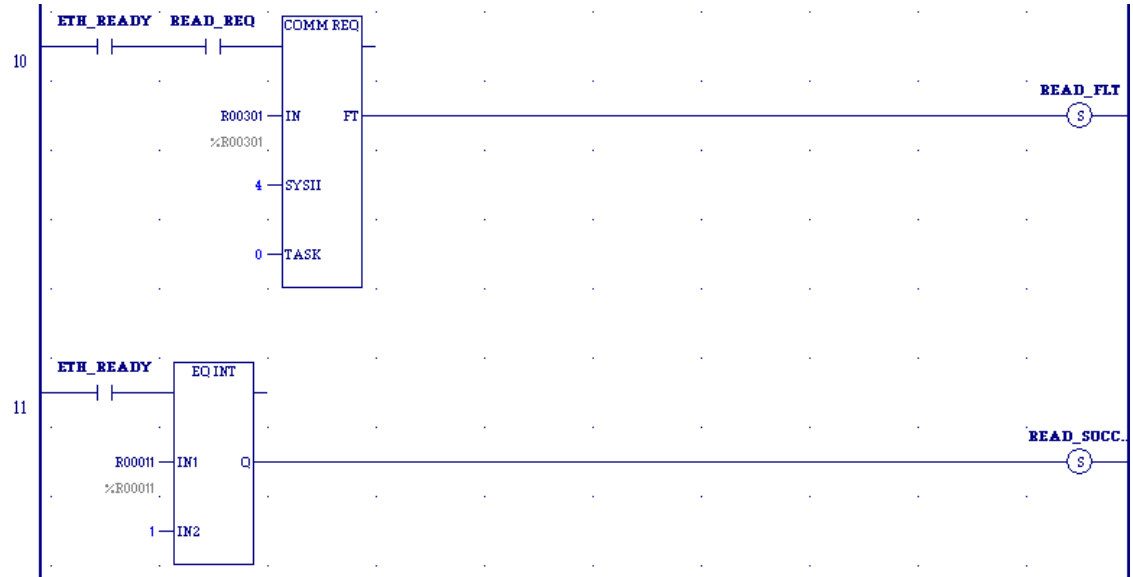


**Rung # 7:** When `OPEN_SUCCESS` and `CH5_OPEN` are set, it triggers `READ_REQ`, which enables execution of the `BLKMOV`, `MOVE`, and `COMMREQ` functions for the Modbus/TCP Read `COMMREQ`. `READ_REQ` is a one-shot (Positive Transition) coil, activating once when `OPEN_SUCCESS` transitions from OFF to ON.

**Rung # 8:** The `MOVE WORD` function moves a zero to the `CRS` word referenced in the Command Block (see rung #9). This clears the `CRS` word. This rung also resets the `READ_FLT` output coil of the `COMMREQ` Function Block in rung #10.

**Rung # 9:** The `BLKMOV INT` functions to set up the `COMMREQ` Command Block contents. When this rung is activated, the constant operands are moved into the memory beginning at the address indicated in the instruction. The constant operands in this example are defined in the Modbus/TCP Read Example in this chapter.

Figure 65: COMMREQ Ladder Logic Segment (continued)

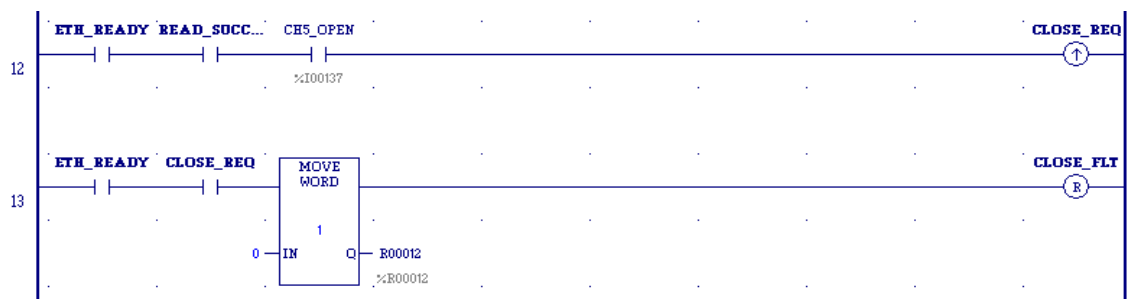


**Rung # 10:** The COMMREQ Function Block has three input parameters and one output parameter.

- The IN field points to the starting location of the Command Block parameters (%R00301 in this example).
- The SYSID field of the COMMREQ Function Block defines the target rack and slot of the Ethernet interface to receive the command data. This is a hexadecimal word value that gives the rack (high byte) and slot (low byte) location of the Ethernet interface module.
- The TASK field of the COMMREQ Function Block indicates which mailbox task ID to use for the specified rack and slot. For the RX3i ETM001-jx Ethernet interfaces TASK must always be set to zero. For PACSystems CPU embedded Ethernet interface, TASK must be set to 65536 (0x10000) to address the CPU’s Ethernet daughterboard.
- The FT output (energizes the READ\_FLT coil in this example) is turned ON (set to 1) if problems were preventing the delivery of the Command Block to the Ethernet interface. In this case, the other status indicators are not updated for this COMMREQ.

**Rung # 11:** When ETH\_READY is set the CRS word for the Modbus/TCP Read COMMREQ is monitored for a status of 1, indicating that the Read COMMREQ completed successfully. The CRS word change to 1 set coil READ\_SUCCESS.

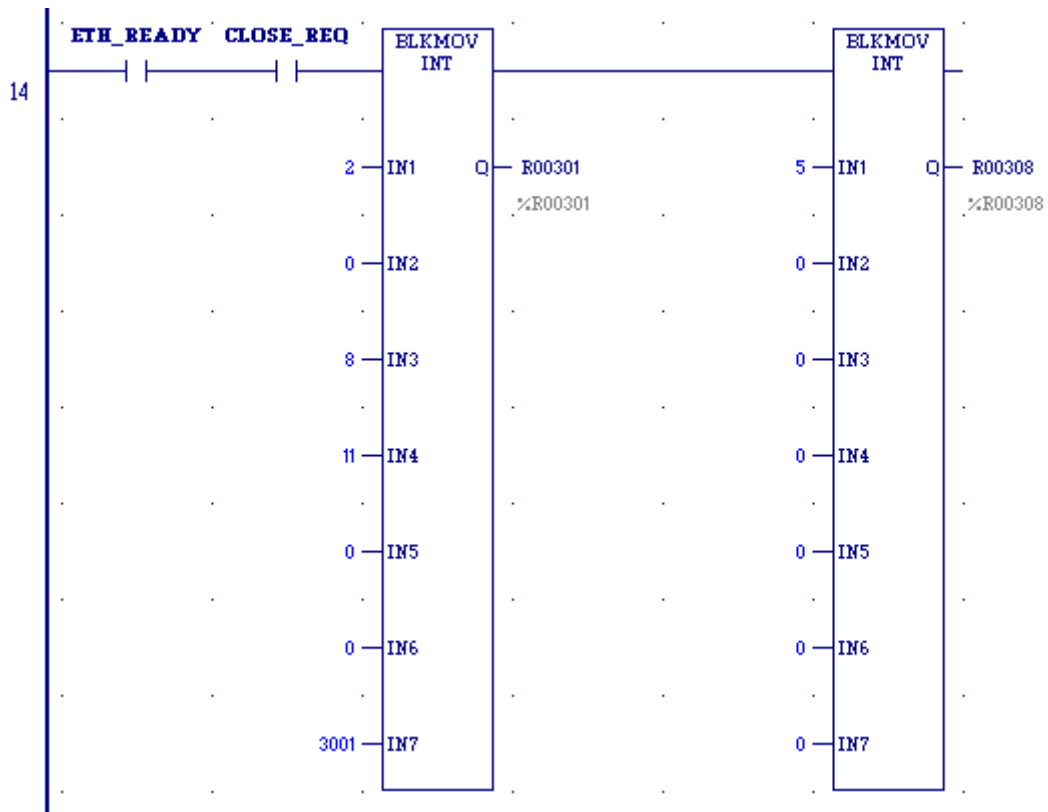
Figure 66: COMMREQ Ladder Logic Segment (continued)



**Rung # 12:** When READ\_SUCCESS and CH5\_OPEN a reset it triggers CLOSE\_REQ, which enables execution of the BLKMOV, MOVE, and COMMREQ functions for the Close Modbus/TCP Connection COMMREQ. CLOSE\_REQ is a one-shot (Positive Transition) coil, activating once when READ\_SUCCESS transitions from OFF to ON.

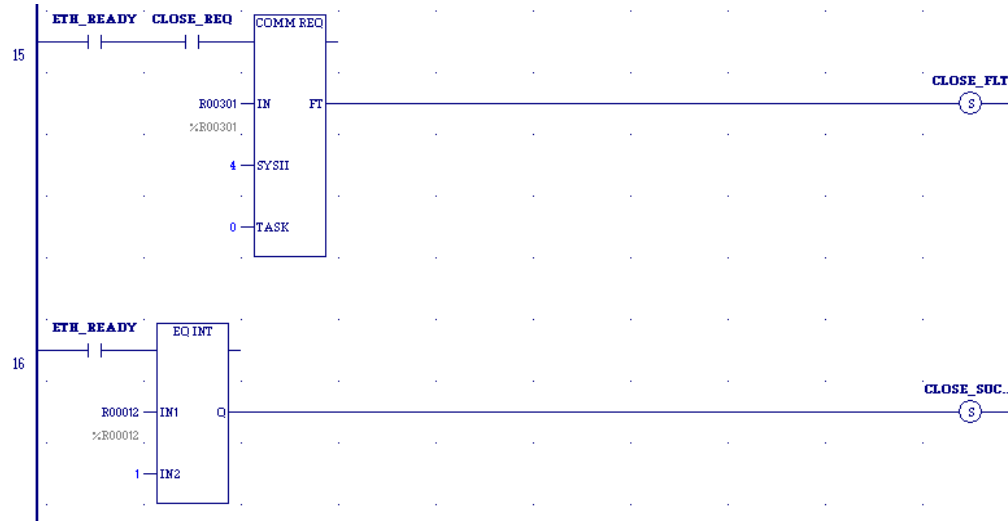
**Rung # 13:** The MOVE WORD function moves a zero to the CRS word referenced in the Command Block (see rung #9). This clears the CRS word. This rung also resets the CLOSE\_FLT output coil of the COMMREQ Function Block in rung #15.

**Figure 67: COMMREQ Ladder Logic Segment (continued)**



**Rung # 14:** The BLKMOV INT functions to set up the COMMREQ Command Block contents. When this rung is activated, the constant operands are moved into the memory beginning at the address indicated in the instruction. The constant operands in this example are defined in the Close Modbus/TCP Connection Example in this chapter.

Figure 68: COMMREQ Ladder Logic Segment (continued)



**Rung # 15:** The COMMREQ Function Block has three input parameters and one output parameter.

- The IN field points to the starting location of the Command Block parameters (%R00301 in this example).
- The SYSID field of the COMMREQ Function Block defines the target rack and slot of the Ethernet interface to receive the command data. This hexadecimal word value gives the rack (high byte) and slot (low byte) location of the Ethernet interface module.
- The TASK field of the COMMREQ Function Block indicates which mailbox task ID to use for the specified rack and slot. For the RX3i ETM001-jx Ethernet interfaces TASK must always be set to zero. For PACSystems CPU embedded Ethernet interface, TASK must be set to 65536 (0x10000) to address the CPU’s Ethernet daughterboard.
- The FT output (energizes the CLOSE\_FLT coil in this example) is turned ON (set to 1) if problems were preventing the delivery of the Command Block to the Ethernet interface. In this case, the other status indicators are not updated for this COMMREQ.

**Rung # 16:** When ETH\_READY is set the CRS word for the Close Modbus/TCP Connection COMMREQ is monitored for a status of 1, indicating that the Close COMMREQ completed successfully. In addition, the Ethernet Module will clear the CH5\_OPEN status. The CRS word change to 1 set coil CLOSE\_SUCCESS.

## 10.6.5 Troubleshooting a Ladder Program

There are several forms of status data that can be accessed by the application program. The use of the LAN interface OK bit in the LAN interface Status Word was described in the example program. Some status data can be used to troubleshoot a program in its developmental stage. The two primary sources of this data are the FT Output on the COMMREQ Function Block and the COMMREQ Status word (CRS word).

### FT Output is ON

If after executing a COMMREQ Function, the FT Output is ON, then there is a programming error in one or more of the following areas.

- Invalid rack/slot specified. The module at this rack/slot is unable to receive a COMMREQ Command Block.
- Invalid Task ID. For the RX3i ETM001-Jx Ethernet interfaces TASK must always be set to zero. For PACSystems CPU embedded Ethernet interface, TASK must be set to 65536 (0x10000) to address the CPU's Ethernet daughterboard.
- Invalid Data Block length (0 or greater than 128).

### COMMREQ Status Word is Zero (0) and FT Output is OFF

If after executing a COMMREQ function, the CRS word is zero (0) and the FT Output is OFF, then the Command Block has been sent to the Ethernet interface, but no status has been returned yet. If this condition persists, check the PLC Fault Table for information.

### COMMREQ Status Word is Not One (1)

If after executing a COMMREQ function, the CRS word is not one (1) indicating success, then there were:

- Errors in the Command Block (the Channel command code or parameters), or
- The command parameters were valid but there was an error in completing the request.

If the CRS word does not contain a 1 indicating success, then it contains either a 0 or a code indicating what error occurred.

## 10.6.6 Monitoring the Communications Channel

The status data can be used to monitor communications and take action after certain events.

### Monitoring the COMMREQ Status Word

It is critical to monitor the CRS word for each COMMREQ function. First, zero the associated CRS word before executing the COMMREQ function. When the CRS word becomes non-zero, the Ethernet interface has updated it. If the CRS word is updated to a one (1), the Command Block was processed successfully by the Ethernet interface. If the CRS word is updated to a value other than 1, an error occurred in processing the Command Block.

Do not use data received from a server until the CRS word for that channel is 1. In addition, do not initiate any additional commands to a channel until the CRS word has been updated. The exception to this rule is when you want to terminate a command by using the Close Modbus/TCP Connection command.

### Monitoring the Channel Open Bit

This bit is 1 when a Channel has successfully established a connection with a remote server and is 0 when a Channel has been closed. The Channel Open Bit is meaningful when the CPU is in Run mode and the particular channel is being used by Modbus/TCP. The Channel Open Bit is set at the same time the successful status is returned to the CRS word for the Open Modbus/TCP Connection COMMREQ.

### Sequencing Communications Requests

If the Ethernet interface receives Command Blocks from the CPU faster than it can process them, the Ethernet interface will log an exception event 08, Entry 2=0024H and will log the PLC Fault Table entry:

“Backplane Communications with PLC Fault; Lost Request”

Only one COMMREQ function per channel can be pending at one time. A COMMREQ function is pending from the time it is initiated in the ladder program until its CRS word has been updated to a non-zero value by the Ethernet interface.

## CMM321 Modbus Client Endian Conversion Example

For example, depending on the direction of the transfer, the end-to-end values result in bytes being swapped for CMM321 Modbus Client. This can be seen in the example table below.

Memory Location / Type	Memory value example	Transfer Direction	Memory Location / Type	Resulting Value After Transfer	Notes
Client Bit	%M16-%M1 = 0x4321	→	Server Word	%R1 = 0x4321	End-to-end bytes un-swapped
Server Bit	%M16-%M1 = 0x4321	→	Client Word	%R1 = 0x2143	<b>End-to-end bytes swapped</b>
Client Word	%R1 = 0x4321	→	Server Bit	%M16-%M1 = 0x4321	End-to-end bytes un-swapped
Server Word	%R1 = 0x4321	→	Client Bit	%M16-%M1 = 0x2143	<b>End-to-end bytes swapped</b>

## PACSystems Modbus Client Endian Conversion Example

The following example table shows the Endian conversion behavior for the PACSystems Modbus Client:

Memory Location / Type	Memory value example	Transfer Direction	Memory Location / Type	Resulting Value After Transfer	Notes
Client Bit	%M16-%M1 = 0x4321	→	Server Word	%R1 = 0x4321	End-to-end bytes un-swapped
Server Bit	%M16-%M1 = 0x4321	→	Client Word	%R1 = 0x4321	<b>End-to-end bytes un-swapped</b>
Client Word	%R1 = 0x4321	→	Server Bit	%M16-%M1 = 0x4321	End-to-end bytes un-swapped
Server Word	%R1 = 0x4321	→	Client Bit	%M16-%M1 = 0x4321	<b>End-to-end bytes un-swapped</b>



# Section 11 OPC UA Server

OPC Unified Architecture, or OPC UA, is a communication standard published by the OPC Foundation to provide data communications interoperability for industrial automation. This standard specifies client-server communications with a service-oriented architecture. It is typically used to allow automation controller servers (such as the PACSystems Controllers) to share process data for the purposes of monitoring, control, supervision, and logging with Human-Machine Interface (HMI), workstation, alarm system, condition monitoring, and historian clients.

The embedded OPC UA server provided supports this standard interface to controller data. The communications mechanism uses standard TCP/IP on the CPE's Embedded Ethernet port. Before getting started with the OPC UA server, you will want to have an OPC UA client (PAC CIMPLICITY HMI, for example) to connect to the OPC UA server.

The following is a high-level list of activities and functionality that is important to understand to startup and use the OPC UA server.

- OPC UA Certificate Security
- Application Logic to Control the OPC UA Server
- Connect OPC UA Client to the OPC UA Server
- OPC UA Client Authentication Settings
- OPC UA Address Space
- Publish Application Variables to OPC UA Address Space
- OPC UA Server Information in Address Space
- OPC UA Automatic Restart Function
- OPC UA Server Certificates

## 11.1 Model Comparison for OPC UA Server Capabilities

PACSystems Ethernet interfaces support OPC UA server capabilities. The values in the table below specify the maximum number of simultaneous OPC UA server connections.

	CPE302	CPE305	CPE310	CPE330	CPE400	CPL410	CPE100	CPE115	ETM001-jx	ETM001-Kxxx
OPC UA Server Concurrent Sessions	5	5	5	5	5	5	2	2	N/A	N/A

## 11.2 OPC UA Certificate Security

Security is an integral part of OPC UA technology. An OPC UA server provides a set of services that allow clients to make a secure connection to the server. Once a secure connection is created, messages between the client and the server can be encrypted and signed. The establishment of a secure connection is done using public-key cryptography.

Public key cryptography uses a paired public and private key to establish a secure channel. The public key is distributed to applications utilizing a certificate. A certificate is an electronic document with information affirmed by a trusted party.

The following sections describe how to configure your system to take advantage of OPC UA certificate security.

### Methods to Control the OPC UA Server

There are two primary methods to control the OPC UA Server. The user can use PAC Machine Edition (PME) or application logic to control the OPC UA server. PME is the preferred way to control the OPC UA server.

## 11.3 Controlling the OPC UA Server with PAC Machine Edition

Administrators can execute OPC UA server commands from PME. **To begin, the PLC must be online and connected.** To go online, open the *Navigator* pane and select the PLC. Right-click the PLC and select *Go Online*. Once the PLC is online, select *Connect* from the *Target* menu.

With the PLC online and the target network element selected, click the menu item *Programmer Mode*.

With the PLC in Programmer mode, right-click the PLC and hover on *Online Commands* to reveal the available OPC UA commands. (Command availability will be dependent on the server's status.)

The administrator can execute the following OPC UA commands from PME:

- Start Server
- Stop Server
- Clear Server
- Restart Server
- Activate Provisioning Mode
- De-Activate Provisioning Mode

**Note:** When both PME 9.8 SIM 5 and PLC Firmware 10.10 (or greater) are in use, configuring a redundant system will automatically enable non-transparent redundancy for OPC UA on the PLCs. In a non-transparently redundant pair configuration, the client is responsible for connecting to the two systems and swapping in the event of a failure.

### Privilege Levels Required for OPC UA Commands

For added security, OPC UA commands are regulated by a four-level privilege hierarchy. The commands are documented as follows:

OPC UA Server Command Privileges	
Server Status Bar	No Privileges Required
Start Server	Level 3 or Higher
Stop Server	Level 3 or Higher
Server Restart	Level 3 or Higher
Clear	Level 4
Activate Provisioning Mode	Level 4
Deactivate Provisioning Mode	Level 4

### OPC UA Server Commands

To execute function-based commands to the OPC UA server, the user can find the same command list under *Online Commands*. Alternatively, the user can select the following commands from the *Server Commands* button located under the *OPC UA* ribbon menu:

**Start:** This command starts the OPC UA Server. If the OPC UA server configuration files and certificates have been cleared or have not yet been generated, they are generated when the server starts. If previous configuration files and server certificates exist, they are used without

change. The server startup process also adds all published variables stored on the controller to the server's address space, up to the variable and element count limit.

**Stop:** This command stops the OPC UA Server on the controller. It does not remove or clear the configuration files.

**Restart:** This command stops and then restarts the OPC UA server on a target.

**Clear:** This command clears the configuration files and certificates used by the OPC UA server on the controller. Beginning with PACSystems firmware version 10.10 or later, if an OPC UA log file was saved during a fatal PLC error this action also will clear that log from the PLC.

**Note:** For more information on the *fatalinfo opc* command, please see GFK-2225, PACSystems TCP/IP Ethernet Communications Station Manager User Manual.

**Activate Provisioning Mode:** Selecting *Activate Provisioning Mode* will display a countdown timer in the Status Bar in the bottom-right corner of PME. Provisioning Mode will only remain activate for 60 minutes or until a single certificate connection has been completed. Subsequent certificate connections will require the administrator to activate Provisioning Mode each time.

**Note:** Selecting *Activate Provisioning Mode* will start a stopped OPC UA server.

**De-Activate Provisioning Mode:** Selecting *De-Activate Provisioning Mode* will take the OPC UA server out of Provisioning Mode, cancel the timer, and refuse certificate connection requests.

**OPC UA Status Bar:** The OPC UA Server status is displayed in a Status Bar in the lower-right corner of PME. The timer will be in a *HH:SS* format.

## Setting OPC UA Server in Provisioning Mode

To place the OPC UA server in Provisioning Mode, the user must have Level 4 privileges. The OPC UA server can be set into Provisioning mode using the *Navigator* pane or using upper menu navigation. With *Programmer Mode* enabled right-click on the PLC in the *Navigator* pane and select *Online Commands* to reveal *Activate Provisioning Mode* and *De-Activate Provisioning Mode*.

Alternatively, with PME in *Programmer Mode*, the user can select the menu button commands under *OPC UA* in the ribbon bar. Selecting the *OPC UA menu* will reveal two-button command categories: *Certificate Provisioning* and *Server Commands*. From this menu button, the user can enable *Provisioning Mode* for the OPC UA server.

## OPC UA Server Certificates

OPC UA client/server connections exchange digital certificates during the connection process. Server certificates follow the process outlined below:

### OPC UA Initial Connection Process

1. The OPC UA server generates a self-signed certificate for the connection process.
2. The OPC UA certificate will include application-specific information within the certificate, such as Target Name and the controller's TCP/IP address. (If this information changes, the certificates should be cleared and regenerated.)
3. The OPC UA server certificates will be stored internally on the controller's non-volatile storage and will be retained through power cycles, clearing of memory and configuration from the programmer, and clearing of flash storage from the programmer.
4. The CONFIG\_STATUS service request returns a bitmask to indicate if the certificates exist on the target, or if they are currently cleared.

### OPC UA Servers Certificates Already Cleared

1. If the OPC UA server is started with the certificates cleared, new certificates are generated during the startup of the OPC UA server.
2. If the OPC UA server is started with certificates already on the target, then those existing certificates are used, and new ones are not generated.

### OPC UA Server Certificates Need to be Cleared

1. If certificates currently exist on the target and need to be cleared, the OPC UA server must be stopped, and then the CLEAR service request can be used to clear the certificates on the controller.
2. When a CLEAR is used to clear the certificates, the certificates are permanently deleted and cannot be restored.
3. Once this occurs, new certificates must be generated.

**NOTE:** The CLEAR service request will not pass power if it is performed with the OPC UA server running.

4. To assist with checking the status of and clearing certificates, the OPC UA subroutine previously discussed offers a **ClrSvr** input that might be used to clear the server certificates any time the server is stopped.

## CAUTION

After successfully provisioning security, the OPC UA server must be restarted.

## OPC UA Performance Considerations

The OPC UA Server supports 5 concurrent clients and 12,500 addressable elements. Care is suggested both approaching and operating at the max-max condition: 5 clients accessing the full complement of 12,500 elements simultaneously.

The OPC UA Server has been designed and implemented to support the maximum number of clients reading the maximum number of OPC UA elements. However, at maximum load OPC UA clients may experience degraded performance, evidenced by slower than usual response times when browsing, reading, and writing addressable elements.

If OPC UA Server response times are not adequate for your application, consider implementing one or more of the following mitigation strategies:

**Controller Communications Window** – OPC UA traffic is processed in the controller communications window. If the window's duration is inadequate for the amount of processing required to service multiple client requests, increasing the comm window will improve response times by the PLC.

**OPC UA Subscriptions** – OPC UA defines a subscription model for the server to transmit data only when an element's value updates, rather than having a client continuously poll the server to read an element that is otherwise unchanged in value. Determine whether your application can use a subscription to collect data from the PLC.

**OPC UA Aggregating Server** – An aggregating server can both allow multiple clients to connect as a single instance and combine read-requests for the same element(s) into one message.

For more information on OPC UA Performance, please see *OPC UA Server Performance Estimation* in Section in Section 11.

## Sessions and Subscriptions for OPC UA

There may be up to five concurrent sessions<sup>27</sup> for the OPC UA server. Across these sessions, there may be up to ten concurrent subscriptions of variables. The subscription limit is shared across all available sessions.

For CPE100 and CPE115 there can be 2 concurrent sessions. Across these sessions, there may be 4 concurrent subscriptions of variables.

### CAUTION

When a secure connection is created, messages between the client and the server are encrypted and signed. Once the security server has a CA-signed certificate, the security server will **only** accept connections from applications that are signed by the same CA.

## 11.4 Log Level Settings

Beginning with PACSystems firmware version 10.10 or later, OPC UA servers have logging capabilities to aid in troubleshooting and debugging OPC UA configuration problems. These logs are disabled by default but can be enabled in the PME Hardware Configuration.

**Note:** Enabling OPC UA server logs will degrade the performance of the OPC UA server.

### Log Level Settings and Log File Length

There are two logging settings available and, in both cases, greater level log settings include all the lower level logs above that level. (For example, the permissions for the *Informational* log level will also include the permissions for *Warnings* and *Errors* log levels as well.)

The total log file length available is limited to 4000 lines. When these 4000 lines of logging information are exceeded, the file is deleted and started over empty. This rollover to a new log file happens automatically without user intervention when the log is full. When this occurs the prior lines of the log are lost, but there is new space for additional logs and the system memory will not be filled with logging info. Care should be taken to set a logging level high enough to troubleshoot the given problem, but also low enough so that the logs do not roll-over too fast to capture the information necessary.

---

<sup>27</sup> The CPE100/115 support up to two concurrent sessions. Across these sessions only a maximum of 4 concurrent subscriptions of variables.

**Figure 69: OPC UA Parameters in Hardware Configuration**

Settings	Scan	Memory	Redundancy	Transfer List	Faults	Scan Sets	Modbus TCP Address Map	Power Consumption	Access Control	SNTP	Time	OPC UA
<b>Parameters</b>												
Server Enabled												True
Automatic Start												True
UTC Offset												-5:00 Eastern Standard Time
Application Type												Server
Application Name												OPC-UA-Test
Product URI												http://ge-ip.com/PACSystems/Rx3i
Server Capabilities												N/A
Certificate Expiration Date Checking												Ignore
Certificate Debug Commands												Disabled
SDK Log Level												Informational
Stack Log Level												Warning

## SDK Log Level

The SDK Log Level is for logs related to the application layer of the OPC UA protocol and has the following options:

- Disabled – Default and recommended for production
- Errors
- Warnings
- Informational – Recommended for most troubleshooting
- Interface Call
- Constructor/Destructor
- Program Flow

## Stack Log Level

The Stack Log Level is for logs related to the low-level OPC UA protocol stack and has the following options:

- Disabled (default and recommended for production)
- Error
- Warning (recommended for most troubleshooting)
- System
- Information
- Debug
- Content
- All

## Reading the Log File

Once a configuration is stored with logging enabled, logging starts automatically and stays on without intervention. To obtain the contents of the current log file use the *opc log* or *opc all* station manager commands. NOTE: The log file obtained by the *opc log* or *opc all* commands is current data from the current OPC UA server run.

Refer to the *PACSystems TCP/IP Ethernet Communications Station Manager User Manual*, GFK-2225, for information on how to use the station manager *opc* commands.

## Clearing the OPC UA Log Files

The current log files will remain until the next OPC UA server start or power on. The log files are cleared every CPU power-on and every OPC UA server start. The logs are, therefore, not retained through a power-off of the PACSystems CPU, nor are they retained for a subsequent OPC UA server start.

## OPC UA Log File During Fatal Error

If the PACSystems CPU experiences a fatal error of any kind while OPC UA logging is enabled, the current OPC UA log will be retained in internal memory and can be read out later after reboot using the *fatalinfo opc* command.

This file saved during a fatal error handling will be retained through power cycles and PME PLC configuration clear commands. When the OPC UA configuration is cleared with the *OPC UA clear server* command either from PME or by Service Request; however, the retained OPC UA log file will be cleared.

**NOTE:** The log file obtained by the *fatalinfo opc* command will contain old data from a previous run; however, it may be useful in some situations when attempting to identify the source of a problem that has occurred.

Refer to the *PACSystems TCP/IP Ethernet Communications Station Manager User Manual*, GFK-2225, for more information on how to use the station manager *fatalinfo* commands.



## 11.5 Application Logic to Control the OPC UA Server

The OPC UA server can also be controlled utilizing a service request (SVC REQ function block). The service request allows you to start, stop, restart, clear, and query the status of OPC-specific information.

### OPC UA Server Service Request

There is one service request dedicated to the PACSystems OPC UA server. This is service request 130, protocol 0x0001. The OPC UA server service request contains a number of sub-functions to accomplish different tasks.

SERVICE\_REQUEST 130 protocols:

Sub-function	Code
OPC UA SERVER	16#0001
<b>Note:</b> All other protocol codes are reserved, and if used, the SVC_REQ function will not	

SERVICE\_REQUEST 130, protocol 1, sub-functions:

Sub-function	Code
START	16#0000
STOP	16#0001
CLEAR	16#0002
SERVER_STATUS	16#0003
CONFIG_STATUS	16#0004
RESTART	16#0005
PROVISIONING_STATUS	16#0006
RESERVED	16#0007
SET_TIME_MS_LIMIT_PER_SWEEP	16#0008
GET_TIME_MS_LIMIT_PER_SWEEP	16#0009

### OPC UA Server – Service Request – START

This function starts the OPC UA server. If the OPC UA server configuration files and certificates have been cleared or have not yet been generated, they are generated when the server starts. If previous configuration files and server certificates exist, they are used without change. The server startup process also adds all published variables stored on the controller to the server's address space, up to the variable and element count limit.

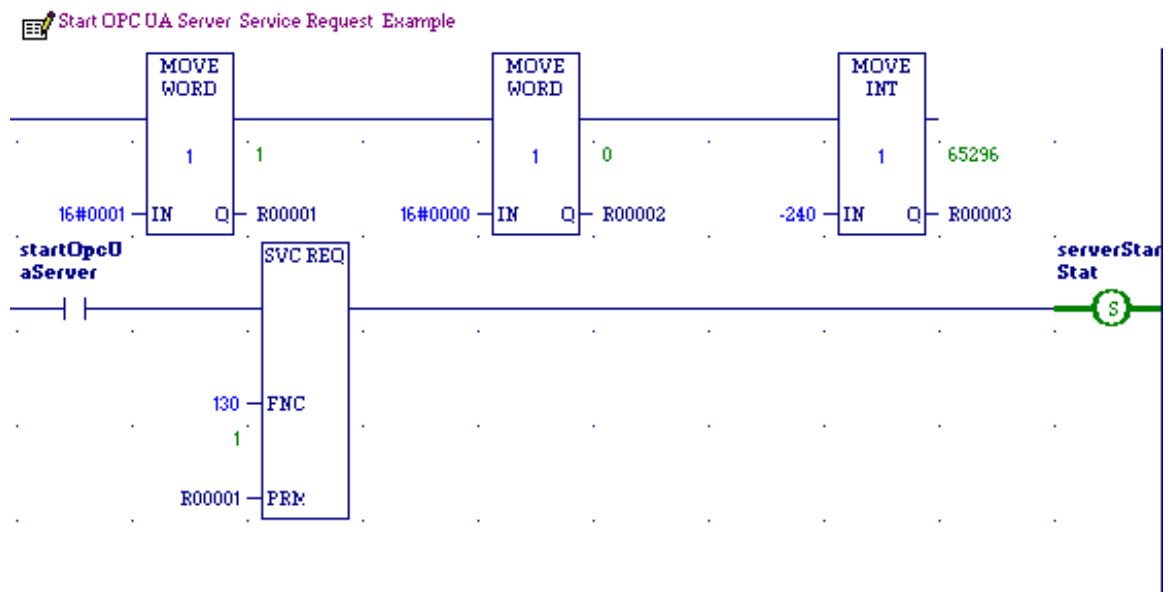
**Note:** The PACSystems OPC UA server takes up to two minutes to start if the server's configuration files and certificates need to be generated. This occurs the first time the server is started. The OPC UA server can take up to 30 seconds to start on subsequent starts. The OPC UA server startup time can be improved by tuning the system as described

Parameters for the START function service request function block are:

Parameter	Summary	Data Direction (LD perspective)
16#0001	OPC UA protocol	IN
16#0000	START request	IN
-1440 to 1440	Time Zone Offset	IN

If the SVC\_REQ does not pass power, the operation did not complete. The time zone offset adjusts the OPC UA server time zone. The Controller's Time of Day (TOD) clock must be synchronized to local time and the time zone offset is your location's offset relative to Universal Time Coordinated (UTC, formerly known as Greenwich Mean Time or GMT).

Figure 70: Example of Start OPC UA Server Service Request



## OPC UA Server – Service Request – STOP

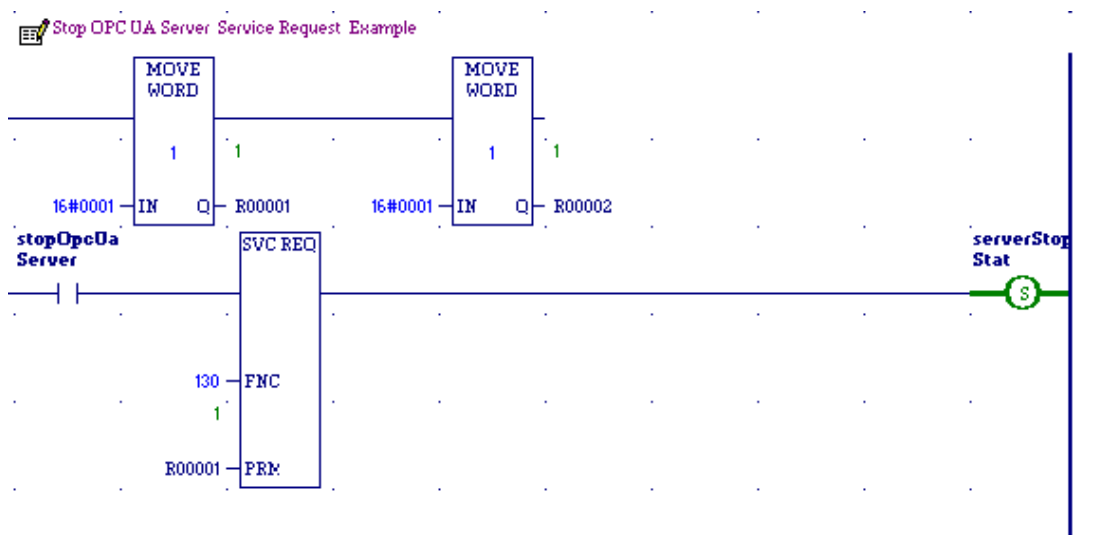
This function stops the OPC UA Server on the controller. It does not remove or clear the configuration files.

Parameters for the STOP function service request are:

Parameter	Summary	Data Direction (LD perspective)
16#0001	OPC UA protocol	IN
16#0001	STOP request	IN

The use of the STOP sub-function code in a ladder diagram is illustrated in the following example. If the SVC\_REQ does not pass power, the operation did not complete.

Figure 71: Example of Stop OPC UA Server Service Request



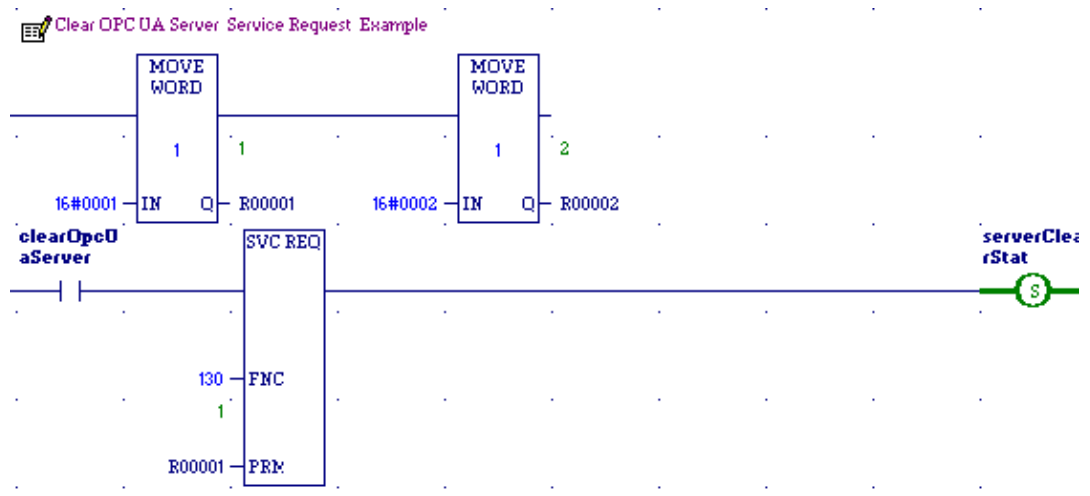
## OPC UA Server – Service Request – CLEAR

This function clears the configuration files and certificates used by the OPC UA Server on the controller. Starting with PACSystems firmware version 10.10 or later, if an OPC UA log file was saved during a fatal PLC error (see ‘fatalinfo opc’ station manager command) this action also will clear that log from the PLC. If the SVC\_REQ does not pass power, the operation did not complete.

Parameters for the CLEAR function service request are:

Parameter	Summary	Data Direction (LD perspective)
16#0001	OPC UA protocol	IN
16#0002	CLEAR request	IN

Figure 72: Example of Clear OPC UA Server Service Request



## OPC UA Server – Service Request – RESTART

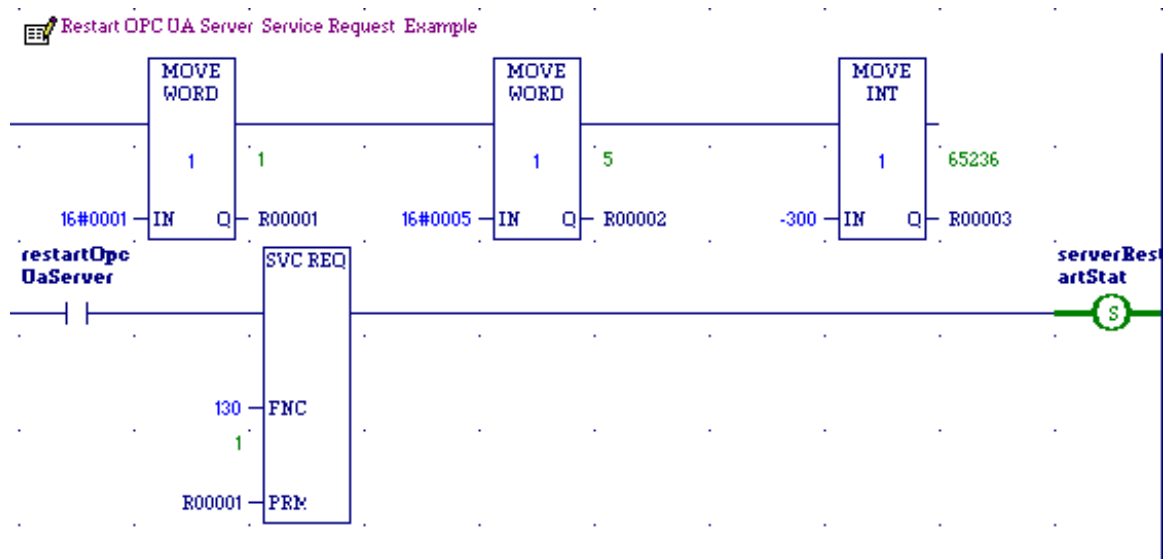
This function stops and then restarts the OPC UA server on a target.

**Note:** The PACSystems OPC UA server takes up to two minutes to start if the server’s configuration files and certificates need to be generated. This occurs the first time the server is started. The OPC UA server can take up to 30 seconds to start on subsequent starts. The OPC UA server startup time can be improved by tuning the system as described on *OPC UA Server – Service Request – START* in Section 11.5.

Parameter	Summary	Data Direction (LD perspective)
16#0001	OPC UA protocol	IN
16#0005	RESTART request	IN
-1440 to 1440	Time Zone Offset	IN

If the SVC\_REQ does not pass power, the operation did not complete. The time zone offset adjusts the OPC UA server time zone. The Controller’s Time of Day clock must be synchronized to ‘local’ time and the time zone offset is your location’s offset relative to UTC time.

Figure 73: Example of Restart OPC UA Server Request



## OPC UA Server – Service Request – SERVER\_STATUS

The SERVER\_STATUS sub-function code can be used to obtain info about the server status.

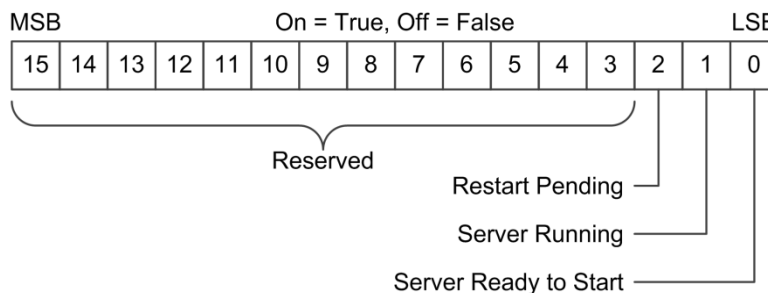
Parameters for the SERVER\_STATUS function service request are:

Parameter	Summary	Data Direction (LD perspective)
16#0001	OPC UA protocol	IN
16#0003	SERVER_STATUS request	IN
0000 0000 0000 0000	Server Status Response – bitmask (see below)	OUT

If the SVC\_REQ does not pass power, the operation did not complete.

The SERVER\_STATUS word bit definitions are displayed below.

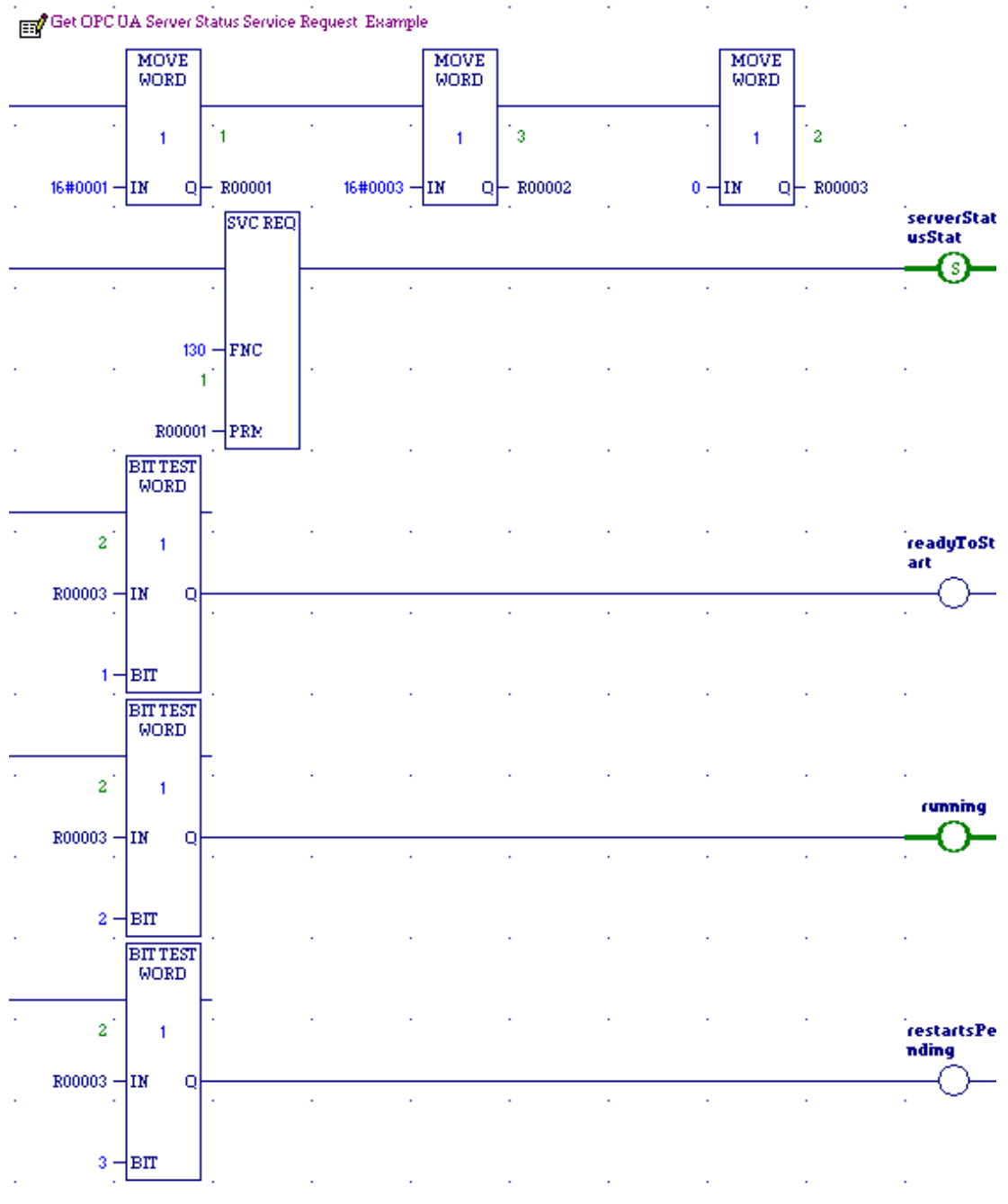
**Figure 74: SERVER\_STATUS Word bit definitions**



**Note:** The PACSystems OPC UA server takes up to two minutes to start if the server’s configuration files and certificates need to be generated. This occurs the first time the server is started. The OPC UA server can take up to 30 seconds to start on subsequent starts.

Once the OPC UA server is running, the OPC UA *SERVER\_STATUS* service request returns 0010h *OPC UA Server Running*.

Figure 75: Example of Get OPC UA Server Status Service Request



## OPC UA Server – Service Request – CONFIG\_STATUS

The CONFIG\_STATUS sub-function code can be used to obtain info about the server status.

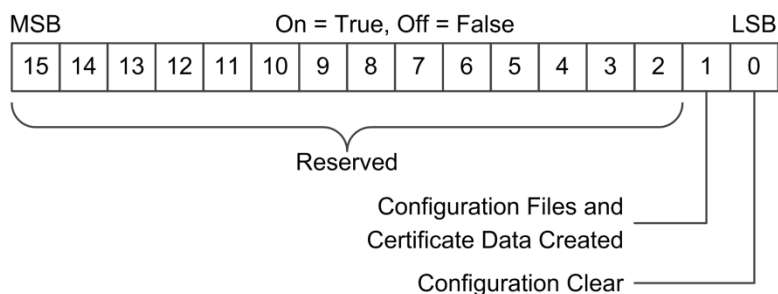
Parameters for the SERVER\_STATUS function service request are:

Parameter	Summary	Data Direction (LD perspective)
16#0001	OPC UA protocol	IN
16#0004	CONFIG_STATUS request	IN
0000 0000 0000 0000	Config Status Response - bitmask	OUT

If the SVC\_REQ does not pass power, the operation did not complete.

The CONFIG\_STATUS word bit definitions are displayed below.

**Figure 76: CONFIG\_STATUS Word bit definitions**





## OPC UA Server – Service Request – PROVISIONING\_STATUS

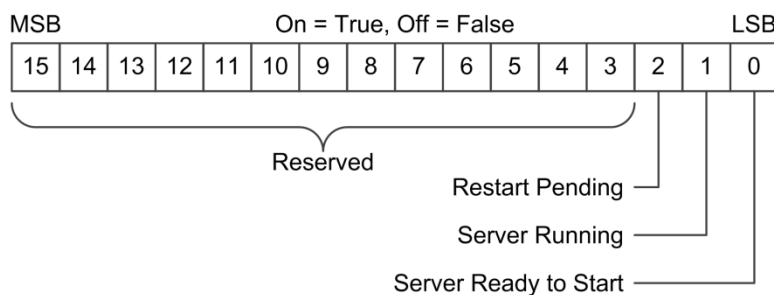
The PROVISIONING\_STATUS sub-function code can be used to obtain info about the server status.

Parameters for the PROVISIONING\_STATUS function service request are:

Parameter	Summary	Data Direction (LD perspective)
16#0001	OPC UA Protocol	IN
16#0006	Request Provisioning Status	IN
0000 0000 0000 0000	Server Status – bitmask	OUT
0000 0000 0000 0000	Config Status - bitmask	OUT
0 – 60 minutes	Provisioning Mode Time Remaining (seconds)	OUT

The PROVISIONING\_STATUS word bit definitions are displayed below.

**Figure 77: Server Status Definition**



**Figure 78: Config Status Definition**

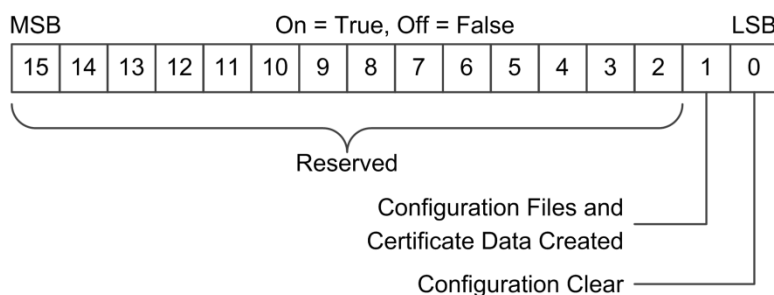
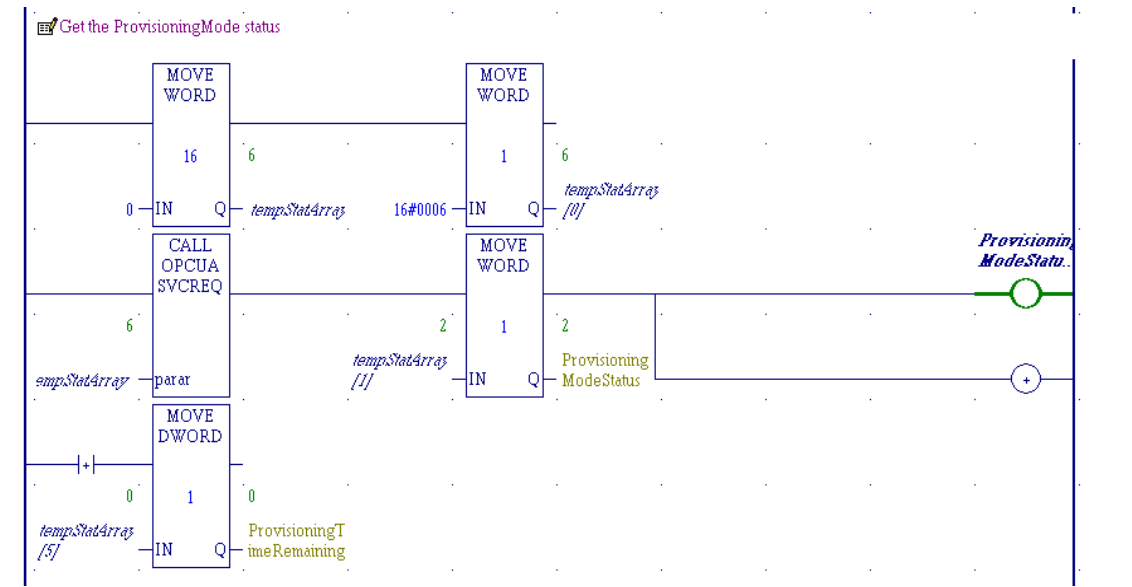


Figure 79: Example of Get Provisioning Mode Status Service Request



## OPC UA Server – Service Request – SET\_TIME\_MS\_LIMIT\_PER\_SWEEP

On PACSystems firmware version 9.90 or later, the SET\_TIME\_MS\_LIMIT\_PER\_SWEEP sub-function code can be used to set the amount of time allowed per sweep for OPC UA task execution. This setting is only used when the PACSystems sweep mode is set to *Normal Sweep Mode* and is not used when the sweep mode is set to *Constant Sweep Mode* or *Constant Window Mode*. The limit set here will not cause the Watchdog Timer to trip or other CPU sweep times to be missed, it is only used as a limit to prevent OPC UA tasks from running for a long time if the settings are such that they would. The default value of this setting for all PACSystems CPUs is 100 milliseconds. The current value for this setting can be read using the GET\_TIME\_MS\_LIMIT\_PER\_SWEEP request.

When the Sweep Mode is set to *Normal* and the Controller Communication Window Mode is set to *Limited* or *Complete*, once OPC UA tasks start to execute for that sweep, they will be allowed to execute for no longer than this time limit set, even if the Controller Communication Window would otherwise be allowed to run longer. If it is required to run OPC UA server tasks for more than the default 100 ms per sweep, extend the SET\_TIME\_MS\_LIMIT\_PER\_SWEEP function to an appropriate value greater than 100 ms. (Alternatively, users may enable *Constant Window* or *Constant Sweep Modes*.)

For example, if the Controller Communication Window Timer is set to 200 ms and the OPC UA task execution sweep time limit is set to 100 ms (the default), then if OPC UA tasks were executing for 100 ms the Controller Communication Window will close and the sweep will advance to the Backplane Communication Window phase. This would occur even if there was time remaining in the Controller Communication Window for that sweep.

In another example, if the Controller Communication Window Timer is set to 10 ms (the default) and the OPC UA task execution sweep time limit is set to 100 ms (also the default) then the OPC UA tasks would be limited to executing the lower of the two limits of just 10 ms per sweep. In this case, if more execution time is desired to be allowed for OPC UA, then the Controller Communication Window Timer limit should be increased and the OPC UA task execution sweep time limit would not need to be adjusted.

Note: Additional examples are provided in *Determine the Controller Communication Time Limit* in Section 11.7

### CAUTION

Should the active PLC fail or be put into STOP mode, the redundant PLCs may perform a role switch, transferring the active unit status to the previous backup unit. This action transfers the backup unit's status to the previous active unit.

In this scenario, CIMPLICITY will begin to update its points with values from the new active unit at the time of the role switch. However, if a point has not changed on the active unit since before the role switch, the CIMPLICITY project will not update its displayed point value. Some values may remain effectively frozen until the unit changes the variable's value.

**The value and its timestamp will continue to show the values obtained from the previously active unit until the variable changes on the new active unit.**

---

## Failover Performance

Performance of failovers in OPC UA Non-Transparent Server Redundancy will vary depending on the system configuration and type of failure mode. Users should expect greater OPC UA server failover delays with *Cold* and *Warm* failover modes because the OPC UA client must do extra work after the failure to reconnect and/or enable sampling. The exact timing for this to occur will be specific to the load on the PLCs, the hardware specifications of the controller, and the network conditions.

Since the OPC UA client subscribes to the Service Level variable in the OPC UA address space and to obtain the current value when a change takes place requires a publishing period of that changed data, the performance of how long this operation will take can be identified by measuring the performance of the specific configuration for variable updates as described in Section 11.7, *OPC UA Service Performance and Estimation* below. The worst-case system subscription update would be the upper bound of how much time it takes for a Service Level change to be set on the server, published over the network, and obtained by the client.

When a network failure or PLC failure that resulted in complete loss of network communication (such as a power loss or watchdog timeout expiration) is detected, the OPC UA Client immediately attempts to connect to the standby server. It can take up to a full connection timeout period for the client to detect the server was lost due to network communication interruption. The specific name for this connection timeout may vary by the type of OPC UA client being used. Please consult the documentation for the specific OPC UA client to determine the value of its network communication timeout setting.

OPC UA Service Performance and Estimation.

Parameters for the SET\_TIME\_MS\_LIMIT\_PER\_SWEEP function service request are:

Parameter	Summary	Data Direction (LD perspective)
16#0001	OPC UA protocol	IN
16#0008	SET_TIME_MS_LIMIT_PER_SWEEP request	IN
1 to 2550	New OPC UA task sweep time limit in milliseconds.	IN

Note: If the SVC\_REQ does not pass Power, the operation did not complete.

## OPC UA Server – Service Request – GET\_TIME\_MS\_LIMIT\_PER\_SWEEP

Starting with PACSystems firmware version 9.90 or later the GET\_TIME\_MS\_LIMIT\_PER\_SWEEP sub-function code can be used to get the current amount of time allowed per sweep for OPC UA task execution. This setting is only used when the PACSystems sweep mode is set to Normal sweep mode and is not used when the sweep mode is set to Constant Sweep or Constant Window mode. The limit set here will not cause the Watchdog Timer to trip or other CPU sweep times to be missed, it is only used as a limit to prevent OPC UA tasks from running for a long time if the settings are such that they could. The default value of this setting for all PACSystems CPUs is 100 milliseconds, and it can be changed by using the SET\_TIME\_MS\_LIMIT\_PER\_SWEEP request.

Parameters for the GET\_TIME\_MS\_LIMIT\_PER\_SWEEP function service request are:



Parameter	Summary	Data Direction (LD perspective)
16#0001	OPC UA protocol	IN
16#0009	GET_TIME_MS_LIMIT_PER_SWEEP request	IN
1 to 2550	Current OPC UA task sweep time limit in milliseconds.	OUT

**Note:** If the SVC\_REQ does not pass power, the operation did not complete

## 11.6 Connect OPC UA Client to OPC UA Server

Once the OPC UA server is running, a client can connect to the server and browse the address space. The OPC UA server uses the OPC UA Binary protocol to communicate with the client. The OPC UA Binary connection strings take the base form displayed in (Figure 80).

**Figure 80: OPC UA Binary Connection String**

<code>opc.tcp://&lt;PACS IP Address&gt;:4840</code>	
	
<b>Specifies the OPC UA Binary Protocol</b>	<b>PACS Server TCP Port Number</b>

As an example, a connection string for the OPC UA server is constructed. To begin, the controller TCP/IP address of the embedded Ethernet port is needed. One method to find this information is to use the PME programmer. Open the controller's project and select the project top level in the *Project* tab of the *Navigator* window. From the *Inspector* window, scroll down to the *IP Address Entry* (see the screenshot below). From the figure, we can see the current IP address is *10.10.1.102*. For this example, the client's connection string for the controller is the following:

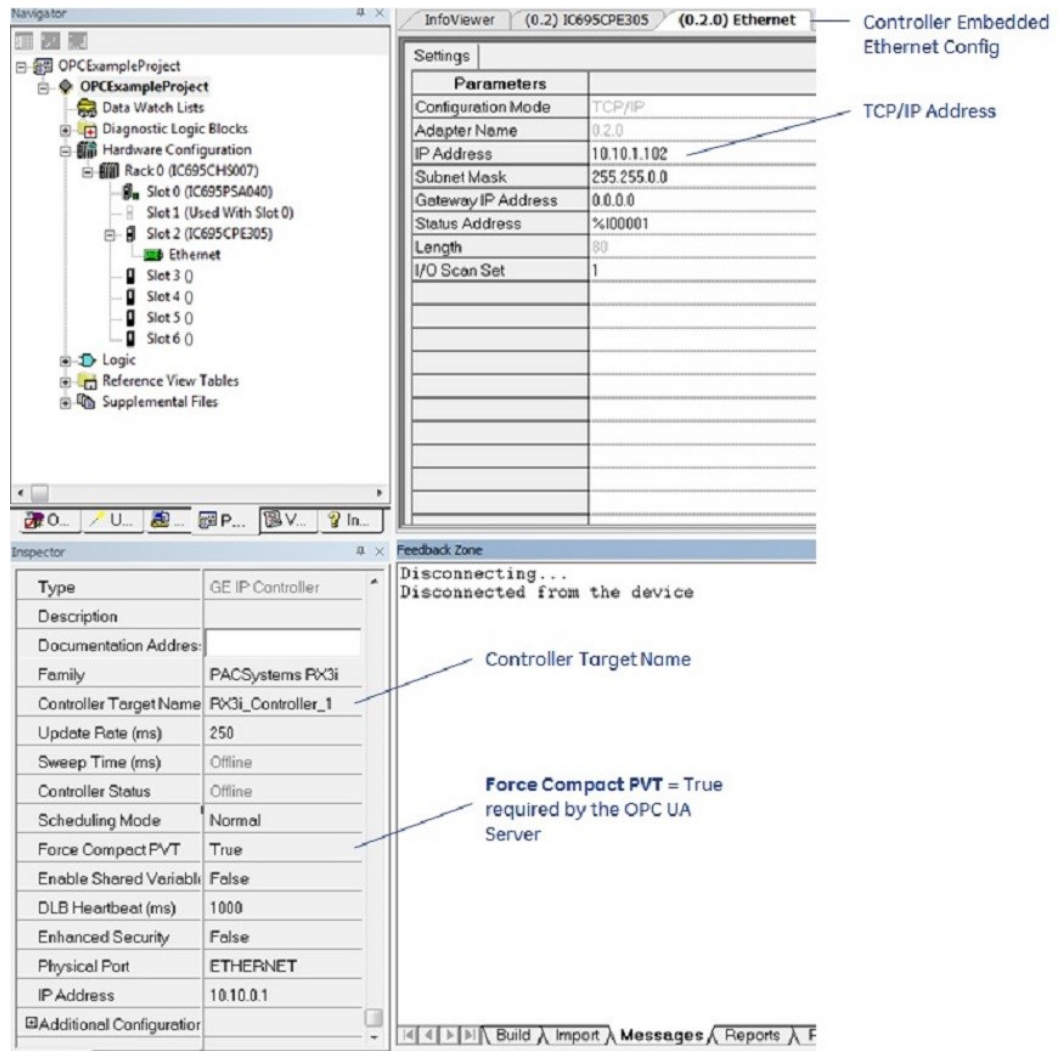
```
opc.tcp://10.10.1.102:4840
```

**Note:** The Redundant IP address cannot be used for OPC UA communication, the OPC UA server will not respond to requests sent to the PACSystems Controller's Redundant IP Address. For use of OPC UA in a redundant configuration the OPC UA Client must support the OPC UA Non-Transparent Redundancy specification.

The specification is available as "OPC Unified Architecture Specification," Release 1.04, November 22, 2017. Published by the OPC Foundation. At the time of this release, online versions of this specification are available at:

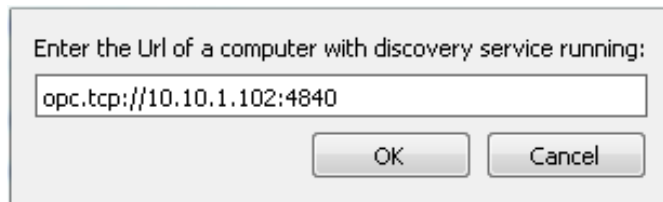
<https://reference.opcfoundation.org/v104/>

Figure 81: Project Inspector/Ethernet Config Window



In Figure 81, Force Compact PVT is set to true. This is the required setting for the OPC UA server. From the client-side, we can establish a connection by placing the above information into the connection string (see screenshot below using an OPC UA client). We can then connect to the OPC UA server.

Figure 82: OPC UA Server Client Connection String

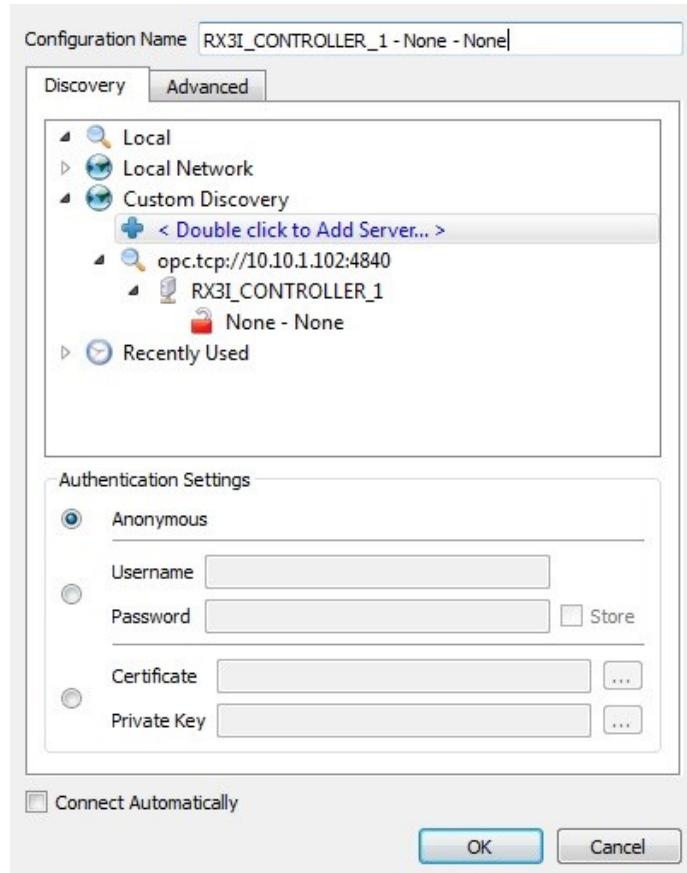


**Note:** The Client can see the Controller Target Name when connecting to the server.



The Controller Target Name is set within PME and is displayed in the screenshot above. A sample client connection can be seen below.

**Figure 83: OPC UA Client Connection Dialog**



**Note:** The RX3i's OPC UA server supports 5 clients. If 5 clients are connected, additional connection requests will be rejected by the OPC UA server.

## OPC UA Client Authentication Settings

OPC UA provides three authentication methods to logon to a server:

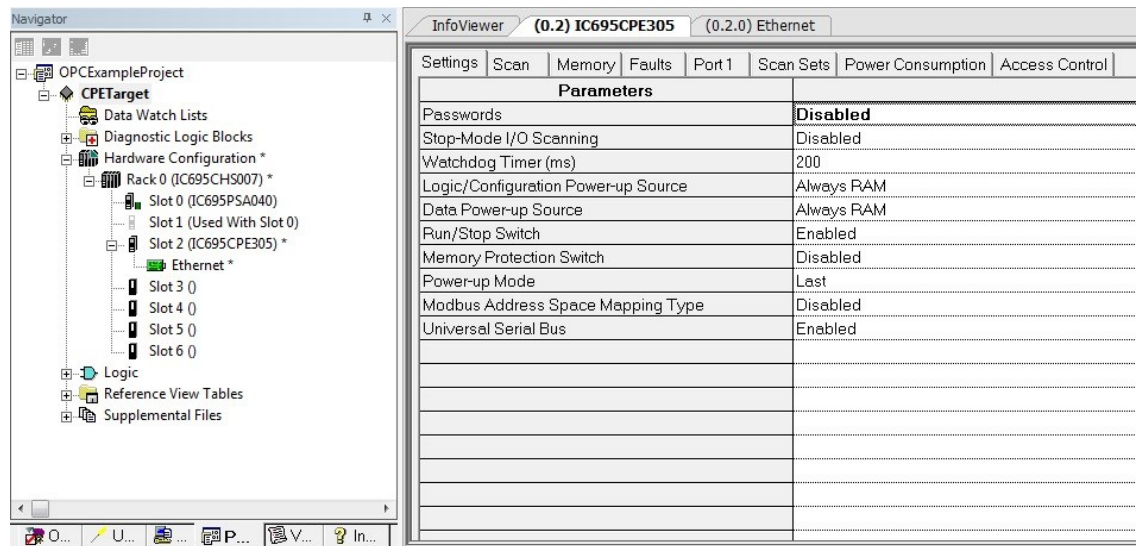
- Anonymous
- Username/Password

The OPC UA server supports Anonymous and Username/Password Authentication methods. PME controller project settings determine the Authentication method used by the OPC UA server.

## Anonymous Authentication

OPC UA server Anonymous Authentication can be enabled by disabling Controller passwords. PME is used to disabled controller passwords. To access this setting using PME, open the Controller *Hardware Configuration* with the *Project* tab within the *Navigator*, expand the *Hardware Configuration*, and select the controller. Double-click the controller tree node to access the controller-specific *Hardware Configuration* settings. Select the *Settings* tab, then set the *Passwords* parameter to *Disabled* (Figure 84).

Figure 84: PME Controller *Hardware Configuration* – Passwords Disabled

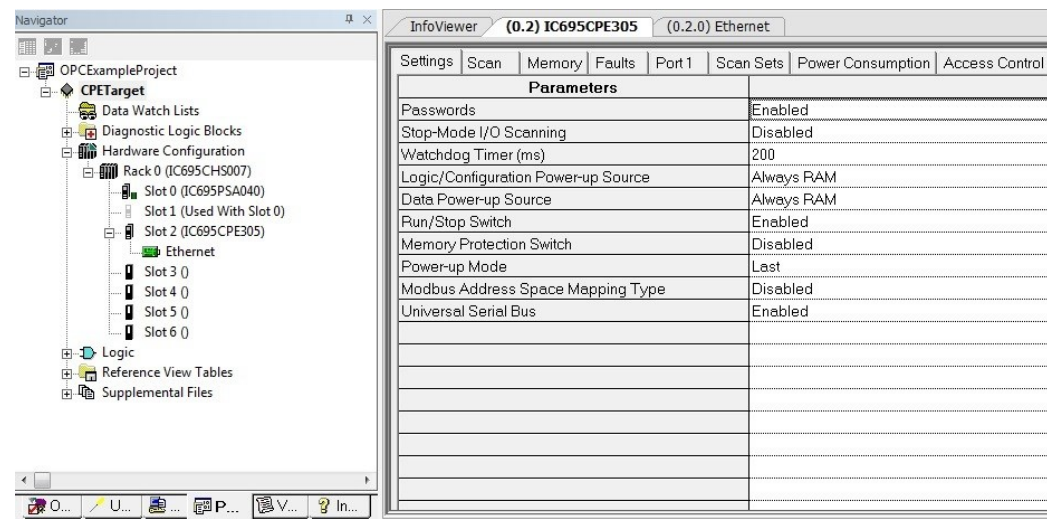


## Username/Password Authentication

OPC UA server Username/Password Authentication is enabled through RX3i controller passwords. PME is used to enable controller passwords. To access this setting using PME, open the RX3i *Hardware Configuration* in the *Project* tab within the *Navigator*, expand the *Hardware Configuration*, and select the controller. Double-click the controller tree node to access the controller-specific *Hardware Configuration* settings. Select the *Settings* tab, then set the Passwords parameter to *Enabled* (Figure 85).s

**Note:** Both the username and password are case-sensitive.

**Figure 85: PME Controller *Hardware Configuration* – Passwords Enabled**



The OPC UA server password is the same as the controller password. Controller passwords are set using the PME commands *Select Target* → *Online Commands* → *Show Status*, which opens the controller status dialog box. Select the *Protection* tab, click the *Passwords* button to set the passwords for the different access levels (see the screenshot below).

The OPC Server assigns usernames to the different access levels. The usernames that correspond to the different levels are as follows:

Level	OPC UA Username	Description
Level 4	OpcUserLevel4	Read/Write Published Variables – Additional Privileges Reserved for Future Use
Level 3	OpcUserLevel3	Read/Write Published Variables – Additional Privileges Reserved for Future Use
Level 2	OpcUserLevel2	Read/Write Published Variables

**For example:**

Level 2 password = *MyLevel2Password*

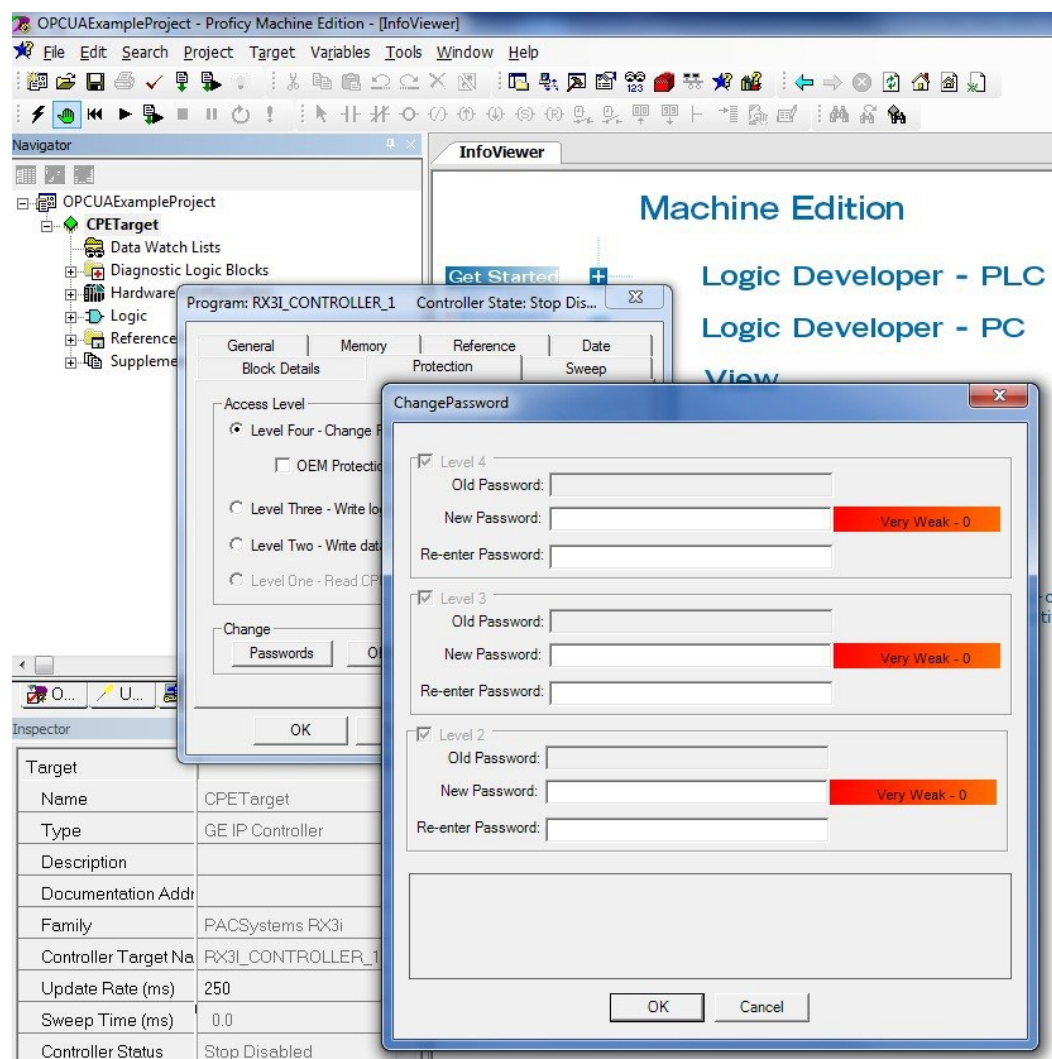
The OPC UA Client would use the following username/password to establish a connection.

Username = OpcUserLevel2

Password = MyLevel2Password

Please reference the PME documentation for additional details regarding setting passwords and the privileges assigned to different levels.

**Figure 86: PME Online Command to Set Passwords**



## OPC UA Security Settings

OPC UA clients typically have two settings for Security: Security Policy and Message Security Mode. The following settings are recommended:

**Security Policy:** Basic256Sha256

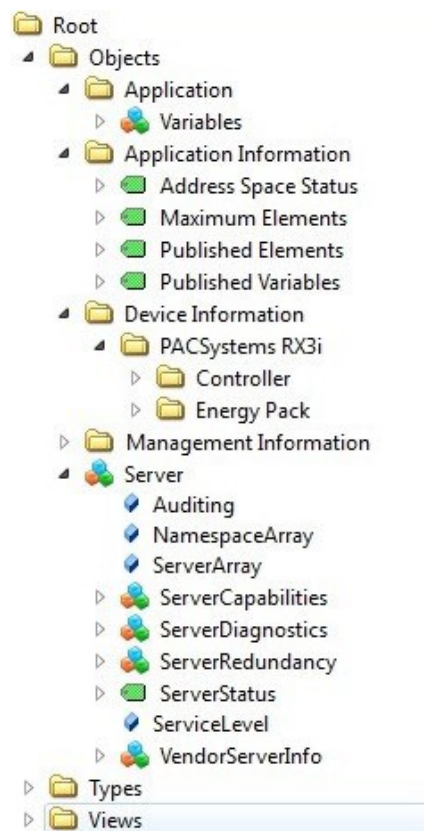
**Message Security Mode:** Sign and Encrypt

**Note:** None/None is supported, but not recommended as it does not provide any cybersecurity protection.

## OPC UA Address Space

The OPC UA address space contains information about the server and its application. An OPC UA client browses the address space to determine server functionality and the controller application variables available from the server. An example client address space view is displayed below.

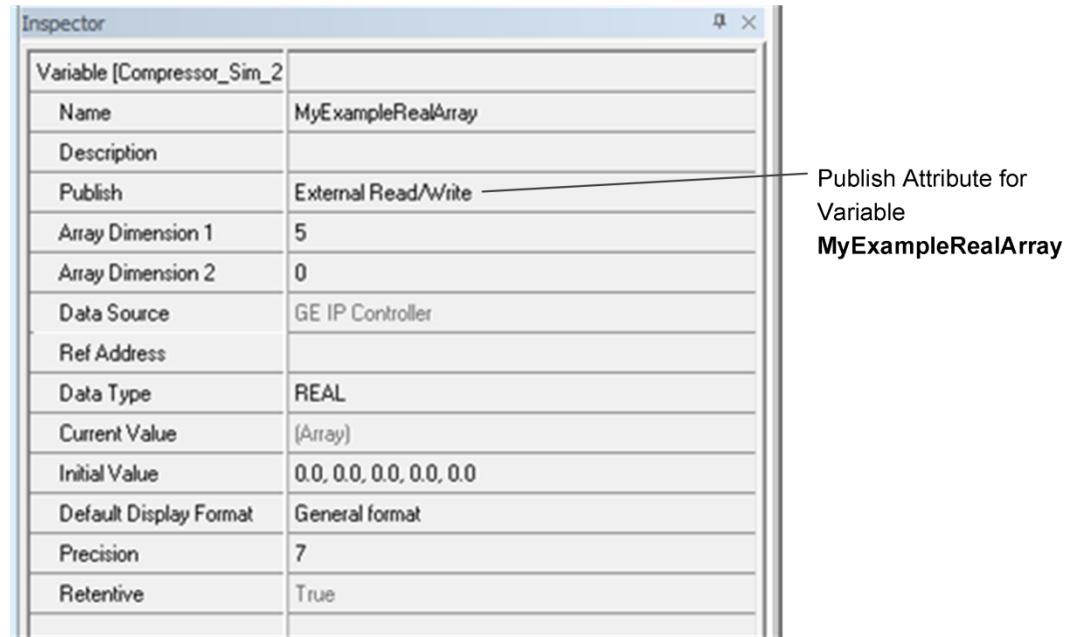
**Figure 87: Example OPC UA Address Space**



## Publish Application Variables to OPC UA Address Space

PME can select application variables to include in the OPC UA address space. This is done utilizing the variable’s publish attribute. The publish attribute is accessed using the variable *Inspector* within PME. The PME variable *Inspector* is displayed in the screenshot below for reference.

**Figure 88: PME Variable Inspector**



The available PME Publish attribute selections are as follows as they apply to the OPC UA server:

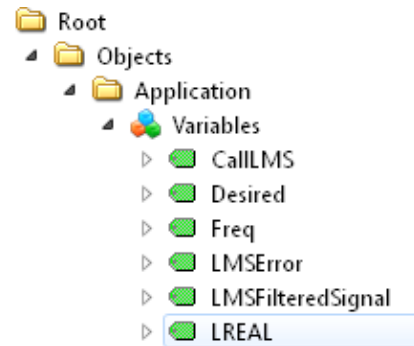
Selection	Description – OPC UA Server Specific Usage
False	Variable is not published to OPC UA Address Space
Internal	Variable is not published to OPC UA Address Space
External Read/Write	Publish variable to OPC UA address space and allow the OPC UA client Read and Write Access
External ReadOnly	Publish variable to OPC UA address space and allow the OPC UA client Read Access only

**Note:** The Controller’s OPC UA address space supports 12,500 addressable elements. If more than 12,500 addressable elements are published, only the first 12,500 (listed alphabetically) will be made available in the OPC UA address space. Each index of a variable array counts as a unique addressable element. CPE100 and CPE115 controller’s OPC UA supports 1000 addressable elements. The OPC UA server regenerates the address space only at startup. Thus, adding a new variable or modifying an existing variable publish attribute requires the server to perform the startup sequence. In most cases, the controller performs this function for the user. Please reference the section entitled *OPC UA Server – Service Request – RESTART* for additional details on server restart functionality.

The published application variable is accessible by the client. One method is to browse the address space, opening the *Application* node displayed in the screenshot below.

**Note:** If the server address space has been updated and the client is currently connected to the server, you may need to refresh the client view. Depending on the client implementation, this may require the client to re-browse the address space.

**Figure 89: Application Variable Address Space**

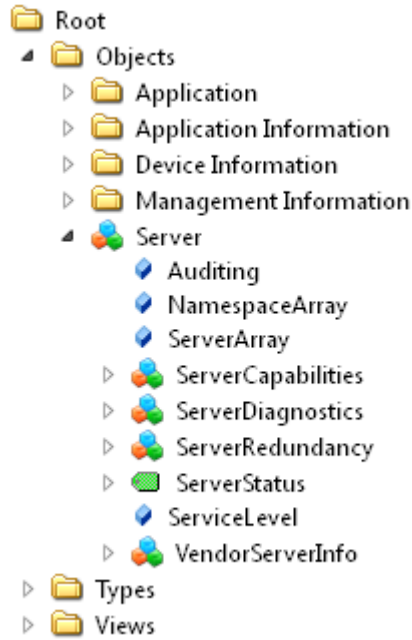


## OPC UA Server Information in Address Space

OPC UA servers allow clients to self-discover the OPC UA servers and server capabilities. Thus, there is significant information on both the application variables themselves and the server contained within the address space. The following highlights some of these attributes. Additional information regarding the address space can be found on the OPC Foundation website and in its publications.

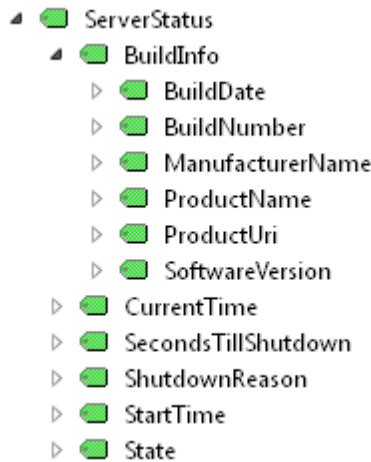
General Server information is contained under the Server node in the address space (see Figure 90).

Figure 90: OPC UA Address Space - Server Node



The Server node can then be used to access server-specific information. For example, the node *Server* → *ServerStatus* → *BuildInfo* (see below) contains information specific to the OPC UA server.

Figure 91: Server Specific Address Space



The address space entries under *BuildInfo* can be accessed to learn more information for a specific controller. Placing the variables in a subscription allows easy access to variable values (see screenshot below).



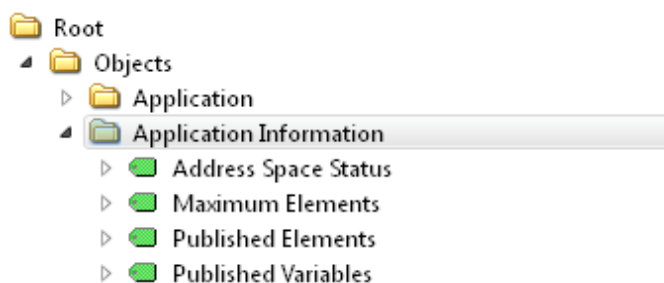
Figure 92: BuildInfo Subscription

#	Server	Node Id	Display Name	Value	Datatype
1	RX3i_CONTROLLER_1	NS0 Numeric 2...	BuildDate	2014-09-17T08:37:14.000Z	DateTime
2	RX3i_CONTROLLER_1	NS0 Numeric 2...	BuildNumber	E476	String
3	RX3i_CONTROLLER_1	NS0 Numeric 2...	ManufacturerN...	GE-IP	String
4	RX3i_CONTROLLER_1	NS0 Numeric 2...	ProductName	PACSystems RX3i	String
5	RX3i_CONTROLLER_1	NS0 Numeric 2...	ProductUri	http://qe-ip.com/PACSystems/RX3i/OPCUAServer	String
6	RX3i_CONTROLLER_1	NS0 Numeric 2...	SoftwareVersion	8.20	String

## OPC UA Server – Application Information

The OPC Server publishes server capabilities within the address space. The information is contained under the Application Information node in the address space (see below).

Figure 93: OPC UA Address Space - Application Information



The variables are defined in the table below.

Variable Name	Description
Address Space Status	Text string indicating variable publish status.
Maximum Elements	Maximum application elements that can be published by the OPC UA server.  Application Variable (Non-Array): 1 variable = 1 element Application Variable (Array): N-Array Dimension = N elements
Published Elements	Number of elements currently published by the OPC UA Server
Published Variables	Number of variables published by the OPC UA Server  Both published variables and arrays count as one Published Variable each, regardless of the array dimension.

Display Name	Value	Data Type
Address Space Status	All Elements Published to Address Space	OpcUaType_String
Maximum Elements	12500	OpcUaType_Int32
Published Elements	58	OpcUaType_Int32
Published Variables	24	OpcUaType_Int32

The example above indicates the following about the PACSystems Controller.

**Address Space Status = All Elements Published to Address Space**

The number of published elements did not exceed the maximum allowed by the controller. Thus, all elements were published. If the maximum had been exceeded, then elements would still be published up to the limit and the text would change to:

**Address Space Status = Maximum Published Elements Exceeded: Address Space Truncated**

**Maximum Elements = 12,500**

Maximum Elements is the maximum number of application elements supported by this controller. In the example above, that limit is 12,500 application elements.

**Published Elements = 58**

**Published Elements** is a count of how many application elements are currently being published. In the example above, the number is 58 application elements.

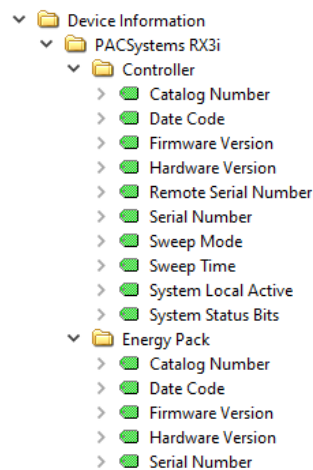
**Published Variables = 24**

**Published Variables** is a count of the controller application variables currently being published. In the example, the number is 24. Note that the **Published Variables** = 24, while **Published Elements** = 58. The difference is due to one of the application variables being a 34-element array.

## OPC UA Server – Device Information

The OPC UA server publishes controller specific information under the Device Information node.

**Figure 94: OPC UA Address Space – Device Information**



The tree structure allows the user to drill down into both the Controller and Energy Pack nodes to get information on these devices. The variables under these nodes are defined as follows

Variable Name	Description
Catalog Number#	(string) Device Catalog Number
Date Code#	(string) Device Date Code
Firmware Version#	(string) Firmware Version installed on the device
Hardware Version#	(string) Hardware Version of the device
Serial Number#	(string) Device Serial Number
Remote Serial Number	(string) Device Serial Number of Redundant Partner
Sweep Mode	(string) Contains a string representation of the CPU's current state. Possible values include: <ul style="list-style-type: none"> <li>• "Run Enabled"</li> <li>• "Run Disabled"</li> <li>• "Stop Enabled"</li> <li>• "Stop Disabled"</li> <li>• "Stop Faulted"</li> <li>• "Stop Halted"</li> </ul>
Sweep Time	(double) Contains the most recent PLC Sweep time in seconds.
System Local Active	(boolean) Returns true if this unit is the Local Active unit and false otherwise. The value of the Boolean will track with that of System Bit %S36 in user logic.
System Status Bits	(uint32) Lower 16-bits contain the CPU's Redundant Status Bits (%S33 through %S48) as a bitmask, the upper 16-bits are reserved and currently will always report 0. <p>Bit 1 = value of %S33 (PRI_UNT)  Bit 2 = value of %S34 (SEC_UNT)  Bit 3 = value of %S35 (LOC_RDY)  Bit 4 = value of %S36 (LOC_ACT)  Bit 5 = value of %S37 (REM_RDY)  Bit 6 = value of %S38 (REM_ACT)  Bit 7 = value of %S39 (LOGICEQ)  Bit 8 = value of %S40 (RESERVED)  Bit 9 = value of %S41 (RDN_COMM_AVAIL)  Bit 10 = value of %S42 (RDN_P1_LINK_UP)  Bit 11 = value of %S43 (RDN_P2_LINK_UP)  Bit 12 = value of %S44 (RESERVED)  Bit 13 = value of %S45 (RESERVED)  Bit 14 = value of %S46 (RESERVED)  Bit 15 = value of %S47 (RESERVED)  Bit 16 = value of %S48 (RESERVED)  Bits 17-32 = RESERVED (set to 0)</p>

**Note:** If the controller does not have an Energy Pack installed, the values for these variables are NA. Only variables marked with a # are available on the Energy Pack node, the rest are Controller only.

## OPC UA Automatic Restart Function

The OPC UA server generates the address space when it starts up. Thus, for a running OPC UA server, adding, deleting, or modifying a publish attribute of a variable requires that the server be restarted.

The OPC UA server automatically restarts when the user changes a variable's published state and return the application to a running state with logic equal. The server automatically restarts to assure that the latest published variables appear in the OPC UA address space. The server will also restart automatically for either a stop-mode store or a run-mode store when the OPC UA server is currently running and the published variable table is changed.

In most cases, the time the server is offline due to the restart operation is relatively short. For Run-Mode Store with either very large programs or significant changes, however, the time period can be extended while the server restart waits for the controller to perform operations necessary to validate the program. Once these operations are complete, the server will return to operational status. If the current runtime status of the server is needed, the SERVER\_STATUS service request can be used.

## OPC UA Non-Transparent Server Redundancy

### Description of OPC UA Non-Transparent Server Redundancy

Starting with PACSystems firmware version 10.10 or later the PACSystems OPC UA server supports OPC UA Non-Transparent Server Redundancy as described in part 4 of the "OPC Unified Architecture Specification". This is achieved by adding the Service Level variable to the OPC UA address space. OPC UA clients can use the OPC UA Non-Transparent Redundancy standard to failover in a redundancy configuration. One suggested OPC UA client supporting OPC UA Non-Transparent Redundancy is GE Digital's CIMPLICITY 10.0 plus using SIM 17 or later.

The specification is available as "OPC Unified Architecture Specification", Release 1.04, November 22, 2017. Published by the OPC Foundation. At the time of this release, online versions of this specification are available at:

<https://reference.opcfoundation.org/v104/>

Only PACSystems CPU that is configured for PACSystems Hot-Standby Redundancy will support OPC UA Non-Transparent Server Redundancy. If a CPU is configured as a simplex unit (either because it does not support redundancy or because it is configured as a simplex CPU), it will offer the Service Level variable in the OPC UA address space, *but it will have a fixed value of 255*. If the CPU is configured as a redundant unit, it will offer the Service Level variable in the OPC UA address space and adjust it as the following table shows:

### Description of Service Levels

The OPC UA client subscribes to the ServiceLevel property provided by each redundant Server. This value indicates which instance is the healthiest (any number >= 200 is considered healthy). If the ServiceLevel drops below 200 the current connection is now considered Degraded and a failover is initiated if the standby server has a greater ServiceLevel.

NOTE: If synchronization is important, take care to place all OPC UA variables on the Redundancy Transfer List, but not strictly required.

Unit Status	Service Level
PLC Run AND Outputs Enabled AND Active Unit (HEALTHY)	220
PLC Run AND Outputs Enabled AND Backup Unit (DEGRADED)	190
PLC Stop OR Outputs Disabled (DEGRADED)	180
PLC Configuration Cleared, OR no PVT table available (NO DATA)	1
Redundancy Disabled in HWC Stored (such as CPE330 simplex HWC) (HEALTHY)	255
Feature Disabled in Model (such as CPE302 or CPE115) (HEALTHY)	255
OPC UA Server Stopped (N/A)	N/A (No Communication Possible)

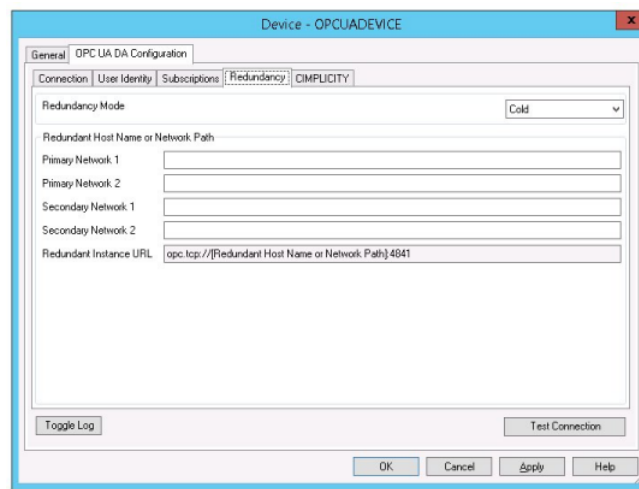
### How to Configure PACSystems Controllers for OPC UA Non-Transparent Redundancy Applications

Emerson supports Non-Transparent Redundancy on select controllers. Their configurations are supported by GE Digital’s CIMPLICITY 10.0 SIM 17 or later. The following instructions are for CIMPLICITY, but the core concepts will apply to other OPC Clients.

The client should be configured for the usual Endpoint URL for the primary PLC. (For example, `opc.tcp://172.31.0.100:4840`).

The OPC UA client will also need to be configured to specify the IP addresses of the primary and secondary PLCs as their Primary Networks 1 and 2. Users can specify these IP addresses from the **Redundancy** tab of CIMPLICITY (Figure 95).

**Figure 95: OPC UA Device Redundancy Configuration in Cimplicity 10 SIM 17**



Redundancy Parameters	Description
Mode	The CIMPLICITY OPC UA Client driver supports <i>Cold</i> , <i>Warm</i> , and <i>Hot</i> redundant OPC UA Servers with or without cabling redundancy. <i>HotandMirrored</i> is not supported.
Primary Network 1	Hostname or network path for primary network 1 for Redundancy feature.
Primary Network 2	Hostname or network path for primary network 2 for Redundancy feature.
Secondary Network 1	Not supported by Emerson PLCs and should remain empty.
Secondary Network 2	Not supported by Emerson PLCs and should remain empty.
Redundant Instance URL	The URL used to connect to a specific redundancy server instance. When running with redundancy the client makes multiple connections to multiple servers which may be used to recover communication when there is a network or device failure. The Redundant Instance URL describes how those URLs are constructed from the information entered in this dialog box. For example: <code>opc.TCP://[Redundant Host Name or Network Path]:4840</code>

For more information on configuring Client Redundancy in CIMPLICITY please visit: <https://www.ge.com/digital/documentation/cimPLICITY/version10/index.html> and search for “OPC UA DA Configuration: Client Redundancy.”

### Failover Modes and Actions

#### Supported Failover Modes†

Failover Modes	Descriptions
<b>Cold</b>	No standby server subscription for all the monitored items is created.
<b>Warm</b>	Two parallel sessions are created with redundant servers. All monitored items are created; however, the standby server subscription has all monitored items set to disabled. When a failover occurs, the sampling is enabled for all monitored items on the standby server and disabled (if possible) on the current master. Hot redundancy minimizes the processing during failover; however, as a result, the underlying data source is sampled by both redundant servers. This is not a concern if the underlying data source is redundant as well. However, if it is shared by the redundant servers, it would be better to use warm redundancy.
<b>Hot</b>	Two parallel sessions are created with redundant servers. All monitored items are created and are set to sample; however, the standby server subscription has to publish disabled. When a failover occurs, publishing is enabled on the standby and disabled (if possible) on the current master.

† **HotandMirrored** Failover Modes are not supported on PACSystems controllers.

### Redundancy Failover Actions

Failover mode and Client options	Cold	Warm	Hot (a)	Hot (b)
<b>On initial connection in addition to actions on Active Server:</b>				
Connect to more than one OPC UA Server.		X	X	X
Create <i>Subscriptions</i> and add monitored items.		X	X	X
Activate sampling on the <i>Subscriptions</i> .			X	X
Activate publishing.				X
<b>At Failover:</b>				
OpenSecureChannel to backup OPC UA Server	X			
CreateSession on backup OPC UA Server	X			
ActivateSession on backup OPC UA Server	X			
Create <i>Subscriptions</i> and add monitored items.	X			
Activate sampling on the <i>Subscriptions</i> .	X	X		
Activate publishing.	X	X	X	

### CAUTION

Should the active PLC fail or be put into STOP mode, the redundant PLCs may perform a role switch, transferring the active unit status to the previous backup unit. This action transfers the backup unit's status to the previous active unit.

In this scenario, CIMPLICITY will begin to update its points with values from the new active unit at the time of the role switch. However, if a point has not changed on the active unit since before the role switch, the CIMPLICITY project will not update its displayed point value. Some values may remain effectively frozen until the unit changes the variable's value.

**The value and its timestamp will continue to show the values obtained from the previously active unit until the variable changes on the new active unit.**

## Failover Performance

Performance of failovers in OPC UA Non-Transparent Server Redundancy will vary depending on the system configuration and type of failure mode. Users should expect greater OPC UA server failover delays with *Cold* and *Warm* failover modes because the OPC UA client must do extra work after the failure to reconnect and/or enable sampling. The exact timing for this to occur will be specific to the load on the PLCs, the hardware specifications of the controller, and the network conditions.

Since the OPC UA client subscribes to the Service Level variable in the OPC UA address space and to obtain the current value when a change takes place requires a publishing period of that changed data, the performance of how long this operation will take can be identified by measuring the performance of the specific configuration for variable updates as described in Section 11.7, *OPC UA Service Performance and Estimation*. The worst-case system subscription update would be the upper bound of how much time it takes for a Service Level change to be set on the server, published over the network, and obtained by the client.

When a network failure or PLC failure that resulted in complete loss of network communication (such as a power loss or watchdog timeout expiration) is detected, the OPC UA Client immediately attempts to connect to the standby server. It can take up to a full connection timeout period for the client to detect the server was lost due to network communication interruption. The specific name for this connection timeout may vary by the type of OPC UA client being used. Please consult the documentation for the specific OPC UA client to determine the value of its network communication timeout setting.



## 11.7 OPC UA Service Performance and Estimation

The throughput of the OPC UA server on PACSystems is determined by several factors, some of which are related to the OPC UA protocol itself, and some that are not. Here are some of the configuration settings that should be considered when trying to determine what to expect for OPC UA performance.

### OPC-UA-Specific Items

- The number of OPC UA address space elements published in PAC Machine Edition configuration
- The number of OPC UA address space elements that are scalar variables or part of an array
  - E.g. Updating an array of 1000 elements is a much faster and efficient operation than updating 1000 individual scalar variables
- The number of OPC UA clients attached to the OPC UA server
- The number of monitored items configured on the OPC UA clients
- The sampling rate configured by the OPC UA clients on the monitored items
- The publishing interval set by the OPC UA clients on the monitored items
- The OPC UA Server Log Level setting
  - E.g. Logging turned off leads to the fastest performance and the highest-level logging will degrade OPC UA server performance significantly. (Logging should only be turned on for debugging OPC UA problems.)

### PACSystems-Specific Items

- The PACSystems controller model used
  - E.g. Using a CPE305 will be slower than a CPE330
- The Ethernet Port data rate
  - E.g. Using Ethernet in 100 MB vs 1 GB mode
- The concurrent use of other protocols on the CPU's embedded Ethernet interface
  - E.g. Profinet, EGD, Modbus TCP, or SRTP
- The PACSystems controller's Sweep Mode settings
- The PACSystems controller's Sweep Time
- Use of Interrupt logic blocks in the PACSystems controller
- How often the published variables are changing in the PACSystems controller
- The volume of Ethernet traffic the PACSystems controller is processing

The OPC UA protocol on PACSystems controllers consumes time from the Controller Communications Window part of the PACSystems controller sweep. This window may be set to various modes and has an impact on how much time the OPC UA server can execute each sweep

in the controller. The OPC UA protocol has a workload based on the quantity of variables that are part of the subscriptions created by connected clients, and the quantity of OPC UA variable read-and-write requests that are in progress. If the workload is greater than the sweep time, then the workload is broken up over one or more sweeps of the PACSystems controller.

If there is not enough time left in the sweep to perform the OPC UA workload, then the PLC will prioritize the timing commitments for User Logic execution and I/O scan. As a result, the OPC UA server performance will be significantly slowed to accommodate the other timing commitments.

## OPC UA Server Sampling and Publishing of Data

The OPC UA subscription data monitoring is impacted by four parameters: the *Sampling Interval Timer*, the *Publishing Interval Timer*, the *Queue Size*, and the *Discard Policy (Oldest/Newest)*.

Item	Description
Sampling Interval	The sampling interval that governs the frequency in which the PLC can queue data and monitor for data change
Publishing Interval	The time interval in which the OPC UA server sends data (publishes) changes to the OPC UA client
Data Queue Size	The number of samples the server can queue within the publishing interval
Discard Policy (Oldest/Newest)	The parameter that determines whether the oldest data or newest data should be discarded first

## Considerations

The Sampling Interval determines the speed at which the controller can monitor data changes. The Sampling Interval is not to exceed a rate at which a change in variable value can be detected. However, the Sampling Interval does not guarantee the rate at which a variable is sampled. If the controller has too large of a workload, the actual timing of the sampling interval will be longer than the time requested, but the PLC will still check for all data changes. These checks will occur before another sample is started; no item will be skipped when the sampling interval time expires.

The Publishing Interval Timer determines the speed at which the OPC UA server sends the detected value changes to the OPC UA client. No matter how fast the changes to data in a subscription are detected, the data updates will not be sent to the client until the next publishing interval. If sampled data changes are enqueued faster than they are dequeued by publishing, then it is possible the change queue could fill up and new samples will require a value to be discarded. The Queue Size setting determines how many samples will be queued on the server-side. The Monitored Item setting determines if the oldest or the newest value will be discarded if the monitored item queue has become full.)

Note: The Sampling Interval, Publishing Interval, Queue Size, and Discard Policy (Oldest/Newest) are all available in the OPC UA protocol; however, it is possible that not all OPC UA clients support

adjusting each of these settings. For example, the UA Expert OPC UA client does support each of these settings.

## OPC UA Server Performance Estimation

### Determine the Baseline Variable Updates

The UA Expert tool can be used to take some measurements that should help to estimate how much processing time it takes to handle a given subscription. To measure this execution time for each anticipated subscription, create an estimate of how much work the PLC needs to handle the subscriptions created by the clients. Once the baseline workload is understood, then an estimate over how many sweeps it will take to complete this work can be calculated based on the Controller Communication window time. From here, estimate the expected frequency of data updates from the OPC UA server to the attached clients.

### Determine the Estimated Time Available to Perform the OPC UA Protocol Workload

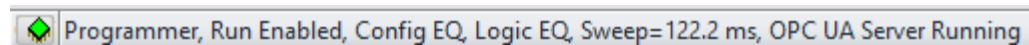
**Note:** For help in understanding the CPU Sweep modes, please see GFK-2222, PACSystems CPU Reference Manual.

### Estimate the PLC Sweep Time

Use the production PME project to determine an estimate for the PLC Sweep time in milliseconds for the expected project. This can be measured by running the expected PME project on the PACSystems controller and observing the “Sweep Time” on the bottom of the PME screen. This gives a good idea of what the PLC’s sweep time is in general for this estimate.

---

**Figure 96: Sweep Time Location**



### Determine the Controller Communication Time Limit

During each sweep, OPC UA tasks may only use a portion of the Controller Communication Window execution time. The PLC will prioritize its timing commitments with other protocols, such as the embedded Ethernet Port (SRTP and Modbus TCP) and Serial communications.

By default, the PACsystems CPU is set up for Normal Sweep Mode; however, this guide will help users determine the controller’s communication time limit in three different sweep modes: *Normal Sweep Mode*, *Constant Sweep Mode*, *Constant Window Mode*. These modes will operate concerning the Controller Communication window as follows:

### NORMAL SWEEP MODE

When in Normal Sweep mode, the PACSystems CPU's Controller Communication Window is either limited to a set time in milliseconds (Limited Setting) or set to run until completion (Complete Setting). In either case, there is an adjustable upper-limit of execution time for OPC UA tasks of 100 ms, so if more than that execution time is expected for OPC UA protocol each sweep, then either use a different sweep mode or adjust the OPC UA task limit timer as described in the following subsections of :

- *OPC UA Server – Service Request – SET\_TIME\_MS\_LIMIT\_PER\_SWEEP*
- *OPC UA Server – Service Request – GET\_TIME\_MS\_LIMIT\_PER\_SWEEP*

This OPC UA task limit is only used when the controller is in Normal Sweep mode. Additionally, when in the Normal Sweep mode, if the Controller Communication Window tasks are complete before the limit timer expires then the PACSystems CPU will immediately advance to the next sweep phase, which is the Backplane Communication Window.

### CONSTANT SWEEP MODE

When in Constant Sweep Mode, the Controller Communication, Backplane Communication, and Background Task windows will run in rotation from the time output scan finishes until the next sweep begins no matter how much OPC UA work is left to complete. In this sweep mode, the windows will continue to run and the PACSystems Controller will not advance to the next sweep until the timer expires.

### CONSTANT WINDOW MODE

When in Constant Window Mode, the Controller Communication, Backplane Communication, and Background Task windows will run in rotation from the time output scan finishes until the Constant Window timer expires no matter how much OPC UA work is left to complete. In this sweep mode, the windows will continue to run and the PACSystems Controller will not advance to the next sweep until the timer expires. There are similarities between Constant Sweep Mode and Constant Window Mode but the main difference between these two modes is when the timer is started. *In Constant Sweep mode, the timer is started at the beginning of the PACSystems CPU sweep, but in Constant Window mode, the timer is started at the end of the Output Scan phase of the sweep.*

Use this information to determine how long in milliseconds will be allowed for OPC UA tasks to execute in the Controller Communication window.

## Calculate the Percentage of Time Available for OPC UA Workload

The amount of time available for OPC UA protocol execution can be calculated with the following formula:

$$\frac{\text{Controller Communication Window Time Limit (ms)}}{\text{Sweep Time (ms)}}$$

### EXAMPLES

The following examples can be used to estimate how much time is dedicated to the OPC UA protocol over the total time available

- A. If the Controller Communication Window is limited to 10 ms (default configuration) and the PLC's sweep time is 90 ms, then about 10 ms/100 ms will be available for OPC UA communication. This could be represented as about 10% of the time is permitted to be used by OPC UA.
- B. If the Controller Communication Window is set to 30 ms and the PLC's sweep time is about 120 ms, then about 30 ms/150 ms will be available for OPC UA communication, which is about 1/5 of 20% of the time is permitted to be used by OPC UA.
- C. If Controller Communication is set to 10 ms and the PLC's sweep time is about 840 ms, then about 10 ms / 850 ms will be available for OPC UA communication, which is about 10 ms /850 ms or about 1/85 or about 1% of the time is permitted to be used by OPC UA. In this case, the same OPC UA configuration would perform 10 times slower than the similar configuration in example A and 5 times slower than the similar configuration in example B.

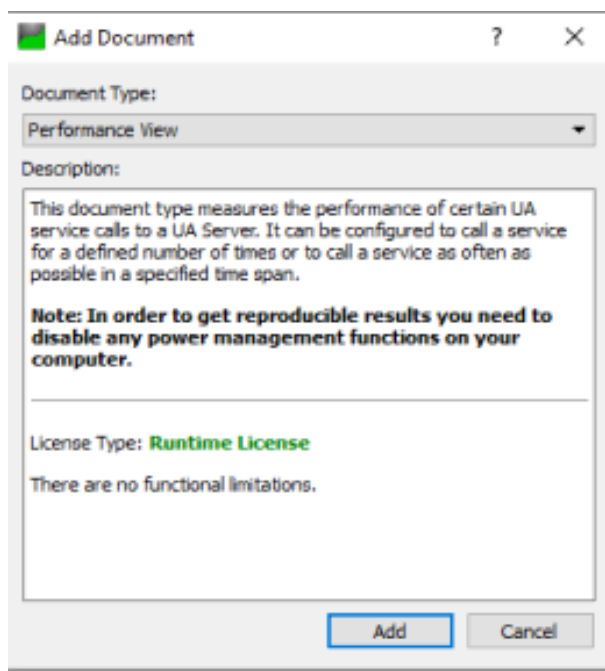
### Measure Each Subscription Time Individually

1. Create a baseline PME configuration with a representative example of variables.

Store the project to the controller and place the controller into Stop/Outputs Disabled mode with the OPC UA server running.

2. Open UA Expert and connect to the controller with UA Expert and add the Performance View document as seen in Figure 97: Add Performance View Document.

**Figure 97: Add Performance View Document**



3. Drag and drop the variables that are expected to be subscribed to into the **Performance View Nodes** section. Note the number of nodes added in Figure 98.

**Figure 98: Nodes in List**

Nodes in List: 10002

4. Select only the **Read Registered** configuration (, use the full number of variables in the **Node Counts** section, and set the **Cycles**: setting to 100 for iterations. Press **Start Test** and wait for the **Read Registered** results to appear in the **Log** at the bottom of the screen.

**Figure 99: Performance View Configuration**

Performance View Configuration window showing settings for a test. The 'Read Registered' option is selected. The 'Node Counts' is set to 10000, 'Cycles' is set to 100, and 'Duration' is set to 00:00:10. The 'Start Test' and 'Stop Test' buttons are visible at the bottom right.

5. Interpret the results. This test informs the user how long it takes to read monitored items assuming they all change every time. This should result in a Min, Average, and Max time in milliseconds for how long it takes to update. Use the Max time and this is a rough estimate of the time it takes to read a full subscription assuming all variables have changed.
6. Remove all variables from the **Nodes** section to prepare for the next measurement.
7. **Important:** Repeat steps 1 through 5 for each expected client subscription. The max value should be summed for each pass through these steps. The maximum number of subscriptions for PACSystems is 10 so no more than 10 passes through steps 1 through 5 should be necessary to estimate the total base workload for the PACSystems OPC UA server. The sum of the max times for all subscriptions planned should be used as an estimate for the OPC UA protocol workload.

## Determine the Worst-Case Estimate of Real-Time used per Subscription

Once the time percentage permitted has been calculated (to be used by the protocol), determine the (estimated) real-time it will take for a given OPC UA subscription. The following formula can be used to estimate the worst-case real-time used per subscription:

$$\left( \frac{100}{\text{Percentage of Time Available for OPC UA}} \right) * \text{Time for OPC UA Workload} = \text{Realtime to Complete}$$

### EXAMPLES

- A. If a system has a 10% sweep dedicated to OPC UA and a 100 ms OPC UA workload, then the real-time subscription would take 1 second to execute.

$$\left( \frac{100\%}{10\%} \right) * 100 \text{ ms} = 10 * 100 \text{ ms} = 1000 \text{ ms or 1 second}$$

- B. If a system has a 20% sweep dedicated to OPC UA and a 400 ms OPC UA workload, then the real-time subscription would take 2 seconds to execute.

$$\left( \frac{100\%}{20\%} \right) * 400 \text{ ms} = 5 * 400 \text{ ms} = 2000 \text{ ms or 2 second}$$

### Sum the results

Now that the real-time needed to process variable sampling and publishing for each subscription has been calculated, determine the estimated real-time performance for the OPC UA clients. Add the real-time estimate for each subscription together and the OPC UA client should see an update of all variables within that sum of time.

Using this calculation method will provide the worst-case results within the estimated real-time performance calculated with this method.

**Note:** If the system is configured with some subscriptions configured for a faster sampling/publishing interval, and some subscriptions with a slower sampling/publishing interval, then this total time is really for the overall system. Those individual subscriptions that are set to a faster rate will update faster than the overall update rate of the system.

## OPC UA Server Performance Measurement

### Measure Subscription Performance with UA Expert

It is possible to get a real-world measurement of actual server performance by using the **Subscribe** Configuration in the UaExpert Performance View document.

**Figure 100: Performance View**

---

Performance View

**Configuration**

<input type="checkbox"/> Read	<input type="checkbox"/> Create / Delete Monitored Items
<input type="checkbox"/> Write	<input checked="" type="checkbox"/> Subscribe
<input type="checkbox"/> Read Registered	Sampling Interval: <input type="text" value="500"/>
<input type="checkbox"/> Write Registered	Publish Interval: <input type="text" value="1000"/>
<input type="checkbox"/> All	Queue Size: <input type="text" value="1"/>

---

Before beginning, the user will need to know the actual variable configuration and the actual project running in an environment that is close to the production environment that will be used for the PACSystems Controller. The user will also need to drag-and-drop the variables into the Performance View Nodes to measure performance. This will allow the user to measure the performance for a single subscription at a time with this view.

**Note:** The additional subscriptions will degrade the performance so the measured number can only be used as an estimate for that subscription in isolation.

**Note:** The UA Expert client does not assume that all variables change every sampling interval and instead only considers the actual changes it received during the test procedure. If the PME project is not running as it would in a production environment, add logic to artificially change the variables to closely simulate the actual real-world performance of the system.



## Section 12      Diagnostics

This chapter describes diagnostic techniques for a PACSystems Ethernet interface.

- What to do if You Cannot Solve the Problem
- Diagnostic Tools Available for Troubleshooting
- States of the Ethernet interface
- ETHERNET OK/OK Blink Codes for Hardware Failures
- Controller Fault Table
- Monitoring the Ethernet interface Status Bits
- Monitoring the FT Output of the COMMREQ Function Block
- Monitoring the COMMREQ Status Word (CSW)
- Using the EGD Management Tool
- Troubleshooting Common Ethernet Difficulties

## 12.1 What to do if You Cannot Solve the Problem

If you cannot solve the problem, contact Technical Support. Please have the following information ready:

- The Name and Catalog Number are marked on the product.
- PLC CPU version number from CME Status screen
- The Ethernet's IC number (Ex: IC695ETM001) Description of symptoms of the problem. Depending on the problem, you may also be asked for the following information:
- The ladder logic application program and the PLC sweep time at the time the problem occurred.
- A listing of the configuration parameters for the Ethernet interface that failed.
- A description of the network configuration. This should include the number of PLCs and host computers accessing the network, the type of network cable used (e.g. twisted pair, fiber optic, etc.), length of network cable, and the number and manufacturer of transceivers, hubs, and network switches used.
- Description of all Ethernet communication activity for the PLC.
- Versions of all software communicating with the PACSystems controller via Ethernet. This includes PAC Logic Developer, CIMPLICITY PE, IFIX, etc.
- Be prepared to provide the Controller Fault Table showing Fault Extra Data
- Be prepared to provide Station Manager Log showing Ethernet Events

## 12.2 Diagnostic Tools Available for Troubleshooting

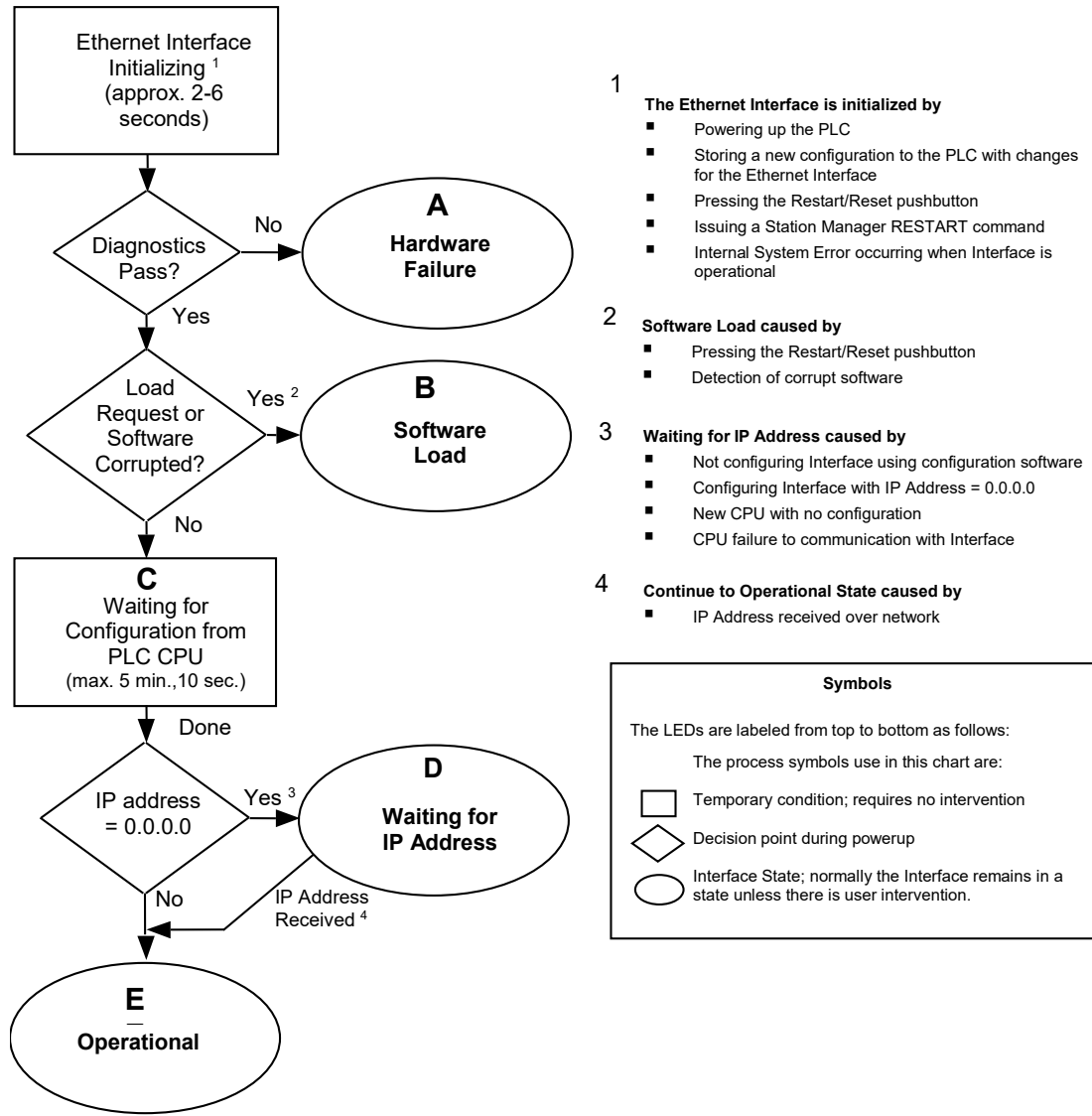
There are several tools to assist you in diagnosing problems with the Ethernet interface and the network.

- Use the *Ethernet Interface LEDs* to troubleshoot a problem on power-up of the Ethernet interface and for an immediate visual summary of the operational state of the interface.
- Use the Controller Fault Table to troubleshoot a problem once the interface is running. It provides a record of exceptions logged by the PLC, the Ethernet interface, and other I/O and communications modules. The Controller Fault Table is accessed through the programming software or, if supported, in a web browser.
- For Controller Fault Table entries generated by the Ethernet interface, the Detailed Fault Data for that entry contains the same data as the corresponding event in the Ethernet interface's exception log. Refer to the *PACSystems TCP/IP Ethernet Communications Station Manager User Manual*, GFK-2225, for information on how to interpret Ethernet exception log events.
- Use the Ethernet Status Data to troubleshoot the Ethernet interface status
- For Ethernet Global Data operation, the EGD Management Tool can be used to check the online operation of the EGD network, and Exchange Status words can be used to troubleshoot exchange operations.
- Use the Station Manager to troubleshoot a problem with the Ethernet interface, the network, PLC backplane communication, or with your application. The LOG, TALLY, EXS, CHANNEL, STAT, and XCHANGE Station Manager commands are especially useful.
- The LOG command provides a complete record of exceptions occurring with the network and interface.
- The TALLY command provides statistics about the operation and performance of the network and interface.
- The EXS command provides information about COMMREQs.
- The CHANNEL command displays detailed information about a specified SRTP or Modbus/TCP communication channel.
- The STAT command provides the current status of specific components of the Ethernet interface. Of particular use, the STAT V and STAT H commands provide SRTP server and SRTP channel status, respectively. The STAT O and STAT M commands provide Modbus/TCP server and channel status, respectively. The STAT G command provides the current status on the operation of EGD communications on the interface.
- The XCHANGE command displays detailed information about a specified Ethernet Global Data exchange.

Refer to the *PACSystems TCP/IP Ethernet Communications Station Manager User Manual*, GFK-2225, for information on how to access and use the Station Manager software.

# 12.3 Initialization Example of the RX3i Ethernet ETM001-Jx Module Interface

Figure 101: States of the Ethernet interface

















































- 1 **The Ethernet Interface is initialized by**
  - Powering up the PLC
  - Storing a new configuration to the PLC with changes for the Ethernet Interface
  - Pressing the Restart/Reset pushbutton
  - Issuing a Station Manager RESTART command
  - Internal System Error occurring when Interface is operational
- 2 **Software Load caused by**
  - Pressing the Restart/Reset pushbutton
  - Detection of corrupt software
- 3 **Waiting for IP Address caused by**
  - Not configuring Interface using configuration software
  - Configuring Interface with IP Address = 0.0.0.0
  - New CPU with no configuration
  - CPU failure to communication with Interface
- 4 **Continue to Operational State caused by**
  - IP Address received over network

**Symbols**

The LEDs are labeled from top to bottom as follows:  
 The process symbols use in this chart are:

- Temporary condition; requires no intervention
- ◇ Decision point during powerup
- Interface State; normally the Interface remains in a state unless there is user intervention.

- Operational**
- Full support for client and server capability
  - Uses user-defined Advanced Parameters

<b>LED Behavior</b>  On  Off  Blinking	<b>Where Stopped</b>	<b>Possible Cause</b>	<b>Corrective Actions</b>
<p align="center"><b>Hardware Failure</b></p> <p>ETM001                      ETM001-Kxxx</p> <p> Ethernet OK             OK</p> <p> LAN OK                     STATUS</p> <p> LOG EMPTY               LAN OK</p> <p>    LOG EMPTY</p>	<p>A</p> <p>Hardware Failure</p>	<p>Fatal Hardware Error</p>	<ul style="list-style-type: none"> <li>• Make sure the PLC has power.</li> <li>• Examine Controller Fault Table for clues.</li> <li>• Recheck PLC Programmer configuration.</li> <li>• Power off baseplate, inspect the interface for loose components, reseat the module, and Restart.</li> <li>• If the problem persists, replace the PLC hardware.</li> </ul>
<p align="center"><b>Software Loader Failure</b></p> <p>ETM001                      ETM001-Kxxx</p> <p> Ethernet OK             OK</p> <p> LAN OK                     STATUS</p> <p> LOG EMPTY               LAN OK</p> <p>    LOG EMPTY</p>	<p>B</p> <p>Software Loader</p>	<p>Software corrupt</p>	<ul style="list-style-type: none"> <li>• Connect a PC Software Loader and load new software.</li> </ul>
<p align="center"><b>Waiting for Config from PLC</b></p> <p>ETM001                      ETM001-Kxxx</p> <p> Ethernet OK             OK</p> <p> LAN OK                     STATUS</p> <p> LOG EMPTY               LAN OK</p> <p>    LOG EMPTY</p> <p><b>Blink behavior:</b> Regular Periodic Blinking</p>	<p>C</p> <p>Waiting for Configuration from PLC</p>	<p>Did not configure slot using the PLC Programmer. CPU not communicating with Ethernet interface. (Condition can last a maximum of 5 minutes.)</p>	<ul style="list-style-type: none"> <li>• Use the PLC Programmer configuration software to configure the interface then store the configuration to the PLC CPU.</li> <li>• Power cycle the PLC.</li> <li>• Clear faults and Restart interface.</li> </ul>
<p align="center"><b>Unrecoverable Hardware Failure</b></p> <p>ETM001                      ETM001-Kxxx</p> <p> Ethernet OK             OK</p> <p> LAN OK                     STATUS</p> <p> LOG EMPTY               LAN OK</p> <p>    LOG EMPTY</p> <p><b>Blink Behavior:</b> Pattern 9-digit Blink Code</p>		<p>Unrecoverable hardware or runtime failure</p>	<ul style="list-style-type: none"> <li>• See the list of blink codes on the next page.</li> </ul>
<p align="center"><b>Waiting for IP Address</b></p> <p>ETM001</p> <p> Ethernet OK</p> <p>  LAN OK</p> <p> LOG EMPTY</p>	<p>D</p> <p>Waiting for IP Address</p>	<p>Interface's IP address has not been configured or has been configured as 0.0.0.0.</p>	<ul style="list-style-type: none"> <li>• Use the PLC Programmer to configure the interface with a non-zero IP address.</li> <li>• Assign IP address over the network</li> </ul>
<p align="center"><b>Operational</b></p> <p>ETM001                      ETM001-Kxxx</p> <p> Ethernet OK             OK</p> <p>  LAN OK                     STATUS</p> <p>  LOG EMPTY                LAN OK</p> <p>     LOG EMPTY</p>	<p>E</p> <p>Operational</p>	<p>If the LAN LED is OFF, the problem may be network cable not connected If the STAT LED is OFF, an exception condition has occurred.</p>	<ul style="list-style-type: none"> <li>• Connect cable.</li> <li>• Examine Controller Fault Table to find out why the STAT LED is OFF.</li> </ul>

## 12.4 ETHERNET OK/OK LED Blink Codes for Hardware Failures (ETM001-Jx)

The ETHERNET OK/OK LED indicates whether the module can perform the normal operation. This LED is on for normal operation and flashing for all other operations. If hardware or unrecoverable runtime failure occurs, the ETHERNET OK/OK LED blinks a two-digit error code. The ETHERNET OK/OK LED first blinks to indicate the most significant error digit, then after a brief pause blinks again to indicate the least significant error digit. After a long pause the error code display repeats

Blink Code	Description	Blink Code	Description
0x12	Undefined or Unexpected Interrupt.	0x42	Firmware Loader error
0x13	Timer failure during power-up diagnostics.	0x51	Unexpected watchdog timer exception
0x14	DMA failure during power-up diagnostics.	0x52	Unexpected debug exception
0x21	RAM failure during power-up diagnostics.	0x61	Boot: Critical interrupt exception
0x22	Stack error during power-up diagnostics.	0x62	Boot: Machine check exception
0x23	Shared Memory interface error during power-up diagnostics.	0x63	Boot: Datastore exception
0x24	Firmware CRC (cyclic redundancy check) error during power-up or Factory Test <b>Note:</b> CRC error or software error during normal operation causes Ethernet restart	0x64	Boot: Instruction store exception
0x25	Run time exception	0x65	Boot: External interrupt exception
0x26	No mail communication available during software load	0x66	Boot: Alignment exception
0x27	Serial EEPROM access exception	0x67	Boot: Program exception
0x28	Serial EEPROM reset exception	0x68	Boot: System call exception
0x31	Machine check exception	0x69	Boot: PIT interrupt exception
0x32	Datastore exception.	0x71	Boot: FIT interrupt exception
0x33	Instruction store exception	0x72	Boot: WDT interrupt exception
0x34	Alignment exception	0x73	Boot: Data cache TLB miss exception
0x35	Program exception	0x74	Boot: Instruction cache TLB miss exception
0x36	System call exception	0x75	Boot: Debug exception
0x37	Unexpected IRQ exception	0x76	Boot: Flash memory CRC error
0x38	Data cache TLB miss exception	0x77	Boot: Unexpected ACFAIL interrupt
0x39	Instruction cache TLB miss exception	0x78	Boot: Unexpected Restart pushbutton interrupt
0x41	BSP startup error		



Controller Fault	User Action
Can't locate remote node; discarded request	Error reported when message received where IP/MAC address cannot be resolved. The error may indicate that the remote host is not operational on the network. Check that the remote host is operational on the network and its addresses are correct.
COMMREQ - Bad task ID programmed	Message from PLC for unknown Ethernet interface task. Check COMMREQ function block.
COMMREQ - Wait for mode not allowed	Check COMMREQ to make sure sent in no-wait mode.
Configured gateway address bad; can't talk off the local net	Configuration error. Verify that IP address, Subnetwork Mask, and Default Gateway IP address are correct.
Connection to remote node failed; resuming without it	Underlying communications software detects error transferring data; resuming. If persistent error, check the connection to LAN and operation of the remote node.
LAN controller fault; restart LAN I/F	HW fault, perform a power cycle. If the problem persists, contact Technical Support.
LAN controller TX underflow; attempt recovery	Internal system error. If the problem persists, contact Technical Support.
LAN controller underrun/overrun; resuming	Internal system error. If the problem persists, contact Technical Support.
LAN data memory exhausted - check parameters; resuming	The Ethernet interface does not have free memory to process communications. If the problem persists, contact Technical Support.
LAN duplicate MAC Address; resuming	A-frame was received in which the source MAC Address was the same as this station's MAC Address. All stations on a network must have a unique MAC address. Immediately isolate the offending station; it may be necessary to turn it off or disconnect it from the network. This station remains Online unless you intervene to take it Offline.
LAN I/F cannot init - check parameters; running soft Sw utl	Internal system error. If the problem persists, contact Technical Support.
LAN I/F capacity exceeded; discarded request	Verify that connection limits are not being exceeded.
LAN interface hardware failure; switched off network	Replace the Ethernet interface.
LAN network problem exists; performance degraded	The excessive backlog of transmission requests due to excessive traffic on the network. For a sustained period, the MAC was unable to send frames as quickly as requested. If the problem persists, contact Technical Support.



Controller Fault	User Action
LAN severe network problem; attempting recovery	The external condition prevented transmission of the frame in the specified time. Could be a busy network or network problem. Check transceiver to make sure it is securely attached to the network.
LAN system-software fault; aborted connection resuming	Internal system error. If the problem persists, contact Technical Support.
LAN system-software fault; restarted LAN I/F	Internal system error. If the problem persists, contact Technical Support.
LAN system-software fault; resuming	Internal system error. If the problem persists, contact Technical Support.
LAN transceiver fault; OFF network until fixed	Transceiver or transceiver cable failed or became disconnected. Reattach the cable or replace the transceiver cable. Check the SQE test switch if present on the transceiver.
Local request to send was rejected; discarded request	Internal error. Check that the Ethernet interface is online. If the problem persists, contact Technical Support.
Memory backup fault; may lose configuration/log on restart	Internal error accessing a non-volatile device. If the problem persists, contact Technical Support. Replace the Ethernet interface.
Module software corrupted; requesting reload	Catastrophic internal system error. Contact Technical Support.
Module state doesn't permit COMMREQ; discarded	COMMREQ is received when the Ethernet interface cannot process COMMREQ. Make sure the Ethernet interface is configured and online. The error may occur if the logic application is sending COMMREQs faster than the Ethernet interface can process them. Reduce the rate at which COMMREQs are sent.
Unsupported feature in the configuration	PLC firmware does not support Ethernet communications software or an attempt has been made to configure a feature not supported by the Ethernet interface. Check CPU and Ethernet interface revisions, order upgrade kit for CPU and/or Ethernet interface.
Can't locate remote node; discarded request	A specified remote device does not exist on the network. Check that the remote device IP address is correct and that the remote device is functioning properly.
Mailbox Queue full – COMMREQ aborted	The CPU is attempting to send COMMREQs faster than the Ethernet interface can receive them. The PLC logic program should retry the COMMREQ after a short delay. If the condition persists, the logic application should be revised to reduce the rate at which it sends COMMREQs to the Ethernet interface.
Non-critical CPU software event	The CPU is attempting to send mail messages faster than they can be retrieved by the Ethernet interface; the messages are discarded. This can result in subsequent "Backplane communications with controller fault; lost request" faults.

## 12.6 Monitoring the Ethernet Interface Status Bits

The Ethernet Interface status bits occupy a single block of memory, which is specified when the Ethernet Interface is configured. The Ethernet Interface updates the status bits in the CPU once each controller scan. These bits can be used to prevent the initiation of a COMMREQ function when certain errors occur or to signal a problem on an established channel.

The first 16 bits of the block are the LAN Interface Status (LIS) bits. The next 64 bits are Channel Status bits (2 for each channel). If the LAN Interface OK bit (bit 16) is not set, the other status bits are invalid.

Status	Description			
	ETM001-Jx	ETM001-Kxxx	RX3i CPE302/CPE305/CPE310 Embedded	RX3i CPE330/CPE400/CPL410 Embedded
1	Port 1A full-duplex	Port 1 full-duplex	Port full-duplex	Port full-duplex LAN1 stays here for compatibility with other CPEs.
2	Port 1A 100 Mbps	Port 1 100 Mbps	Port operating at the highest supported speed	Port operating at the highest supported speed LAN1 stays here for compatibility with other CPEs.
3	Port 1B full-duplex	Port 2 full-duplex	Reserved	LAN2 Port 1 (Top Port) full-duplex
4	Port 1B 100 Mbps	Port 2 100 Mbps	Reserved	LAN2 Port 1 (Top Port) operating at the highest supported speed
5	Network Time Locked	Network Time Locked	Reserved	Network Time Locked
6	A redundant IP address is active	A redundant IP address is active	Reserved	A redundant IP address is active
7	Reserved	LAN1 Port 1 @ 1000 Mbps	Reserved	LAN2 Port 2 (Bottom Port) full-duplex
8	Reserved	LAN2 Port 2 @ 1000 Mbps	Reserved	LAN2 Port 2 (Bottom Port) operating at highest supported speed.
9	Any Channel Error (error on any channel)	Any Channel Error (error on any channel)	Any Channel Error (error on any channel)	Any Channel Error (error on any channel)
10	Reserved	Reserved	Reserved	LAN2 Port 1 Link Indicates that the port has a link.
11	Reserved	Reserved	Reserved	LAN2 Port 2 Link Indicates that the port has a link.

Status Bits	Description			
	ETM001-jx	ETM001-Kxxx	RX3i CPE302/CPE305/CPE310 Embedded	RX3i CPE330/CPE400/CPL410 Embedded
12	Reserved	Reserved	Reserved	LAN2 OK Indicates that the application can get out in the network on at least 1 port.
13	LAN OK	LAN OK	LAN OK	LAN OK LAN1 has only 1 port.
14	Resource problem	Resource problem	Resource problem	Resource problem
15	Module Overtemp (RX3i rack-based ETM001-jx only)	Module Overtemp (RX3i rack-based ETM001-jx only)	Reserved	Reserved
16	LAN interface OK	LAN interface OK	LAN interface OK	LAN interface OK
17	Channel 1 Status SRTP: Data Transfer Modbus TCP Client: Channel Open	Channel 1 Status SRTP: Data Transfer Modbus TCP Client: Channel Open	Channel 1 Status SRTP: Data Transfer Modbus TCP Client: Channel Open	Channel 1 Status SRTP: Data Transfer Modbus TCP Client: Channel Open
18	Channel 1: Modbus TCP Client - Reserved SRTP Client - Channel Error	Channel 1: Modbus TCP Client - Reserved SRTP Client - Channel Error	Channel 1: Modbus TCP Client - Reserved SRTP Client - Channel Error	Channel 1: Modbus TCP Client - Reserved SRTP Client - Channel Error
19-48	Channels 2-16 (same pattern as bits 17&18)	Channels 2-16 (same pattern as bits 17&18)	Channels 2-16 (same pattern as bits 17&18)	Channels 2-16 (same pattern as bits 17&18)
49-80	Channels 17-32 (same pattern as bits 17&18)	Channels 17-32 (same pattern as bits 17&18)	Reserved	Channels 17-32 (same pattern as bits 17&18)

Status Bits	Description
RSTi-EP CPE100/CPE115 Embedded	
1	Port full-duplex LAN1 Port 0 stays here for compatibility with other CPEs.
2	Reserved
3	LAN2 Port 2 full-duplex
4	Reserved
5	Reserved
6	Reserved
7	LAN2 Port 3 full-duplex
8	Reserved
9	Any Channel Error (error on any channel)
10	LAN2 Port 2 Link Indicates that the port has a link.
11	LAN2 Port 3 Link Indicates that the port has a link.
12	LAN2 OK Indicates that the application can get out in the network on at least 1 port.
13	LAN OK LAN1 has only 1 port.
14	Resource problem
15	Reserved
16	LAN interface OK
17	Channel 1 Status SRTP: Data Transfer Modbus TCP Client: Channel Open
18	Channel 1: Modbus TCP Client - Reserved SRTP Client - Channel Error
19-32	Channels 2-8 (same pattern as bits 17&18)
33-80	Reserved

## 12.6.1 LAN Interface Status (LIS) Bits

The LAN interface Status bits monitor the health of the Ethernet interface.

### **Bit 1, Port 1A Full-duplex (RX3i Ethernet Modules) Port Full-duplex (RX3i Embedded Ethernet Interfaces)**

This bit is set to 1 when the port is set to full-duplex. Full-duplex or half-duplex operation is automatically negotiated between the Ethernet interface and its immediately-connected network device, usually a network hub or switch. If this bit is 0, the port is in half-duplex Ethernet mode. This bit is only valid if bit 13 (LAN OK) is 1.

### **Bit 2, Port 1A 100 Mbps (RX3i Ethernet Modules) Port Operating at Highest Supported Speed (RX3i Embedded Ethernet Interfaces)**

This bit is set to 1 when the port is operating at its highest supported speed.

### **Bit 3, Port 1B Full-duplex (RX3i Ethernet Modules)**

This bit is set to 1 when Port 1B is set to full-duplex. Full-duplex or half-duplex operation is automatically negotiated between the Ethernet interface and its immediately-connected network device, usually a network hub or switch. If this bit is 0, the port is operating in half-duplex Ethernet mode. This bit is only valid if bit 13 (LAN OK) is 1.

### **Bit 3, LAN2 Port 1: Port Operating at Full Duplex (RX3i CPE330/CPE400/CPL410 Embedded)**

This bit is set to 1 when the port is operating in full-duplex mode.

### **Bit 4, Port 1B 100 Mbps (RX3i Ethernet Module)**

This bit is set to 1 when Port 1B is operating at 100 Mbps.

### **Bit 4, LAN2 Port 1: Port Operating at Highest Supported Speed (RX3i CPE330/CPE400/CPL410 Embedded)**

This bit is set to 1 when the port is operating at its highest supported speed.

### **Bit 5, Network Time Locked (RX3i Ethernet Module and RX3i CPE330/CPE400/CPL410 Embedded)**

The Ethernet clock is locked to a network SNTP timer server. When this bit is 0, the Ethernet module has lost its lock to a network time server or was never locked to a timeserver. This bit is updated whether or not the SNTP Time Transfer feature is configured and whether or not the logic application has enabled CPU Time Update interrupts. For more information, see Section 7.4, *Time-Stamping of Ethernet Global Data Exchanges*.

### **Bit 6, Redundant IP Address Active (RX3i Ethernet Module and RX3i CPE330/CPE400/CPL410 Embedded)**

This bit is set to 1 when the configured Redundant IP address is active. Otherwise, this status bit is set to 0.

### **Bit 7, LAN2 Port 2: Port Operating at Full Duplex (RX3i CPE330/CPE400/CPL410 Embedded)**

#### **LAN1 Port 1 (ETM001-jx)**

This bit is set to 1 when the port is operating in full-duplex mode.

### **Bit 8, LAN2 Port 2: Port Operating at Highest Supported Speed (RX3i CPE330/CPE400/CPL410 Embedded)**

#### **LAN1 Port 2 (ETM001-jx)**

This bit is set to 1 when the port is operating at its highest supported speed.

**Bit 9, Any Channel Error (All models)**

This bit (normally 0) indicates one or more of the channels are in error.

**Bit 10, LAN2 Port 1 Link (RX3i CPE330/CPE400/CPL410 Embedded)**

This bit is set to 1 once the port has established a link. 0 indicates no link.

**Bit 11, LAN2 Port 2 Link (RX3i CPE330/CPE400/CPL410 Embedded)**

This bit is set to 1 once the port has established a link. 0 indicates no link.

**Bit 12, LAN2 OK (RX3i CPE330/CPE400/CPL410 Embedded)**

1 indicates that the LAN can be accessed by at least one port. 0 indicates no network access for LAN2.

**Bit 13, LAN OK (All models) (RX3i CPE330/CPE400/CPL410 Embedded LAN1 only)**

This bit is 1 as long as the Ethernet interface software can communicate on the network. If the network becomes inaccessible due to local or network problems, this bit is set to 0. If LAN communication becomes possible again, it is set to 1.

**Bit 14, Resource Problem (All models)**

This bit is set to 1 if the Ethernet interface software has a resource problem (i.e., lack of data memory). The bit is reset to 0 on a subsequent PLC sweep. The Ethernet interface may or may not be able to continue functioning, depending on the severity of the problem. Look in the Controller Fault Table for details. Besides, the Station Manager STAT B and LOG commands can be used. Refer to the *PACSystems TCP/IP Ethernet Communications Station Manager User Manual*, GFK-2225, for more information.

**Bit 15, Module Over-Temperature (RX3i Rack-Based)**

This bit is set if the Ethernet interface hardware has detected that the internal temperature has exceeded normal limits. The bit is cleared when the internal temperature has not exceeded normal limits or has recovered from an over-temperature condition. (Overtemperature indication is available only in the RX3i rack-based Ethernet interface.)

**Bit 16, LAN interface OK Bit (All models)**

This bit is set to 1 by the Ethernet interface each PLC scan. If the Ethernet interface cannot access the PLC, the CPU sets this bit to 0. *When this bit is 0, all other Ethernet interface Status bits are invalid.*

## 12.6.2 Channel Status Bits

The Channel Status bits provide runtime status information for each communication channel. Each channel has two status bits; the meaning of the channel status bits depends upon the type of communication performed on that channel.

### Modbus TCP Client Channels

Each Modbus channel has a dedicated status bit:

**Bits 17, 19, 21 ... 79 (odd-numbered bits), Connection Open Bit (RX3i Ethernet Module)**

**Bits 17, 19, 21 ... 47 (odd-numbered bits), Connection Open Bit (RX3i Embedded Ethernet Interfaces)**

**Bits 17, 19, 21 ... 31 (odd-numbered bits), Connection Open Bit (RSTi-EP Embedded)**

This bit is 1 when a TCP connection exists for the associated channel. The bit is 0 when the connection does not exist or is unused (either never created or has disconnected). The bit is also set to zero when the controller goes to STOP because all connections are automatically closed upon STOP transition.

**Bits 18, 20, 22 ... 46, 48–80 (even-numbered bits), Reserved (All models)**

When a Channel is used as a Modbus TCP Channel, these bits are not used.

### SRTP Client Channels

Each SRTP channel has two status bits: a Data Transfer bit and a Channel Error bit:

**Bits 17, 19, 21 ... 79 (odd-numbered bits), Data Transfer Bit (RX3i Ethernet Module)**

**Bits 17, 19, 21 ... 47 (odd-numbered bits), Data Transfer Bit (RX3i Embedded Ethernet Interfaces)**

**Bits 17, 19, 21 ... 31 (odd-numbered bits), Data Transfer Bit (RSTi-EP Embedded)**

Typically, a channel is used to perform repetitive reads or writes. The Data Transfer bit pulses (0 □ 1 □ 0) each time there is a successful read or write. This can be an indicator of the ladder program to move the most recent data to another location.

This bit is *not* closely synchronized in time with the transfer. It indicates only that a transfer has occurred during the preceding read or write period. A rising edge on the bit indicating that a transfer has been completed does not guarantee that the next transfer has not begun or completed.

After an Establish Channel command, the COMMREQ status word (CSW) is always updated *before* the Data Transfer bit is set to 1. The Data Transfer bit for a channel is not meaningful until the Ethernet interface updates the CSW. Do not use data received from a server until the CSW confirming the Read command for that channel is 1 and the Data Transfer bit goes to 1.

**Bits 18, 20, 22 ... 80, (even-numbered bits), Channel Error Bit (RX3i Ethernet Module)**

**Bits 18, 20, 22 ... 48, (even-numbered bits), Channel Error Bit (RX3i Embedded Ethernet Interfaces)**

**Bits 18, 20, 22 ... 32, (even-numbered bits), Channel Error Bit (RSTi-EP Embedded)**

This bit (normally 0) is the primary indicator for an error on a channel. It indicates any channel error, fatal or non-fatal. It does not necessarily indicate that the channel is idle.

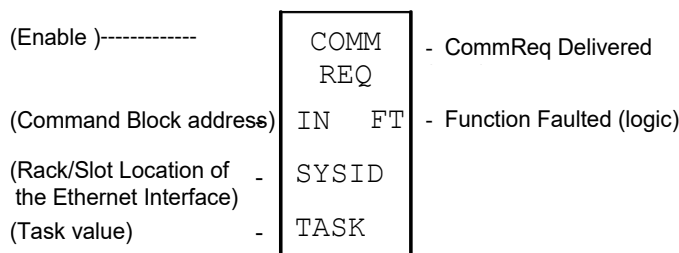
A Channel Error bit is not meaningful until the Ethernet interface has updated the COMMREQ status word confirming the Read or Write command for that channel. For an Establish Channel command, the COMMREQ status word is updated before the Channel Error bit is set to 1.

- A Channel Error bit is set to 1 when an error is detected on the channel. It is set to 0 when the channel is initially established and if the channel resumes normal operation after a transient error condition subsides. The Channel Error bit is also set to 0 when the channel is aborted by an Abort Channel command or when the CPU transitions from RUN to STOP. In the case of an Establish Channel command, the COMMREQ status word is always updated *before* the Channel Error bit is set to 1.
- If this bit indicates an error, initiate the Abort command, and then reinitiate the Read or Write command. If the error persists, initiate the Retrieve Detailed Channel Status command to find out if the channel is idle, and possibly why it is idle. The status code may change between the time the Channel Error bit indicates an error and the time the Retrieve Detailed Channel Status command retrieves the code.

## 12.7 Monitoring the FT Output of the COMMREQ Function Block.

The COMMREQ function block indicates its status through its FT output:

**Figure 103: Monitoring FT Output in COMMREQ Function Block**



If after executing a COMMREQ Function, the FT Output is ON, there is a programming error in one or more of the following areas.

- Invalid rack/slot specified. The module at this rack/slot is unable to receive a COMMREQ Command Block.
- Invalid Task ID. This value should always be 65536 decimal (10000H) for the CPU Ethernet daughterboard, or 0 decimal (0000H) for the Ethernet module.
- Invalid Data Block length (0 or greater than 128).

This output also may indicate that no more COMMREQ functions can be initiated in the ladder program until the Ethernet interface has time to process some of the pending COMMREQ functions.

If the FT Output is set, the CPU did not transfer the Command Block to the Ethernet interface. In this case, the other status indicators are not updated for this COMMREQ. The Ethernet interface is unable to return a COMMREQ Status Word to the PLC logic application.



## 12.8 Monitoring the COMMREQ Status Word

Every COMMREQ Command Block instruction specifies a 1-word memory address to receive status information about the execution of the command.

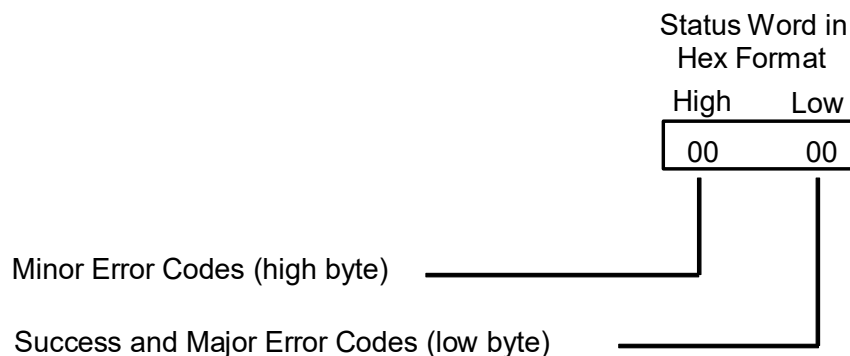
Before executing a COMMREQ for the Ethernet interface, the application program logic should the associated status word zero (for example, using a MOVE Word instruction). After executing a COMMREQ, the program should monitor its status word. If the status word is updated to a one (1), the command has been processed successfully. If the status word is updated to a value other than 1, an error has occurred. Any data returned by that command should not be used until the problem is corrected and the status word indicates success. It is critical to monitor the COMMREQ status word for each COMMREQ function.

If after executing a COMMREQ function, the COMMREQ status word is zero (0), the success Output is ON and the FT Output is OFF, the Command Block has been sent to the Ethernet interface, but no status has been returned. If this condition persists, check the Controller Fault Table for information.

### 12.8.1 Format of the COMMREQ Status Word

Displaying the status word in hexadecimal form makes it easier to differentiate the high and low bytes. This can be done using a MOVE WORD function block to display the hexadecimal value within the ladder program.

**Figure 104: Decoding the COMMREQ Status Word**



The following tables list the error codes that are reported in the COMMREQ Status word after the execution of a COMMREQ function.

## 12.8.2 Major Error Codes in the COMMREQ Status Word

Success or a Major Error Code appears in the low byte of the COMMREQ Status Word. Hexadecimal values for the low byte are listed below. For many Major Error Codes, additional information appears as a Minor Error Code in the high byte of the COMMREQ Status Word. Hexadecimal values for the high byte are listed on the following pages.

Error Status (Hexadecimal)	Major Error Code Description
01H	Successful Completion. (This is the expected completion value in the COMMREQ Status word.)
02H	Insufficient Privilege at server PLC. For a PACSystems or Series 90-70 server PLC, the minor error code contains the privilege level required for the service request.
04H	Protocol Sequence Error. The server CPU has received a message that is out of order. <i>Contact Technical Support for assistance.</i>
05H	Service Request Error at server PLC. The minor error code contains the specific error code. See the following table of Minor Error codes.
06H	Illegal Mailbox Type at server PLC. Service request mailbox type is either undefined or unexpected. <i>Contact Technical Support for assistance.</i>
07H	The server PLC CPU's Service Request Queue is full, usually due to heavy CPU loading. The client should retry later. It is recommended that the client wait a minimum of 10 milliseconds before sending another service request.
0BH	Illegal Service Request. The requested service is either not defined or not supported at the server PLC. (This value is returned instead of the actual service request error (01H), to avoid confusion with the normal successful COMMREQ completion.) <i>Contact Technical Support for assistance.</i>
11H	SRTP Error Code at the server. An error was detected at the SRTP server. See the following table of Minor Error codes.
82H	Insufficient Privilege at client PLC. The minor error code contains the privilege level required for the service request.
84H	Protocol Sequence Error. The CPU has received a message that is out of order. <i>Contact Technical Support for assistance.</i>
85H	Service Request Error at the client PLC. The minor error code contains the specific error code. See the following table of Minor Error codes.
86H	Illegal Mailbox Type. Service request mailbox type is either undefined or unexpected. <i>Contact Technical Support for assistance.</i>
87H	The client PLC CPU's Service Request Queue is full. The client should retry later. It is recommended that the client wait a minimum of 10 milliseconds before sending another service request.
8BH	Illegal Service Request. The requested service is either not defined or not supported. (This value is returned instead of the actual service request error (01H), to avoid confusion with the normal successful COMMREQ completion.) <i>Contact Technical Support for assistance.</i>
90H	Client (Channels) error. See the following table of Minor Error codes. (Some EGD command errors also use major code 90 when indicating the same error condition as channels.)
91H	Modbus/TCP error code at server. An error was detected at the Modbus/TCP server. See the following table of Minor Error codes.
A0H	EGD Command error. See the following table of Minor Error codes.

## 12.8.3 Minor Error Codes for Major Error Codes 05H (at Remote Server PLC) and 85H (at Client PLC)

Error Status (Hexadecimal)		Minor Error Code Description
Remote Server	Client	
8F05H	8F85H	Session already exists.
8E05H	8E85H	Memory write is prohibited.
9005H	9085H	Invalid PLC memory reference range.
9305H	9385H	Text buffer length/count does not agree with request parameters.
C105H	C185H	Invalid block state transition.
C305H	C385H	Text length does not match the traffic type.
C605H	C685H	Control Program (CP) tasks exist but the requestor is not logged into the main CP.
C705H	C785H	Passwords are set to inactive and cannot be enabled or disabled.
C805H	C885H	Password(s) already enabled and cannot be forced inactive.
C905H	C985H	Login using a non-zero buffer size required for block commands.
CA05H	CA85H	The device is write-protected.
CB05H	CB85H	A comm or write verify error occurred during save or restore.
CC05H	CC85H	Data stored on the device has been corrupted and is no longer reliable.
CD05H	CD85H	An attempt was made to read a device but no data has been stored on it.
CE05H	CE85H	The specified device has insufficient memory to handle the request.
CF05H	CF85H	The specified device is not available in the system (not present).
D105H	D185H	Packet size or total program size does not match the input.
D205H	D285H	Invalid write mode parameter.
D505H	D585H	Invalid block name specified.
D605H	D685H	Total datagram connection memory exceeded.
D705H	D785H	Invalid datagram type specified.
D805H	D885H	Point length is not allowed.
D905H	D985H	Transfer type invalid for this Memory Type selector.
DA05H	DA85H	Null pointer to data in Memory Type selector.
DB05H	DB85H	Invalid Memory Type selector in a datagram.
DC05H	DC85H	Unable to find a connection address.
DD05H	DD85H	Unable to locate given datagram connection ID.
DE05H	DE85H	Size of datagram connection invalid.
DF05H	DF85H	Invalid datagram connection address.
E005H	E085H	Service in the process cannot log in.
E405H	E485H	Memory Type for this selector does not exist.
E905H	E985H	Memory Type selector not valid in context.
EA05H	EA85H	Not logged in to process the service request.
EE05H	EE85H	Could not return block sizes.
EF05H	EF85H	The programmer is already attached.
F005H	F085H	Request only valid in stop mode.
F105H	F185H	Request only valid from the programmer.

Error Status (Hexadecimal)		Minor Error Code Description
Remote Server	Client	
F205H	F285H	The invalid program cannot log in.
F405H	F485H	Invalid input parameter in the request.
F505H	F585H	Invalid password.
F605H	F685H	Invalid sweep state to set.
F705H	F785H	Required to log in to a task for service.
F805H	F885H	Invalid program name referenced.
F905H	F985H	Task address out of range.
FC05H	FC85H	I/O configuration is invalid.
FE05H	FE85H	No privilege for attempted operation.
FF05H	FF85H	A service request has been aborted.

## 12.8.4 Minor Error Codes for Major Error Code 11H (at Remote Server PLC)

Error Status (Hex)	SRTP Error Code Description
0111H	Generic SRTP error.
0211H	The PLC is inaccessible.
0311H	Reserved.
0411H	Unexpected SRTP version encountered in the received message.
0511H	Unrecognized SRTP message received.
0611H	Data present in SRTP message, which should not contain data.
0711H	Generic resource problem detected.
0811H	SRTP message encountered in inappropriate connection state.
0911H	Generic refusal by backplane driver to handle the request.
0A11H	Recognized but unsupported SRTP message received.
0B11H	Lost transaction in server.
0C11H	Error sending SRTP PDU to the client PLC.
1411H	Unable to allocate a text buffer from dual-port memory.
1711H	Invalid text length detected in a mailbox message.
1811H	An invalid number of destinations detected in a mailbox message.
1911H	Invalid source detected in a mailbox message.
1A11H	Invalid slot number detected in a mailbox message.
1B11H	Invalid rack number detected in a mailbox message.
1D11H	Bad text buffer address in dual-port memory.
2111H	Unable to find control data required to send a mailbox message to the PLC.
2211H	Timed out waiting for the availability of mail communications with the PLC.
2311H	Invalid task ID detected while attempting to send a mailbox message to the PLC.

Error Status (Hex)	SRTP Error Code Description
2411H	Unable to send mailbox message to PLC because the mail queue is full.
2611H	Unable to communicate with PLC.
2711H	Backplane driver not initialized or unable to acquire a dual-port memory semaphore.
2A11H	The backplane driver could not access the PLC.
2B11H	Invalid binding on the message sent to the backplane driver.
2C11H	The message could not be sent to its destination because the mailbox was not open.
2D11H	The maximum number of transfers to the destination is already taking place.
2E11H	The maximum number of transfers of this transfer type is already taking place.
2F11H	Cannot obtain a backplane transfer buffer.
3011H	Cannot obtain resources other than backplane transfer buffers.
3111H	Connection ID or block transfer ID is not valid.
3211H	Timed out waiting for PLC CPU response.
3311H	The PLC CPU aborted the request.
3411H	An invalid message type was specified.
3511H	The specified task is not registered.
3611H	The mailbox offset specified is invalid.
3711H	The backplane task could not be registered because the message response handler was not specified.
3811H	The backplane task could not be registered because the unsolicited mailbox message handler was not specified.
3911H	The backplane task could not be registered because a required parameter was not specified.
3A11H	More than the allowable byte length in a single transfer.
3B11H	The bad sequence number in the request.
3C11H	Invalid command in the request.
3D11H	Response length does not match the length specified in the response qualifier.
3E11H	Request failed because the PLC's Service Request Processor is not initialized.
3F11H	Request failed due to an error in the remote device, most likely running out of Dual-Port RAM text buffers.
4011H	Unable to free dual-port memory that was allocated for a connection or block transfer area.
4111H	The backplane task could not be registered because the service request handler was not specified.
4211H	No dual-port memory was allocated for the connection or block transfer area needed to process the request.

Error Status (Hex)	SRTP Error Code Description
4311H	Failure to register with backplane driver because the requested task is already registered.
4411H	Request failed because an invalid field was identified in the request mailbox qualifier.
E811H	Unable to send the request to the PLC because an internal message queue is full.
E911H	Unable to send the request to the PLC because the text buffer type is invalid.
EA11H	Unable to send the request to the PLC because the mailbox utility function is invalid.
EB11H	Unable to send the request to the PLC because the mailbox message is not specified.
EC11H	Unable to send the request to the PLC because the internal message queue is not initialized.
FE11H	Request failed due to mailbox error on the remote device. The remote device log will have more information.
2911H	The backplane driver is not initialized.
2A11H	The backplane driver could not access the PLC.
2F11H	Request failed due to an invalid parameter detected in the remote device. The remote device log will have more information.
3011H	The specified task is not registered.
3111H	Failure to register with backplane driver because the requested task is already registered.
3211H	Unable to find the resource necessary for backplane driver to process a service request.
3311H	Bad sequence number detected in the service request because it is already in use.
3411H	Invalid data detected that prevents backplane driver from completing a request.
3611H	More than the allowable byte length in a single transfer.
4811H	Memory resource problem detected.
4911H	Network buffer resource problem detected.
4C11H	Error detected while attempting to receive mailbox messages from the PLC.
4D11H	Timed out waiting to obtain a backplane transfer buffer.
4E11H	Timed out waiting to transfer a mailbox message to the PLC.
4F11H	Timed out waiting for PLC CPU response.

## 12.8.5 Minor Error Codes for Major Error Code 90H (at Client PLC)

Error Status (Hex)	Error Description
0190H	Timeout expired before transfer completed; still waiting on transfer.
0290H	Period expired before transfer completed; still waiting on transfer. Or, an attempt to establish a TCP connection with a Remote Server through a gateway has failed. Check the following: <ul style="list-style-type: none"> <li>• Make sure the Server is turned on.</li> <li>• Make sure cables are connected.</li> <li>• Make sure the Gateway is reachable from both the client and the server.</li> <li>• Make sure the switch is turned on when using a switch.</li> </ul>
8190H	COMMREQ data block too short for the command.
8290H	COMMREQ data block too short for server PLC node address.
8390H	Invalid server memory type.
8490H	Invalid Program Name.
8590H	Invalid Program Block Name.
8690H	Zero server unit length is not allowed.
8790H	The server unit length is too large.
8890H	Invalid channel number.
8990H	Invalid time unit for the period. (Maximum permitted 3965 hours)
8A90H	The period value is too large.
8B90H	Zero server memory starting address is not allowed.
8C90H	Invalid client memory type.
8D90H	Invalid server host address type.
8E90H	Invalid IP address integer value. (Must be 0–255)
8F90H	Invalid IP address class. (Must be valid Class A, B, or C IP address) May also occur if the destination IP address in the COMMREQ is the same as the sender's IP address.
9090H	Insufficient TCP connection resources to do requests.
9190H	Zero local starting address is not allowed.
9290H	Address length value invalid. Must be 4 for address type 1.
9390H	COMMREQ data block too short for Program Block name (including 0 pad).
9490H	COMMREQ data block too short for Program name (including 0 pad).
9590H	Internal API error. See Controller Fault Table or exception log for details. This problem may occur due to the Ethernet interface being asked to perform beyond its capacity. Try transferring less data per message or establishing fewer simultaneous connections.
9690H	Underlying TCP connection aborted (reset) by the server endpoint.
9790H	Underlying TCP connection aborted by the client endpoint.
9890H	The remote server has no Service Request Processor.

Error Status (Hex)	Error Description
9A90H	Response to session request did not arrive in proper order.
9B90H	Session denied by server PLC.
9C90H	Data response did not arrive in proper order.
9D90H	Data response had an unexpected size.
9E90H	Unrecognized COMMREQ command code, an invalid COMMREQ status word address, or the Ethernet interface is not connected to the network. Check the network cable and the connection to the network hub or switch.
A190H	Invalid CRS word memory type.
A290H	Failed an attempt to update the CRS word.
A390H	<i>Reserved.</i>
A490H	<i>Reserved.</i>
A590H	<i>Reserved.</i>
A690H	Invalid bitmask.
A790H	Unable to connect to the remote device.
A890H	Channel Resources in Use. Try the command again; a resource will become available.
A990H	“Establish Read/Write/Send Info Report Channel” COMMREQ was received while an Abort was in progress.
AA90H	An attempt to establish a TCP connection with a Remote Server has failed. Check the following: <ul style="list-style-type: none"> <li>▪ Make sure the Server is turned on.</li> <li>▪ Make sure cables are connected.</li> <li>▪ Make sure the switch is turned on if using a switch.</li> </ul>
AB90H	A COMMREQ was discarded because the application program issued the COMMREQ before the COMMREQ Status Word for the previous COMMREQ was set.
AC90H	A protocol error occurred while communicating with the local PLC.
AD90H	A TCP Timeout occurred while communicating with the Remote PLC.
AE90H	A protocol error occurred while communicating with the local PLC.
B490H	The channel that the application is trying to open is already open.
B590H	The channel the application is trying to access is owned by a different protocol.
B690H	COMMREQ specified an invalid Modbus function code.
B790H	COMMREQ specified an invalid Modbus unit ID.
B890H	COMMREQ specified an invalid number of subrequests.
B990H	A COMMREQ subrequest specified an invalid record number.
C090H	(Redundancy only) COMMREQs commands are not allowed when the Redundant IP address is not active at this Ethernet interface.
FF90H	Abort in Progress on a Channel



## 12.8.6 Minor Error Codes for Major Error Code 91H (at Remote Modbus/TCP Server)

The Minor codes for Major Error Code 91H indicate standard Modbus exception codes returned from the remote Modbus/TCP server/slave device. (These Modbus exception codes are taken from Modbus Application Protocol V1.1b, December 28, 2006.)

Error Status (Hex)	Error Description
0191H	Illegal function. The function code received in the query is not an allowable action for the server. (Modbus exception code 01 ILLEGAL FUNCTION)
0291H	Illegal Data Address. The data address received in the query is not an allowable address for the server. The combination of the reference number and transfer length is invalid. (Modbus exception code 02 ILLEGAL DATA ADDRESS)
0391H	Illegal Data Value. A value in the query field is not an allowable value for the server. This indicates a fault in the remainder of the request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in the server has an incorrect value. (Modbus exception code 03 ILLEGAL DATA VALUE)
0491H	Slave Device Failure. An unrecoverable error occurred while the server was attempting to perform the requested action. (Modbus exception code 04 SLAVE DEVICE FAILURE)
0591H	Acknowledge. Used for Programmer operations only. Our Modbus/TCP server does not support Modbus programmer operations. (Modbus exception code 05 ACKNOWLEDGE)
0691H	Slave Device Busy. The server is unable to accept and process handle this Modbus request. (Modbus exception code 06 SLAVE DEVICE BUSY)
0791H	Negative Acknowledge. An internal server error occurred while attempting to process a Modbus request. (Modbus exception code 07 NEGATIVE ACKNOWLEDGE)
0891H	Memory Parity Error. (Function codes 20 and 21 only.) The extended file area failed to pass a consistency check. (Modbus exception code 08 MEMORY PARITY ERROR)
0991H	Reserved. (Modbus exception code 09 RESERVED)
0A91H	Gateway Path Unavailable. Gateway was unable to allocate a PATH to process the request. Usually means the gateway is misconfigured or overloaded. (Modbus exception code 10 GATEWAY PATH UNAVAILABLE)
0B91H	Gateway Target No Response. No response was obtained from the target device. Usually means that the device is not present on the network. (Modbus exception code 11 GATEWAY TARGET NO RESPONSE)

## 12.8.7 Minor Error Codes for Major Error Code A0H (at Client PLC)

Error Status (Hex)	Error Description
01A0H	Remote exchange is not healthy.
02A0H	Remote exchange is not defined.
03A0H	The remote exchange signature does not match.
04A0H	Request data length is invalid.
05A0H	Response data length is invalid.
06A0H	Invalid memory type selector or address range at the remote device.
07A0H	Password protection does not permit access to a remote device.
08A0H	Attempt to write to a consumed exchange; this is not permitted.
09A0H	Internal resource error at a remote device (memory allocation failed, etc.)
0AA0H	Message delivery error; command was not processed.
0BA0H	Software initialization error; command was not processed.
0CA0H	Invalid RDS session was specified.
0DA0H	Data buffer length is invalid.
0EA0H	The invalid response message from a remote device.
0FA0H	The address type is not supported on a remote device.
10A0H	A memory access error occurred while processing this command.
11A0H	The remote device did not understand the request.
12A0H	The remote device has no variable defined at the specified address.
13A0H	An attempt was made to write a Read-Only variable at the remote device.
14A0H	Data length or contents are invalid for transfer according to the data type of that variable at the remote device.
15A0H	Response message would exceed max response size (1400 bytes).
50A0H	The remote server detected an unsupported protocol version in the request.
51A0H	The remote server did not recognize the requested command.
52A0H	The remote server detected a configuration time mismatch in the request.
53A0H	The remote server detected that the request was not a valid RDS message. The RDS_Header bit (required by RDS version 2.01 and higher) was not set.
54A0H	Attempt to establish a second session to a remote server. Only one session at a time is permitted between this device and each remote server.
55A0H	All available RDS sessions are currently in use. (The number of simultaneous RDS sessions is limited to a maximum of 10.)
56A0H	EGD signature mismatch amid a run mode store. Retry your COMMREQ after updates to the target device are complete.

## 12.9 Using the EGD Management Tool (RX3i Ethernet Module)

The EGD Management Tool can perform online monitoring of EGD class 2 devices such as the PACSystems Ethernet interfaces. It can quickly look at the Ethernet Global Data traffic across an entire network of EGD devices to spot problems. To use the EGD Management Tool, you must have configured Ethernet Global Data using the EGD Configuration Server option as described in Section 4.

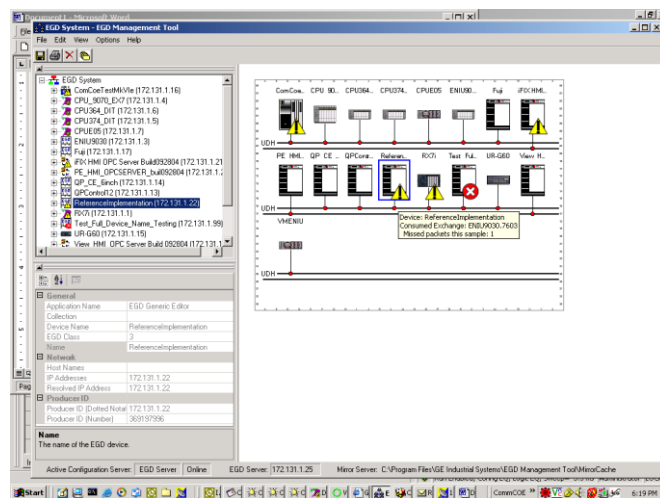
### 12.9.1 Installing the EGD Management Tool

The EGD Management Tool is not automatically installed when you install the Programmer. To install the EGD Management Tool, look in the directory where you installed the programmer and you will find a subdirectory named *EGD Installs*. In that directory, you will find a file named *EgdManagementToolSetup.msi*. Double-click on this file to install the EGD Management Tool.

### 12.9.2 Launching the EGD Management Tool

To run the EGD Management Tool, select the Ethernet Global Data node in the Navigator and right click. Select *Launch EGD Management Tool*. The EMT will begin execution in a separate frame on your desktop.

**Figure 105: EGD Management Tool Screenshot**



The right side of the screen shows a graphical representation of the EGD network based on the configuration data stored in the EGD Configuration Server. EGD collections are displayed as a folder icon. The navigator on the left side allows specific devices, exchanges, and variables in the configuration to be examined. Properties for these elements are shown in the property pane at the lower left.

The EGD Management Tool displays devices and networks based on the configuration information in the EGD Configuration Server for the machine it is running on. Using the options menu, you can configure the server information much as you do for the programming tool, and also set options

for the online operation of the tool. Be aware that changing the server configuration will change it for all tools running on that machine, including the programming software.

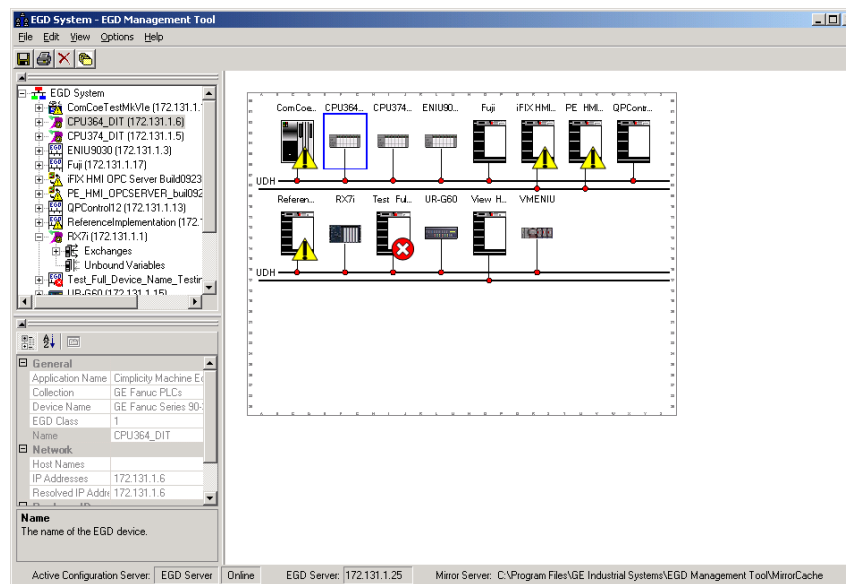
In addition to the online operations described below, the EGD Management Tool has a number of offline capabilities (such as View/Reports) for analyzing the Ethernet Global Data configuration. See the EGD Management Tool help for more information.

### 12.9.3 Monitoring EGD Devices

The EGD Management Tool monitors the devices on the Ethernet Global Data network provided it has access to that network. To have access to the EGD network, the computer running the EGD Management Tool must have a Network Interface Card that connects to the EGD network. Consult with your local network administrator if you need help connecting the computer to the Ethernet Global Data network.

The screen below shows the EGD Monitoring Tool connected to and monitoring an EGD network.

**Figure 106: EGD Monitoring Tool Monitoring EGD Network**



Devices that have a red 'x' are not responding to communications from the EGD Management Tool. Devices that have a yellow triangle have some kind of error or warning condition that may require attention. Use the browser pane to select the device to get further information about the failures being reported. The EGD Management Tool reports a configuration mismatch for PLCs that have multiple Ethernet interfaces. Only one of the interfaces in a PLC is queried by the EGD Management Tool, so only a subset of the exchanges in the PLC is visible online through that interface.

Online information is only available for EGD Class 2 devices (devices that support the EGD commands). This includes all PACSystems controllers.

When the EGD Management Tool is used online, it periodically sends Ethernet Global Data commands to each device. This may have a performance impact on the network and the devices on the network. Before using the EGD Management Tool in a production environment, be sure to assess the performance impact of its use on your application.

## 12.9.4 Monitoring Status of Ethernet Global Data for a Device

The EGD Management Tool can display detailed information for each exchange in an EGD Class 2 device such as a PACSystems controller. Selecting the Exchanges node for the device in the navigator pane will display the list of exchanges for the device.

### Configuration Summary

Selecting the *Configuration Summary* tab displays information about the exchanges defined in the device.

**Figure 107: EGD Management Tool Displaying EGD Exchange Information**

The screenshot shows the EGD Management Tool interface. On the left is a tree view with 'Exchanges' selected. The main area displays a table with the following columns: Exchange, Producer ID, Destination, Mode, Type, Configuration Time, Signature, Length, and Period. The table contains 20 rows of data for various exchanges.

Exchange	Producer ID	Destination	Mode	Type	Configuration Time	Signature	Length	Period
ReferenceImplementation.122	172.131.1.22	172.131.1.1	Unicast	Producer	2004-10-15 19:47:31	1.1	100	500
ReferenceImplementation.222	172.131.1.22	172.131.1.2	Unicast	Producer	2004-10-15 19:49:40	4.0	100	500
ReferenceImplementation.522	172.131.1.22	172.131.1.5	Unicast	Producer	2004-10-15 19:49:40	3.0	100	500
ReferenceImplementation.622	172.131.1.22	172.131.1.6	Unicast	Producer	2004-10-15 19:49:40	3.0	100	500
ReferenceImplementation.2122	172.131.1.22	172.131.1.21	Unicast	Producer	2004-10-15 19:49:40	3.0	100	500
ReferenceImplementation.5122	172.131.1.22	224.0.7.22	Multicast	Producer	2004-09-16 17:14:09	1.0	2	500
ReferenceImplementation.7622	172.131.1.22	172.131.255.255	Broadcast	Producer	2004-09-16 17:14:09	1.0	2	500
ReferenceImplementation.9922	172.131.1.22	172.131.1.1	Unicast	Producer	2004-10-15 15:23:08	2.1	3	500
RX7i.7601	172.131.1.1	172.131.255.255	Broadcast	Consumer	2004-09-09 20:13:37	2.0	2	500
RX7i.7602	172.131.1.1	172.131.255.255	Broadcast	Consumer	2004-09-09 20:13:37	1.1	2	500
RX7i.5101	172.131.1.1	224.0.7.1	Multicast	Consumer	2004-09-09 20:13:37	1.0	2	500
RX7i.5102	172.131.1.1	224.0.7.2	Multicast	Consumer	2004-09-09 20:13:37	1.1	2	500
ENIU9030.7603	172.131.1.3	172.131.255.255	Broadcast	Consumer	2004-08-10 15:00:34	1.0	56	500
CPU374_DIT.7605	172.131.1.5	172.131.255.255	Broadcast	Consumer	2004-09-08 20:42:25	3.1	2	500
CPU374_DIT.5105	172.131.1.5	224.0.7.5	Multicast	Consumer	2004-09-08 20:42:25	3.1	2	500
CPU364_DIT.7606	172.131.1.6	172.131.255.255	Broadcast	Consumer	2004-09-08 20:41:24	3.1	2	500
CPU364_DIT.5106	172.131.1.6	224.0.7.6	Multicast	Consumer	2004-09-08 20:41:24	3.1	2	500
VMENIU.1	172.131.1.8	172.131.255.255	Broadcast	Consumer	2004-08-20 19:42:59	0.0	40	500
UR-G60.3	172.131.1.15	172.131.255.255	Broadcast	Consumer	2004-07-06 15:44:38	16.0	22	1000
ComCoeTestMkVle.7616	172.131.1.16	172.131.255.255	Broadcast	Consumer	2004-02-16 11:16:39	1.1	2	480
Fuji.1	172.131.1.17	172.131.255.255	Broadcast	Consumer	2004-02-16 11:16:39	0.0	6	500
Fuji.2	172.131.1.17	172.131.255.255	Broadcast	Consumer	2004-02-16 11:16:39	0.0	56	5000

The configuration summary data for each exchange has the following information:

**Exchange** –The name of the exchange as it is stored in the EGD configuration server

**Producer ID** –The producer ID of the exchange as it is stored in the EGD configuration server

**Destination** –The destination IP address for the exchange

**Mode** – *Unicast*, *Multicast*, or *Broadcast* based on the mode of the exchange

**Type** – *Producer* or *Consumer* depending on the type of the exchange

**Configuration Time** –The configuration timestamp of the exchange as it is stored in the EGD configuration server

**Signature** –The signature value of the exchange as it is stored in the EGD configuration server

**Length** –The byte size of the exchange as it is stored in the EGD configuration server

**Period** –The production period for a produced exchange or the consume timeout for a consumed exchange as it is stored in the EGD configuration server

## Online EGD Statistics

Selecting the *Online Statistics* tab displays a list of the exchanges in the device and statistical information about each exchange. The statistics are updated periodically based on a rate in the Options menu.

**Figure 108: EGD Management Tool Displaying EGD Statistics**

Exchange	Configuration Time	Due Time	Status	Length	Message Count	Missed Count	Refresh Errors
ReferenceImplementation.122	2004-10-15 19:47:31	2004-10-17 22:20:40	Producing	100	10656	0	0
ReferenceImplementation.222	2004-10-15 19:49:40	2004-10-17 22:20:40	Producing	100	10656	0	0
ReferenceImplementation.522	2004-10-15 19:49:40	2004-10-17 22:20:40	Producing	100	10656	0	0
ReferenceImplementation.622	2004-10-15 19:49:40	2004-10-17 22:20:40	Producing	100	10656	0	0
ReferenceImplementation.2122	2004-10-15 19:49:40	2004-10-17 22:20:40	Producing	100	10656	0	0
ReferenceImplementation.5122	2004-09-16 17:14:09	2004-10-17 22:20:40	Producing	2	10656	0	0
ReferenceImplementation.7622	2004-09-16 17:14:09	2004-10-17 22:20:40	Producing	2	10656	0	0
ReferenceImplementation.9922	2004-10-15 15:23:08	2004-10-17 22:20:40	Producing	3	10656	0	0
RX7i.7601	2004-09-09 20:13:37	2004-10-17 22:20:40	Healthy	2	20948	120166	2
RX7i.7602	2004-09-09 20:13:37	2004-10-17 22:20:40	Healthy	2	20824	120290	2
RX7i.5101	2004-09-09 20:13:37	2004-10-17 22:20:40	Healthy	2	20976	120138	2
RX7i.5102	2004-09-09 20:13:37	2004-10-17 22:20:40	Healthy	2	20865	120249	2
ENIU9030.7603	2004-08-10 15:00:34	2004-10-17 22:20:40	Healthy	56	34823	50061	0
CPU374_DIT.7605	2004-09-08 20:42:25	2004-10-17 22:20:40	Healthy	2	21191	63815	0
CPU374_DIT.5105	2004-09-08 20:42:25	2004-10-17 22:20:40	Healthy	2	21170	63836	0
CPU364_DIT.7606	2004-09-08 20:41:24	2004-10-17 22:20:40	Healthy	2	21194	19897	0
CPU364_DIT.5106	2004-09-08 20:41:24	2004-10-17 22:20:40	Healthy	2	21175	19914	0
VMENU.1	2004-08-20 19:42:59	2004-10-17 22:20:40	Healthy	40	35126	37748	0
UR-G60.3	2004-07-06 15:44:38	2004-10-17 22:20:40	Healthy	22	10567	25723	0
ComCoeTestMkVle.7616	2004-02-16 11:16:39	2004-10-17 22:20:40	Healthy	2	22040	50945	0
Fuji.1	2004-02-16 11:16:39	2004-10-17 22:20:40	Healthy	6	10600	31649	0
Fuji.2	2004-02-16 11:16:39	2004-10-17 22:20:40	Healthy	56	1304	44937	0

The statistics data for each exchange has the following information:

**Exchange** –The name of the exchange as it is stored in the EGD configuration server

**Configuration Time** –The date and time that the configuration for the exchange was created

**Due Time** –The date and time that a sample is due. For a produced exchange, this is the time that the next sample will be produced. For a consumed exchange, this is the time at which the exchange will time out if data is not received

**Status** –Status variable of the exchange. For a produced exchange, the status will be Producing if the exchange is actively being sent to the network and Pending if the exchange is defined but not producing. A Pending status in a PACSystems exchange may indicate that the controller has its I/O disabled thus stopping the production of EGD. For a consumed exchange, the status will be Healthy if no timeout has occurred for the exchange and Unhealthy if the exchange is timed out

**Length** –The byte size of the data for the exchange

**Message Count** –The number of samples transferred on the exchange

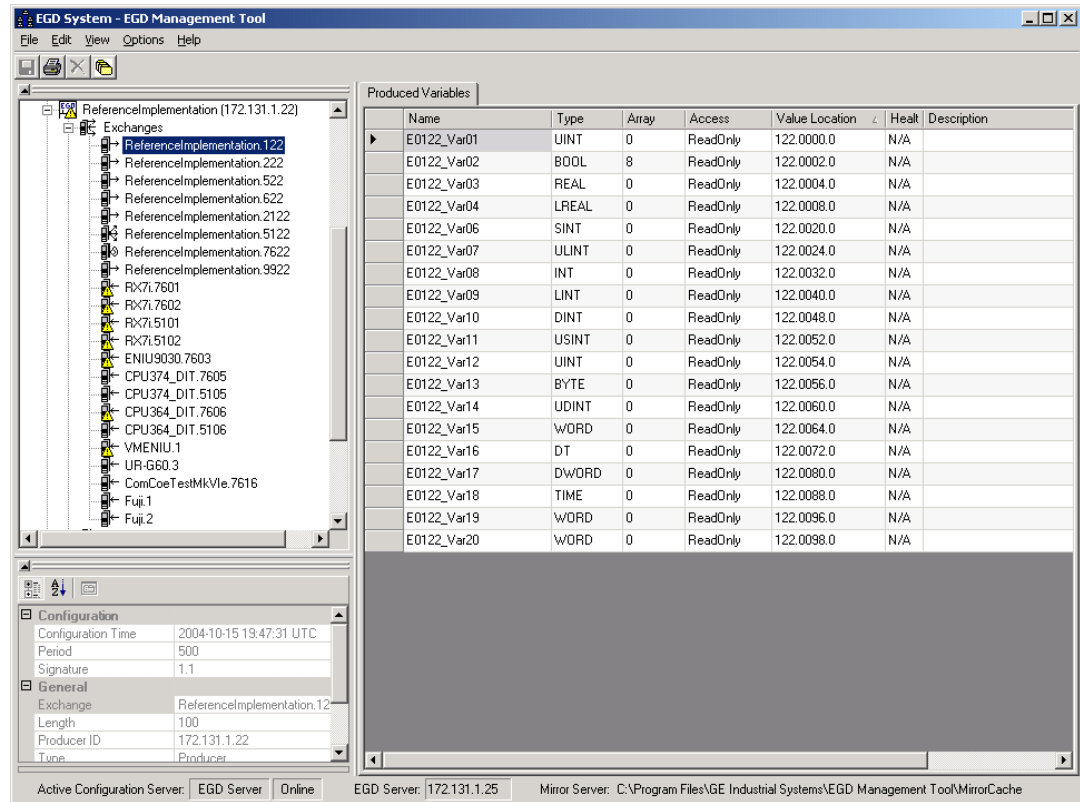
**Missed Count** –The number of samples that were missed on the exchange. Missed samples may indicate issues with the underlying Ethernet network or overloading of the consuming device

**Refresh Errors** –The number of timeouts that have occurred for a consumed exchange

## Produced Variables

Expanding the Exchanges node in the navigator pane displays the list of exchanges for the device as recorded in the EGD Configuration Server. Selecting an exchange brings up a list of variables for that exchange as shown below. This can be used to look at the details of the data for an exchange.

**Figure 109: EGD Management Tool Displaying List of Variables for an Exchange**



## 12.10 Troubleshooting Common Ethernet Difficulties

Some common Ethernet errors are described below. Ethernet errors are generally indicated in the Controller Fault Table and the Ethernet exception log. Controller Faults generated by the Ethernet interface contain Ethernet exception events within the extra fault data. Refer to the *PACSystems TCP/IP Ethernet Communications Station Manager User Manual*, GFK-2225 for detailed descriptions of Ethernet exception events.

### 12.10.1 COMMREQ Fault Errors

When the PLC CPU attempts to initiate COMMREQs to the Ethernet interface more rapidly than the Ethernet interface can accept them, the COMMREQ delivery will fail. The fault output of the COMMREQ function block will be set and the COMMREQ will not be delivered to the Ethernet interface. In this case, the PLC logic program should attempt to initiate the COMMREQ on another sweep after a very short delay. This condition may arise when the logic Program attempts to initiate greater than 16 COMMREQs in the same logic sweep.

Sustained heavy COMMREQ delivery from the PLC CPU to the Ethernet interface can use a considerable portion of the Ethernet interface's processing capability. Under heavy COMMREQ load, the Ethernet interface may discard some received COMMREQs until it is once again able to process further COMMREQs. In such cases, the Ethernet interface increments the "CmrqDscd" tally; this tally is available via the TALLY C Station Manager command.

Under sustained extremely heavy COMMREQ load, the Ethernet interface may not respond to Station Manager commands and possibly some network communications. A COMMREQ fault may be logged in the Controller Fault Table (see Controller Fault Table Descriptions, earlier in this chapter.) If this occurs, first switch the PLC CPU to STOP mode, which ceases COMMREQ delivery to resume normal Ethernet operation. Then modify the PLC logic application to reduce the COMMREQ traffic to a manageable level.

### 12.10.2 PLC Timeout Errors

PLC timeout errors may occur when the SRTP traffic to the Ethernet interface exceeds the PLC's ability to process the requests, or when the PLC is unable to deliver mail to the Ethernet interface. PLC Timeout errors will take down an SRTP Server connection. Therefore, the remote SRTP client must reestablish a new SRTP connection to the Ethernet interface.

This error is indicated in the Controller Fault Table as:

Backplane communication with controller fault; lost request  
with exception Event = 8, Entry 2 = 8

These errors may also be accompanied by any of the following:

Backplane communication with controller fault; lost request  
with exception Event = 8, Entry 2 = 6; location = Ethernet interface

LAN system-software fault; resuming  
with exception Event = 8, Entry 2 = 16; location = Ethernet interface

Non-critical CPU software event  
status code (bytes 5-8) = 80 3a 00 12; location = CPU module



The PLC Timeout condition occurs when the CPU cannot process requests within a specified timeout period. The remedy is to reduce the rate of requests or increase the processing capacity in the PLC.

Cause	Corrective Action
Heavy COMMREQ traffic	Reduce the rate at which the logic application sends COMMREQs to the Ethernet interface.
Heavy SRTP traffic	Reduce the size, number, or frequency of SRTP requests at the remote SRTP client.
Long PLC sweep time	Modify the PLC application to reduce the PLC sweep time.
PLC Communication Window set to LIMITED mode.	Change to RUN-TO-COMPLETION mode.

**Note:** The rack-based Ethernet modules use the Backplane Communications Window.

### 12.10.3 Application Timeout Errors

Application timeout errors include:

- SRTP Channel timeout errors (COMMREQ Status 0190H or 0290H at the client)
- EGD Command timeout errors (COMMREQ Status 0190H at the client)
- EGD consumed exchange refresh errors (Exchange Status 6 or 7).

Application timeout errors can happen for several reasons, including:

- Incorrect destination device address, or destination device not on the network. Verify that the destination device address is correct and that the destination device is functioning properly. Ping the destination device to check that it is present on the network.
- The network throughput cannot keep up with the traffic generated by the application. This condition can occur when the intervening network components between the application devices cannot handle the volume of network traffic, causing network packets to be dropped.  
For SRTP, this causes TCP retransmissions; repetitive retransmissions can slow the SRTP responses enough that the client detects an application timeout error.  
For EGD, this causes samples to be dropped. If the consumer misses enough samples, it detects a consumer timeout error; when that exchange subsequently receives samples, the consumer may detect a Data with Refresh error.  
This condition typically arises when intermediate network routers or switches lack the buffering or processing capacity to handle the network load. Reduce the volume of traffic on the network or identify and upgrade the network component(s) that are unable to handle the traffic volume. Consult your network administrator for assistance.
- The SRTP channel timeout and period include the time required to establish the TCP connection. It is important to consider the connection time when configuring these values. If more than one SRTP channel is being established and the PACSystems server has just been restarted or updated with a new *Hardware Configuration*, the channel timeout and period should be more than one second. This allows sufficient time for the high level of TCP traffic required to establish new network connections. When first establishing a channel, a channel timeout lower than one second may result in a 0190H (channel timeout) COMMREQ status and a channel period lower than one second may result in a 0290H (period expired error).

## 12.10.4 EGD Configuration Mismatch Errors

When using Ethernet Global Data, the produced exchange (defined at the producer) must agree with the consumed exchange (defined at the consumer). The consumer generates an error when the size of an exchange received from the network differs from the configured size for that consumed exchange.

This error is indicated in the Controller Fault Table as:

“LAN system-software fault; resuming”  
 with exception Event = 28, Entry 2 = 1d

As this error is generated each time the mismatched exchange is received, the Ethernet exception log can quickly fill up with mismatch error events.

Cause	Corrective Action
Producer and Consumer exchange definitions are of different size	Review the conflicting exchange definitions at the producer and the consumer. Change the incorrect exchange definition so that produced and consumed definitions are the same size.

If the consumer wishes to ignore certain portions of a consumed exchange, be sure that the length of the ignored portions is correct. The ignored portion is specified as a byte count.

## 12.10.5 Station Manager Lockout under Heavy Load

Sustained heavy EGD and/or SRTP Server load can utilize all processing resources within the Ethernet interface, effectively locking out the Station Manager function. The Station Manager appears inoperative under either local or remote operation. The Ethernet interface always gives higher priority to data communication functions than to the Station Manager. When the processing load is reduced, the Station Manager becomes operative once again.

This condition is not reported to the Controller Fault Table or Ethernet exception log.

## 12.10.6 PING Restrictions

To conserve network data buffer resources, the CPU process only one ICMP control message at a time. An ICMP Echo (ping) request that arrives while the CPU is processing another ICMP control message is discarded. When multiple remote hosts attempt to ping the CPU at the same time, some individual ping requests may be ignored depending upon the timing of the ping requests on the network.

The CPU may initiate ping requests to another host on the network via the “ping” Station Manager command. The ping request sequence is restricted to one remote host at a time.

Discarded ping requests are not reported to the Controller Fault Table or Ethernet exception log.

## 12.10.7 SRTP and Modbus/TCP Connection Timeout

When the Ethernet interface is abruptly disconnected from a remote SRTP or Modbus/TCP device (for example, by disconnecting the Ethernet cable), the underlying TCP connection attempts to re-establish communication. By default, the underlying TCP connection in the Ethernet interface remains open for seven minutes while TCP attempts to reconnect. During this interval, the SRTP or Modbus/TCP connection is unavailable. If all the SRTP or Modbus/TCP connections in the Ethernet interface are in use or otherwise unavailable, a new SRTP or Modbus/TCP server connection must wait until an existing SRTP or Modbus/TCP connection times out. If the SRTP server connection was used by the Programmer, any new Programmer connection is restricted to Monitor operation until the previous connection times out and is cleaned up.

**SRTP Inactivity Timeout.** This feature reduces the amount of time required to terminate and clean up an SRTP programmer connection to 20 – 30 seconds. The SRTP inactivity timeout is initially set by the “vconn\_tout” AUP parameter for programmer connections. The PME programmer can override this initial value. See 1.3.9, *SRTP Inactivity Timeout* for details.

If desired, the TCP connection timeout duration may be adjusted via AUP parameters. See *Configuring Advanced User Parameters* to configure and use AUP parameters. The TCP connection timeout interval (in seconds) is calculated as:

$$\text{TimeoutSeconds} = \text{wkal\_idle} + ( (\text{wkal\_cnt} + 1) \times \text{wkal\_intvl} )$$

For example, the following set of AUP parameters will establish the TCP connection timeout as 25 seconds:

```
wkal_idle = 10
wkal_cnt = 2
wkal_intvl = 5
```

Note that the TCP connection timeout interval applies to *all* TCP-based connections at this Ethernet interface. This includes all SRTP, Modbus/TCP, FTP, and (where supported) webserver communications. To allow for normal TCP reconnection, any adjusted TCP connection timeout must exceed the longest application data transfer interval.

The underlying TCP connection timeout is normal expected behavior and is consistent with our other PLC products.

## 12.10.8 Sluggish Programmer Response after Network Disruption

The network programmer attempts to use a special “privileged” SRTP server connection at the Ethernet interface to establish and maintain a connection even under heavy load due to EGD and other SRTP connections. The Ethernet interface supports only one such privileged connection. When the maximum number of privileged connections is in use, no other privileged connections are permitted until a current privileged connection is terminated. This normally occurs when the network programmer disconnects from the target PLC.

When the programmer-PLC network connection is abruptly broken (not the orderly termination performed during disconnection), the SRTP server connection and its underlying TCP connection remain alive until either an SRTP inactivity timeout (see 1.3.9, SRTP Inactivity Timeout details) occurs (20 –30 seconds), or the TCP connection eventually times out (about 7 minutes). If the maximum privileged connections are in use and the programmer reconnects during this interval, it obtains a new, non-privileged connection. Under heavy load at the Ethernet interface, the programmer may experience sluggish response over this non-privileged connection. If this occurs, you can manually disconnect and reconnect the programmer after the previous connection has timed out. Upon reconnection, the programmer should once again obtain a privileged connection.

## 12.10.9 EGD Command Session Conflicts

EGD Commands support only one pending EGD command from a client device to each server device. Attempts to issue a second EGD command from a client to the same server before completion of the first command will result in an error. Some examples are:

1. The logic application issues a second EGD Command COMMREQ to the same remote server, perhaps from a different location in the logic program.
2. The EGDCMD Station Manager command issues a command to the same remote server device as the logic application.

## 12.10.10 SRTP Request Incompatibility with Existing Host Communications Toolkit Devices or Other SRTP Clients

The Advanced User Parameter (AUP) named “chct\_comp” provides greater compatibility with existing Host Communication Toolkit devices. Some Host Communication Toolkit devices generate incorrectly-encoded SRTP messages. In some cases, PACSystems Ethernet interfaces detect and report SRTP encoding errors that were ignored by previous Series 90 products; these errors cause the PACSystems SRTP server to drop the SRTP connection to the Host Communications Toolkit device. If possible, the Host Communications Toolkit device should be upgraded so that it will generate properly-encoded SRTP messages. If the device cannot be upgraded, the “chct\_comp” AUP parameter can be used to tell the PACSystems Ethernet interface to ignore known SRTP errors that were not detected by previous Series 90 products. (See for details of the “chct\_comp” parameter.)

## 12.10.11 COMMREQ Flooding Can Interrupt Normal Operation

The PLC logic application program should generally wait for a response from each COMMREQ function block before activating another COMMREQ function block to the same endpoint. Extremely heavy COMMREQ delivery loading, such as activating the same COMMREQ every logic sweep, can prevent normal SRTP, Modbus, EGD, and Station Manager operation. During such loading, the Ethernet LAN LED may be frozen. Under extreme COMMREQ loading, the Ethernet interface may automatically restart.

## 12.10.12 Accelerated EGD Consumption Can Interfere with EGD Production

Consumed EGD exchanges received from the network normally receive accelerated processing for increased overall EGD performance. This accelerated processing can preempt EGD production activity, possibly delaying the transmission of produced exchanges to the network. Such delay varies with network loading and the volume of consumed exchanges. In applications requiring minimal produced exchange timing variability, the consumed exchange acceleration may be disabled via the “gc\_accel” AUP parameter. (See 13.3.3Appendix A: for details of the “gc\_accel” parameter.) Under extreme network load, accelerated processing of the incoming EGD samples may consume so much processing time that the watchdog timer for the network interface expires and the network interface is reset.

### 12.10.13 Channels Operation Depends Upon PLC Input Scanning

Communication channel operation always includes updating the Channel Status Bits (located within the Ethernet Status data) into PLC memory, which occurs when the PLC scans inputs from the Ethernet module. At least one PLC input scan must occur for each data transfer on a channel, so the channel can run no faster than the PLC scans the Ethernet Status data. When the Ethernet interface is configured to use an I/O Scan Set that runs more slowly than the PLC sweep, each channel must wait until the next time that its scan set runs to transfer its Channel Status bits. This can reduce channel performance.

If the Ethernet interface is configured to use an inactive I/O Scan Set, the Channels Status bits will not be transferred and channel operations will not complete.



## Section 13 Network Administration

This chapter discusses how devices are identified on the network and how data is routed among devices. The main topics covered are:

- IP Addressing
- Gateways
- Subnets and Supernets

## 13.1 IP Addressing

Each TCP/IP node on a network must have a unique IP address. The TCP/IP Ethernet interface is such a node, as is a PC running TCP/IP. There may be other nodes on the network that are not involved with communications to the PLCs, but no matter what their function, each TCP/IP node must have its own IP address. It is the IP address that identifies each node on the IP network (or system of connected networks). The term “host” is often used to identify a node on a network.

### 13.1.1 IP Address Format for Network Classes A, B, C

The IP address is 32 bits long and has a *netid* part and a *hostid* part. Each network is a Class A, Class B, or Class C network. The class of a network determines how an IP address is formatted and is based on the number of bits in the netid part of the IP address.

**Figure 110: IP Address Format for Network Classes A, B, C**

	0	1	8	16	24	31
<b>Class A</b>	0	netid		hostid		
<b>Class B</b>	1	0	netid		hostid	
<b>Class C</b>	1	1	0	netid		hostid

In general, the netid part is assigned by the Internet authorities and the hostid part is assigned by your local network administrator. The class of the network determines the number of hosts that can be supported. A Class A network can support  $2^{24}-2$  (16,777,214) hosts, Class B,  $2^{16}-2$  (65,534) hosts, and Class C,  $2^8-2$  (254) hosts. The minus 2 refers to host numbers reserved for the network itself and the local broadcast.

Each node on the same physical network must have an IP address of the same class and must have the same netid. Each node on the same physical network must have a different hostid thus giving it a unique IP address.

IP addresses are written in “dotted-decimal” format as four decimal integers (0-255) separated by periods where each integer gives the value of one byte of the IP address. For example, the 32-bit IP address:

00001010 00000000 00000000 00000001

is written as

10.0.0.1

To determine the class of an IP address, examine the first integer in its dotted-decimal IP address and compare it with the range of values in the following table.



Range of the first integer	Class
0 – 126	A
127	Loopback
128 - 191	B
192 - 223	C
224 - 239	D (Reserved for Multicast Use)
240 - 255	E (Reserved for Experimental Use)

## 13.1.2 IP Addresses Reserved for Private Networks

RFC 1918 reserves IP addresses in the following ranges to be used for private networks.

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

## 13.1.3 Multicast IP Addresses

Multicast IP Addresses are used in multicasting, a technique that allows delivery of a single packet of data to multiple nodes on the network. Any node that joins a Multicast group will respond to the Multicast IP address assigned to that group. Subsequently, any data sent to that Multicast IP address may be received by all nodes that are members of that multicast group. Multicast (Class D) IP addresses (224.0.0.0 through 239.255.255.255) are reserved by the Internet authorities for multicasting.

Multicasting is a feature of Ethernet Global Data. For more information on the use of multicasting in Ethernet Global Data, see Section 5.

## 13.1.4 Loopback IP Addresses

Class A IP Addresses in the 127.xxx.xxx.xxx range are reserved for loopback addressing. A network packet using a loopback destination address is not transmitted on the network but instead is processed by the same device as if it were received from the network.

PACSystems Ethernet interfaces recognize only the IP address 127.0.0.1 as a loopback address. All other addresses in the range 127.0.0.2 – 127.255.255.255 are ignored and do not provide loopback operation.

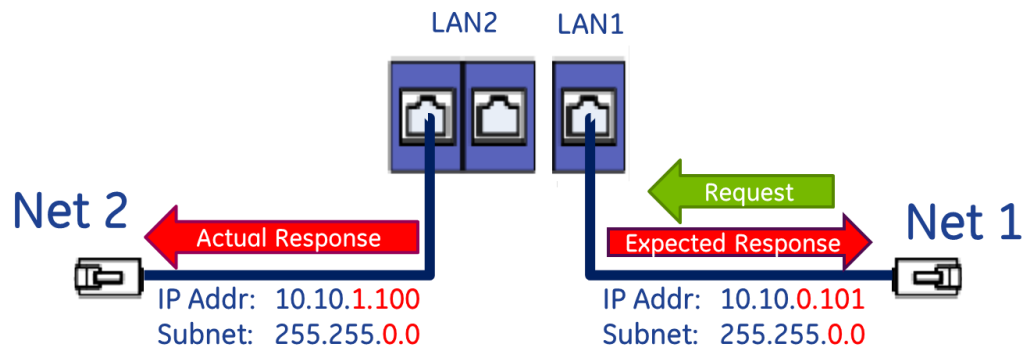
## 13.1.5 Overlapping Subnets

Each interface on a LAN must have a unique IP Address *and* a non-overlapping IP subnet. This is configured in PME. Care must be taken to survey the entire connected network architecture to tabulate the IP addresses and IP subnets already in use, both on the local networks and any of its routed subnets connected with a gateway. Never assign a conflicting IP Address or configure duplicate IP subnets.

The following examples would be problematic:

Problem example #1:

Figure 111: CPE330 Overlapping Local IP Subnet Example



The issue demonstrated in Figure 14 is that requests entering one CPE330 interface can be routed out the other interface since both CPE330 Ethernet ports have been configured to be on the same network (255.255.0.0) but are physically connected to separate networks. Avoid this by assigning non-overlapping Subnets.

**Problem example #2:**

A user wishes to communicate through a routed network to an RX3i CPU with multiple network interfaces (CPE330, in this example). This remote IP device is configured with the following IP parameters:

	LAN1
IP	192.168.0.5
Subnet Mask	255.255.255.0
Gateway	192.168.0.250

LAN1 and LAN2 on the CPE330 are initially configured with the following problematic IP parameters:

	LAN1	LAN2
IP	10.10.0.1	192.168.0.1
Subnet Mask	255.255.255.0	255.255.255.0
Gateway	10.10.0.249	0.0.0.0

The user intends to communicate between the remote device and CPE330 LAN1 (Figure 117). IP Address routing allows the CPE330 to receive the remote IP requests through the respective gateways (192.168.0.250 for the remote node and 10.10.0.249 for CPE330 LAN1). However, since CPE330 LAN2 shares the same IP subnet as the remote network (192.168.0.x), responses may be routed to the local 192.168.0.x network rather than to the remote network (

Figure 113).

The duplicate IP subnet in the example must be eliminated. One way to do this is simply to change the IP Address assigned to CPE330 LAN2 from 192.168.0.1 to 192.168.1.1 thereby creating a non-overlapping 192.168.1.x network. In short, consider the totality of the network when assigning IP subnets and IP Addresses.

**Figure 112: Expected Response Path**

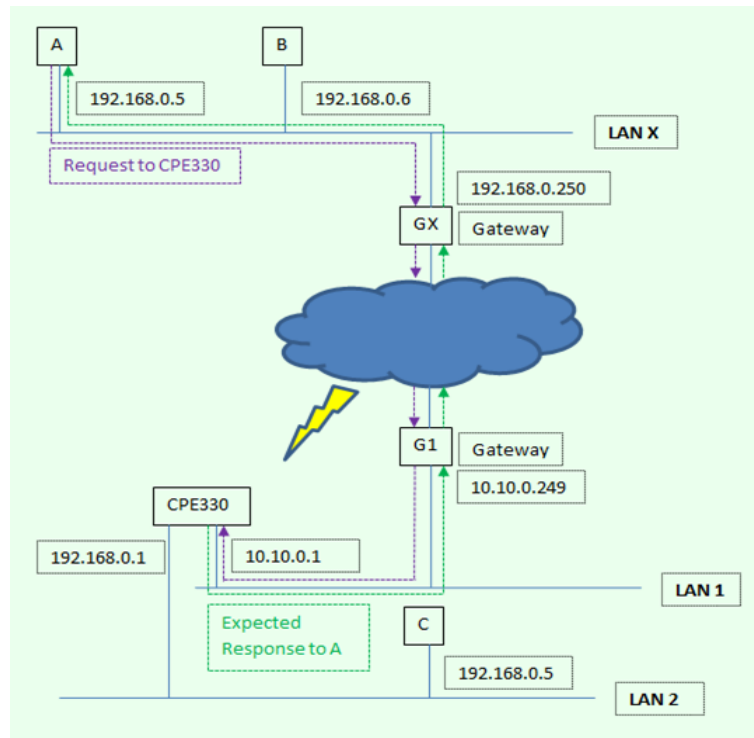
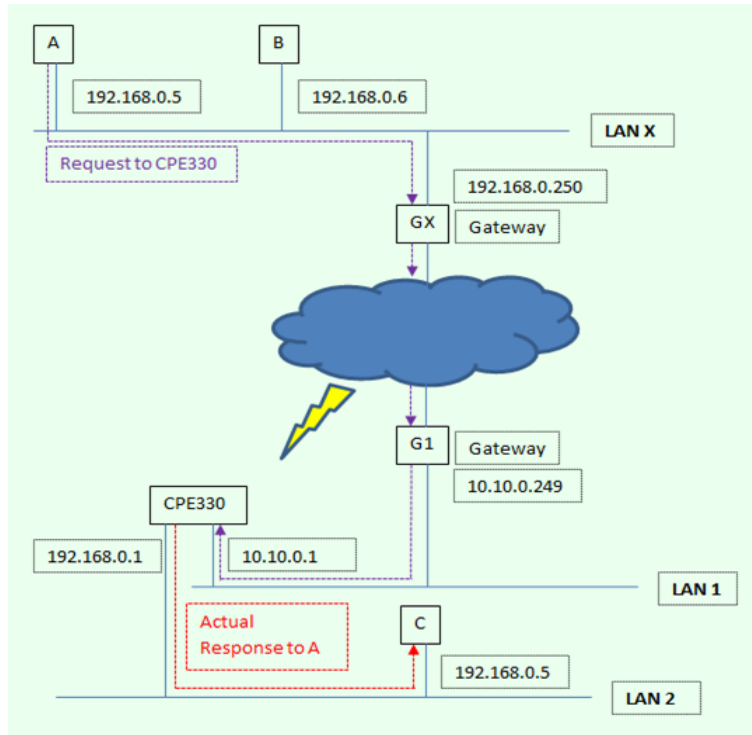


Figure 113: Actual Response Path



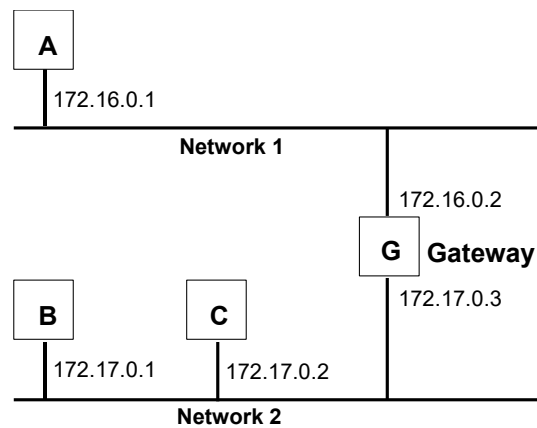
## 13.2 Gateways

Gateways (also known as routers) connect individual physical networks into a system of networks. When a node needs to communicate with a node on another physical network, a gateway transfers the data between the two networks.

### 13.2.1 Networks Connected by a Gateway

The following example shows Gateway G connecting Network 1 with Network 2.

**Figure 114: Gateway Connected to Two Networks**



When host B with IP address 172.17.0.1 communicates with host C, it knows from C's IP address that C is on the same network. In an Ethernet environment, B can then resolve C's IP address to a MAC address (via ARP) and communicate with C directly.

When host B communicates with host A, it knows from A's IP address that A is on another network (the *netids* are different). To send data to A, B must have the IP address of the gateway connecting the two networks. In this example, the gateway's IP address on Network 2 is 172.17.0.3. This address would be configured in the Ethernet interface's module configuration for PLC B as its default gateway address.

Note that the gateway has two IP addresses (172.16.0.2 and 172.17.0.3). The first must be used by hosts on Network 1 and the second must be used by hosts on Network 2. To be usable, a host's gateway must be addressed using an IP address with a *netid* matching its own.

## 13.3 Subnets and Supernetting

Subnets allow a site's network administrators to divide a large network into several smaller networks while still presenting the overall network as one single entity to the outside world. Each of the site's interior gateways needs only maintain the subnet numbers of other interior gateways instead of every single host on the entire network.

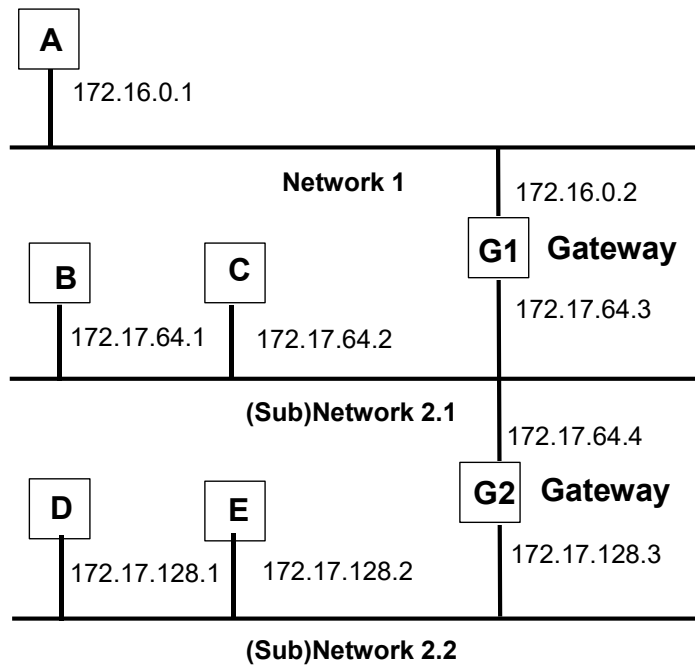
PACSystems Ethernet interfaces support "supernetting," a technique of configuring the subnet mask to allow communication to multiple subnets. The resulting supernet is a block of contiguous subnets addressed as a single subnet.



### 13.3.2 Example: Network Divided into Two Subnets

The new network configuration dividing Network 2 into Subnets 2.1 and 2.2 is shown in Figure 117.

**Figure 117: Network 2 Divided into Subnets 2.1 and 2.2**



Here, a second network with Hosts D and E has been added. Gateway G2 connects Subnet 2.1 with Subnet 2.2. Hosts D and E will use Gateway G2 to communicate with hosts, not on Network 2.2.

Hosts B and C will use Gateways G1 and G2 to communicate with hosts, not on Network 2.1. When B is communicating with D, G2 (the configured Gateway for B) will route the data from B to D through Gateway G2.

Host A will use Gateway G1 to communicate with hosts, not on Network 1.

### 13.3.3 Example: Two Networks Combined into a Supernet

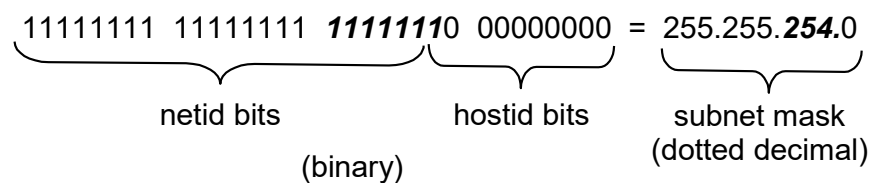
*Supernetting* is a technique used to combine two smaller networks into a larger network by extending the host portion of the subnet mask and reducing the network portion. Supernetting works only with adjacent networks that share the same network id value, such as networks 1 and 2 in this example (

Figure 119).

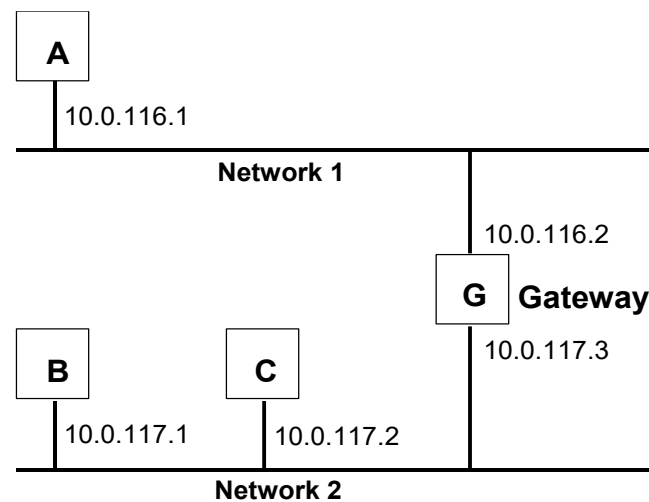
As with subnets, the *subnet mask* is used to divide the IP address into a custom netid portion and hostid portion.

For example, the two networks 10.0.117.0 and 10.0.116.0 can be combined into a larger 10.0.116.0 network if the subnet mask (Figure 118) 255.255.254.0 is applied to both addresses.

**Figure 118: Subnet Mask Used to Affect a Supernet**



**Figure 119: Resulting Supernet**







# Appendix A: Configuring Advanced User Parameters

Advanced User Parameters (AUPs) are internal operating parameters used by an Ethernet interface. For most applications, the default AUPs should not be changed. If it is necessary to change the AUPs, they can be configured within the *Hardware Configuration* tool available in PME.

Alternatively, some Ethernet interfaces can modify these parameters through a user-created AUP file that can be imported into PME and assigned to one or more Ethernet interfaces. These files can be created using any ASCII text editor so long as the file contains the names and values of only those parameters being changed.

This appendix will address all AUPs that can be configured in the AUP file and imported into the Programmer. Most AUPs in the following section can be accessed and configured in the Ethernet settings of PME.

## A.1 Advanced User Parameter Definitions

System Memory Parameters (task b)		Default	Range
staudp <sup>28</sup>	Remote command UDP port	18245 (4745H)	1 – 65535 (ffffH) Only the gdata_port and gXX_udp parameters may share the same UDP port number. All other UDP port number parameters in the AUP file must use unique port numbers.

Backplane Driver Parameters (task c)		Default	Range
crsp_tout	CPU response timeout. Amount of time to wait for the CPU to respond to a request sent through the PLC Driver.	60 seconds	10 – 3600 (E10H)
chct_comp	HCT compatibility option. Allows Ethernet interface to ignore SRTP header errors (typically generated by remote HCT devices) that were not detected in previous Series 90 products. 0 = HCT compatibility disabled (= report all errors) 1 = HCT compatibility enabled (= ignore some errors)	0 (0H)	0, 1
cstorm	COMMREQ storm onset threshold. Establishes a number of COMMREQs per second at or above which the PLC application is considered to be sending COMMREQs so rapidly that the Ethernet interface cannot continue normal operation. Setting this parameter to 0 disables COMMREQ storm error detection.	500 (01F4H)	0 – 10,000 (2710H)
cnostorm	COMMREQ storm end threshold. Establishes the number of COMMREQs per second at or below which the COMMREQ storm condition (see above) is considered to have ended. If the <i>cstorm</i> parameter is not set to 0, this parameter should always be less than <i>cstorm</i> . If <i>cstorm</i> is set to 0, this parameter is ignored.	100 (0064H)	0 – 10,000 (2710H)

<sup>28</sup> Supported by RX3i CPE302, CPE305 and CPE310 models.

<i>RDS Parameters (task d)</i>		Default	Range
		None	None

<b>ARP Parameters (task f)</b>		Default	Range
fflush	Interval in seconds at which to flush the ARP cache	600 (10 minutes)	0 – 604800 (93A80H)

<b>Ethernet Global Data Parameters (task g)</b>		Default	Range
gctl_port	UDP port for EGD control messages	7937 (1f01H)	1 – 65535 (ffffH) Only the gdata_port and gXX_udp parameters may share the same UDP port number. All other UDP port number parameters in the AUP file must use unique port numbers.
gdata_port	UDP port for point-to-point (unicast) EGD messages	18246 (4746H)	1 – 65535 (ffffH) Only the gdata_port and gXX_udp parameters may share the same UDP port number. All other UDP port number parameters in the AUP file must use unique port numbers.
gbcast_ttl	IP time-to-live for global broadcast messages (hop count)	1 (1H)	0 – 255 (00ffH)
gucast_ttl	IP time-to-live for point-to-point (unicast) messages (hop count)	16 (10H)	0 – 255 (00ffH)
gp_phase	Startup delay time in ms for successive produced exchanges	0 (0H)	0 – 65535 (ffffH)
gcmd_pri	EGD command processing priority relative to data production. 0 = EGD commands have lower priority. 1 = EGD commands have equal priority. 2 = EGD commands have higher priority.	0 (0H)	0, 1, 2

gc_accel	Enable consumed exchange acceleration. 0= Acceleration disabled; 1= Acceleration enabled.	1 (1H)	0, 1
gnostale	When bit zero in the <i>Production Status</i> field of the PDU of a consumed sample is set, sample is stale. 0 = allow status to be sent to the application when exchange status indicates stale data. 1 = prevent the new status from being sent to the application if exchange status indicates stale data.	0 (0H)	0, 1

EGD provides a UDP port parameter and host group IP address parameter for each of 32 possible host groups (1-32). The parameter formats for each host group are shown below. XX specifies host group 1-32.

gXX_udp	UDP port for host group XX	18246 (4746H)	1 – 65535 (ffffH) Only the gdata_port and gXX_udp parameters may share the same UDP port number. All other UDP port number parameters in the AUP file must use unique port numbers.
gXX_addr	Multicast host group IP Address (must be Class D address)	224.0.7.XX	224.0.0.2 – 239.255.255.255
gXX_ttl	IP time-to-live for host group (multicast) messages (hop count)	1 (1H)	0 – 255 (00ffH)

**Note:** If you configure different TTL values for EGD exchanges with Unicast and Broadcast destination types on IC695ETM001-Jx, the largest value will be used for all Unicast and Broadcast exchanges.

Other models have broadcast fixed to a TTL of 1.

If you configure multiple gXX\_ttl values for different Multicast exchanges, the smallest value among the configured parameters will be used for all exchanges.

This applies only to PACS Ethernet interface modules.

SRTP Client (Channels) Parameters (task h)		Default	Range
hconn_tout	TCP Connect timeout (in milliseconds)	75000 (124F8H)	10 – 75000 (124F8H)

IP Parameters (task i)		Default	Range
ttl	IP header default time-to-live (hop count)	64 (0040H)	0 – 255 (00ffH)
ifrag_tmr	IP fragment timeout interval in seconds	3 (0003H)	0 – 65535 (ffffH)

ICMP/IGMP Parameters (task j)		Default	Range
		None	None

Network Interface Parameters (task l)		Default	Range
lduplex0	Ethernet duplex for Controller 0=auto-detect, 1 = half, 2= full)	0	0,1,2
lduplex1a	Ethernet duplex for Port 1A (0=auto-detect, 1=half, 2=full)	0	0,1,2
lduplex1b	Ethernet duplex for Port 1B (0=auto-detect, 1=half, 2=full)	0	0,1,2
lspeed0	Ethernet speed for Controller (0=auto-detect, 1=10Mbit, 2=100Mbit)	0	0,1,2
lspeed1a	Ethernet speed for Port 1A (0=auto-detect, 1=10Mbit, 2=100Mbit)	0	0,1,2
lspeed1b	Ethernet speed for Port 1B (0=auto-detect, 1=10Mbit, 2=100Mbit)	0	0,1,2

Modbus TCP/IP Server Parameters (task m)		Default	Range
		None	None

SNTP Time Transfer to CPU Parameters (task n)		Default	Range
ncpu_sync	Configures this Ethernet interface to support CPU TOD clock synchronization with network timeserver. (0=Not supported; 1=Supported)	0	0, 1

Unicast SNTP AUP Parameters (task n)		Default	Range
nmode	SNTP Mode of operation 0 = Multicast and Broadcast mode 1 = Unicast mode This parameter is required when unicast mode is used.	0	0-1.
nprimary	IP address of the primary time server in dotted decimal format (xxx.xxx.xxx.xxx). This parameter is required when unicast mode is used.	None	Any valid unicast IPv4 address
nsecondary	IP address of the secondary time server in dotted decimal format (xxx.xxx.xxx.xxx). This parameter is optional.	None	Any valid unicast IPv4 address
npollinterval	Poll interval of Unicast Period, in seconds, at which new time requests are sent to the server. The specified period will be rounded to the nearest power of 2. This parameter is optional.	32	16 - 1,024
npollcount	Number of retransmissions that will be sent when no timely response is received from the server. This parameter is optional.	3	1 – 100
npolltimeout	The time, in seconds, that the module will wait for a response from the server. This parameter is optional.	2	1 – 100.

SNTP Local Time Corrections (LTC) and Daylight Savings Time (DST) Parameters (task n)		Default	Range
nltc_offset	This signed value indicates the hours and minutes of the offset of local time from UTC. The minutes must be specified by one of four values, 0, 15, 30, or 45.	0:00	-12:45 to +14:45
ndst_offset	The offset between DST and standard time in hours and minutes, where the minutes are limited to the values 0, 15, 30, and 45.	None	0:15 to 1:00
ndst_start_month	The month when DST begins.	None	1 – 12
ndst_start_day	The day of the week when DST begins. 1 = Sunday 7 = Saturday	None	1 – 7
ndst_start_week	The number of the occurrence of ndst_start_day in the month. (1 is the first occurrence.)	None	1 – 4
ndst_start_time	The time, in hours and minutes, when DST begins.	None	0:00 – 23:59
ndst_ref_zone	Indicates the time zone of reference for ndst_start_time and ndst_end_time. L = Local Time U = UTC	None	L or U
ndst_end_month	The month when DST ends. Note that in the southern hemisphere, this value will be smaller than the start value.	None	1 – 12
ndst_end_day	The day of the week when DST ends. 1 = Sunday 7 = Saturday	None	1 – 7
ndst_end_week	The number of the occurrence of ndst_end_day in the month. (1 is the first occurrence.)	None	1 – 4
ndst_end_time	The time, in hours and minutes, when DST ends.	None	0:00 – 23:59

Modbus TCP/IP Client Parameters (task o)	Default	Range
	None	None



Ethernet Redundancy Parameters (task q)		Default	Range
rdipckival	Interval between additional checks for Redundant IP address in use (in milliseconds).  When activating the Redundant IP address, the ETM sends a burst of three ARP requests at 20 ms intervals.  If the ETM receives an SRP response, it delays for the interval specified by <i>rdipckival</i> , plus an additional 20ms. After the specified interval has passed, the ETM tries again, repeating the cycle of three ARP requests. The ETM repeats the request cycle after each SRP response; however, the delay interval after a response is received doubles each cycle, to a maximum of 2.0 seconds.	100 (0064H)	1 – 1000ms
rdiparpivl	Interval between gratuitous ARP requests sent by the backup unit on behalf of the new active unit (in ms).	100 (0064H)	1 – 1000ms
rdipnumarp	Number of gratuitous ARP requests to send out during Redundant IP activation process.	1 (0001H)	1 – 25
rdiparplog	Number of gratuitous ARP requests to send by backup unit before a “Redundant IP not available” exception is logged. (The backup unit continues to send ARP requests as long as it receives network packets addressed to the Redundant IP Address.)	5 (0005H)	1 – 25

FTP Parameters (task t)		Default	Range
tpassword	Password for login for FTP access.	“system”	0 to 8 characters

UDP Parameters (task u)		Default	Range
		None	None

SRTP Parameters (task v)		Default	Range
vconn_tout	SRTP inactivity timeout (in seconds). Amount of time to wait before cleaning up an abandoned privileged SRTP server connection. Any non-zero value is rounded up to the next multiple of 5 seconds. See 1.3.9, SRTP Inactivity Timeout, for details.  All privileged connections initially use the SRTP inactivity timeout specified by this AUP parameter. Inactivity timeouts established by an SRTP Client on an individual connection will override any AUP specified inactivity timeout on that connection.  0 = SRTP Inactivity Timeout disabled.	30 seconds	0 – 420 seconds

TCP Parameters (task w)		Default	Range
wndelay	TCP nodelay option (0= inactive; 1 = active)	1 (1H)	0, 1
wkal_idle	TCP keepalive timer value (in seconds)	240 (4.0 min)	1 – 65535 (ffffH)
wkal_cnt	TCP keepalive extra probe count (in addition to single probe always performed)	2	0 – 65535 (ffffH)
wkal_intvl	TCP keepalive probe interval (in seconds)	60 seconds	1 – 65535 (ffffH)
wsnd_buf	TCP send buffer size (in bytes)	65535 (ffffH)	0 – 65535 (ffffH)
wrcv_buf	TCP receive buffer size (in bytes)	4096 (1000H)	0 – 32767 (7fffH)

## A.2 AUP Support by Ethernet Interface

The following tables illustrate which parameters are supported by each Ethernet interface and by what method.

**AUP File** describes a parameter that can only be updated by storing an AUP file in PME.

**HWC** describes a parameter that can only be updated changing AUP values in a Hardware Configuration in PME.

**Default Value Only** describes a parameter that is static and cannot be changed.

**Ignored** describes a static parameter that cannot be changed and will ignore parameter values in an imported AUP File.

**Not Supported** describes a parameter that is not supported by this Ethernet interface.

System Memory Parameters (task b)	CPE302/305/310	CPE110/115/330/400 and CPL410	ETM001-jx	ETM001-Kxxx
staudp	AUP File	Default Value Only	AUP File	Ignored
stpasswd	AUP File	Default Value Only	AUP File	Ignored
<b>Backplane Driver Parameters (task c)</b>				
Backplane Driver Parameters (task c)	CPE302/305/310	CPE330/400 and CPL410	ETM001-jx	ETM001-Kxxx
crsp_tout	AUP File	Default Value Only	AUP File	AUP File
chct_comp	AUP File	Default Value Only	AUP File	AUP File
cstorm	AUP File	Default Value Only	AUP File	AUP File
cnostorm	AUP File	Default Value Only	AUP File	AUP File
<b>RDS Parameters (task d)</b>				
RDS Parameters (task d)	CPE302/305/310	CPE330/400 and CPL410	ETM001-jx	ETM001-Kxxx
Not Available	Not Available	Not Available	Not Available	Not Available
<b>ARP Parameters (task f)</b>				
ARP Parameters (task f)	CPE302/305/310	CPE330/400 and CPL410	ETM001-jx	ETM001-Kxxx
fflush	AUP File	Not Supported	AUP File	AUP File
fstale	Not Supported	Not Supported	Not Supported	Not Supported
<b>EGD Parameters (task g)</b>				
EGD Parameters (task g)	CPE302/305/310	CPE330/400 and CPL410	ETM001-jx	ETM001-Kxxx
gctl_port	AUP File	Default Value Only	AUP File	Ignored
gdata_port	AUP File	Default Value Only	AUP File	Ignored
gbcast_ttl	AUP File	Default Value Only	AUP File	Ignored

gucast_ttl	AUP File	HWC	AUP File	Yes
gmcast_ttl	Not Supported	HWC	Not Supported	AUP File or HWC
gmcast_ttl2	Not Supported	HWC	Not Supported	Not Supported
gp_phase	AUP File	HWC	AUP File	AUP File or HWC
gcmd_pri	AUP File	Default Value Only	AUP File	Ignored
gc_accel	AUP File	Default Value Only	AUP File	Ignored
gnostale	AUP File	HWC	AUP File	AUP File or HWC
gXX_udp	AUP File	Default Value Only	AUP File	Ignored
gXX_addr	AUP File	HWC	AUP File	Yes
gXX_udp2	Not Supported	Default Value Only	Not Supported	Not Supported
gXX_addr2	Not Supported	HWC	Not Supported	Not Supported
gXX_ttl	AUP File	Not Supported	AUP File	AUP File
<b>SRTP Client (Channels) Parameters (task h)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
hconn_tout	AUP File	Default Value Only	AUP File	AUP File
<b>IP Parameters (task i)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
ittl	AUP File	Default Value Only	AUP File	AUP File
ifrag_tmr	AUP File	Default Value Only	AUP File	AUP File
<b>ICMP/IGMP Parameters (task j)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
Not Available	Not Available	Not Available	Not Available	Not Available
<b>Network Interface Parameters (task l)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
lduplex0	AUP File	Not Supported	AUP File	Ignored
lduplex1a	AUP File	Default Value Only	AUP File	Ignored
lduplex1b	AUP File	Not Supported	AUP File	Ignored

Ispeed0	AUP File	Not Supported	AUP File	Ignored
Ispeed1a	AUP File	Default Value Only	AUP File	HWC
Ispeed1b	AUP File	Not Supported	AUP File	HWC
<b>Modbus TCP/IP Server Parameters (task m)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
osnd_buf	Not Supported	Default Value Only	Not Supported	Not Supported
orcv_buf	Not Supported	Default Value Only	Not Supported	Not Supported
<b>SNTP Time Transfer to CPU Parameters (task n)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
ncpu_sync	AUP File	Default Value Only	AUP File	AUP File
<b>Unicast SNTP AUP Parameters (task n)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
nmode	AUP File	HWC	AUP File	AUP File or HWC
nprimary	AUP File	Default Value Only	AUP File	AUP File
nsecondary	AUP File	Default Value Only	AUP File	AUP File
npollinterval	AUP File	Default Value Only	AUP File	AUP File
npollcount	AUP File	HWC	AUP File	AUP File or HWC
npolltimeout	AUP File	HWC	AUP File	AUP File or HWC
<b>SNTP Local Time Corrections (LTC) and Daylight Savings Time (DST) Parameters (task n)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
nltc_offset	AUP File	HWC	AUP File	AUP File or HWC
ndst_offset	AUP File	HWC	AUP File	AUP File or HWC
ndst_start_month	AUP File	HWC	AUP File	AUP File or HWC

ndst_start_ day	AUP File	HWC	AUP File	AUP File or HWC
ndst_start_ week	AUP File	HWC	AUP File	AUP File or HWC
ndst_start_ time	AUP File	HWC	AUP File	AUP File or HWC
ndst_ref_ zone	AUP File	HWC	AUP File	AUP File or HWC
ndst_end_ month	AUP File	HWC	AUP File	AUP File or HWC
ndst_end_ day	AUP File	HWC	AUP File	AUP File or HWC
ndst_end_ week	AUP File	HWC	AUP File	AUP File or HWC
ndst_end_ time	AUP File	HWC	AUP File	AUP File or HWC
<b>Modbus TCP/IP Client Parameters (task o)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
Not Available	Not Available	Not Available	Not Available	Not Available
<b>Ethernet Redundancy Parameters (task q)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
rdipckival	AUP File	Default Value Only	AUP File	AUP File
rdiparpivl	AUP File	Default Value Only	AUP File	AUP File
rdipnumarp	AUP File	Default Value Only	AUP File	AUP File
rdiparplog	AUP File	Default Value Only	AUP File	AUP File
<b>FTP Parameters (task t)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
tpassword	AUP File	Default Value Only	AUP File	Ignored
<b>UDP Parameters (task u)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
Not Available	Not Available	Not Available	Not Available	Not Available
<b>SRTP Parameters (task v)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
vconn_tout	AUP File	Default Value Only	AUP File	AUP File

TCP Parameters (task w)	CPE302/305/310	CPE330/400 and CPL410	ETM001-jx	ETM001-Kxxx
wndelay	AUP File	Default Value Only	AUP File	AUP File
wkal_idle	AUP File	Default Value Only	AUP File	AUP File
wkal_cnt	AUP File	Default Value Only	AUP File	AUP File
wkal_intvl	AUP File	Default Value Only	AUP File	AUP File
wsnd_buf	AUP File	Default Value Only	AUP File	Ignored
wrcv_buf	AUP File	Default Value Only	AUP File	Ignored
<b>Application Driven Data Exchange (ADDE) (task ?)</b>	<b>CPE302/305/310</b>	<b>CPE330/400 and CPL410</b>	<b>ETM001-jx</b>	<b>ETM001-Kxxx</b>
xenable	AUP File	Not Supported	AUP File	Ignored
xdata_port	AUP File	Not Supported	AUP File	Ignored
xucast_ttl	AUP File	Not Supported	AUP File	Ignored
xmcast_ttl	AUP File	Not Supported	AUP File	Ignored
xr_accel	AUP File	Not Supported	AUP File	Ignored
xresp_tout	AUP File	Not Supported	AUP File	Ignored

## A.3 Format of the Advanced User Parameters File

The AUP file can be constructed from an text editor, but must have this format:

*AUP\_r\_s.apf*

where *r* and *s* indicate the Rack and Slot location of the Ethernet interface. (For an embedded Ethernet interface, *r* and *s* indicate the Rack and Slot location of the CPU module.)

<parameter name> = <parameter value>  
<parameter name> = <parameter value>  
<parameter name> = <parameter value>

- Note:** 1. Entering an invalid AUP parameter within the AUP file will force a *Stop Faulted* condition in the CPU. Additionally, two faults should appear in the Controller Fault Table:
- a. “Unable to deliver configuration to module”
  - b. “LAN system-software fault; resuming” the extra fault data should indicate that an AUP file has caused the faulted condition along with a pointer to line with the invalid entry.

The  
AUP file  
has the

following requirements:

- The first line of the file must consist only of the text: *AUP\_r\_s* where *r* and *s* indicate the Rack and Slot location of the Ethernet interface (or, for an embedded Ethernet interface, the location of the CPU module).  
(For example, an Ethernet Module in rack 0, slot 11 would be indicated as *AUP\_0\_11*.  
  
This is intended as a convenient way to differentiate AUP files for different modules. Any rack and slot number will do, so that the same AUP file can be imported for use by multiple Ethernet interfaces if desired.
- All parameter names are lowercase. The equal sign (=) is required between the parameter name and parameter value.
- Spaces are allowed, but not required, between the parameter name and the equal symbol (=) and between the equal symbol and the parameter value.
- Character string values are case-sensitive; as with Station Manager commands, uppercase parameter values must be enclosed within a pair of double quotes.
- Numeric parameters are entered in decimal or hexadecimal format; hexadecimal values must be terminated with an 'h' or 'H' character.
- IP addressing parameters must be entered in standard dotted decimal format.
- Comments in the file must start with a semicolon character. All characters in the same line following a semicolon are ignored.
- Blank lines are ignored.
- The maximum line length in the AUP file is 80 characters. Any line, including comments, that exceeds this length will cause errors in processing.



**Example:**

The following example shows a sample AUP file which specifies the Ethernet speed for the Ethernet ports.

```
A_0_4  
lspeed1a=2      ;      sets Ethernet speed for Port 1A to 100 Mbps  
lspeed1b=2 ;      sets Ethernet speed for Port 1B to 100 Mbps
```

# Support Links

Home link: <http://www.Emerson.com/Industrial-Automation-Controls>

Knowledge Base: <https://www.Emerson.com/Industrial-Automation-Controls/Support>

## Customer Support and Contact Information

### Americas

Phone: 1-888-565-4155  
1-434-214-8532 (If toll free option is unavailable)

Customer Care (Quotes/Orders>Returns): [customercare.mas@emerson.com](mailto:customercare.mas@emerson.com)  
Technical Support: [support.mas@emerson.com](mailto:support.mas@emerson.com)

### Europe

Phone: +800-4444-8001  
+420-225-379-328 (If toll free option is unavailable)

Customer Care (Quotes/Orders>Returns): [customercare.emea.mas@emerson.com](mailto:customercare.emea.mas@emerson.com)  
Technical Support: [support.mas.emea@emerson.com](mailto:support.mas.emea@emerson.com)

### Asia

Phone: +86-400-842-8599  
+65-6955-9413 (All other countries)

Customer Care (Quotes/Orders>Returns): [customercare.cn.mas@emerson.com](mailto:customercare.cn.mas@emerson.com)  
Technical Support: [support.mas.apac@emerson.com](mailto:support.mas.apac@emerson.com)

Any escalation request should be sent to: [mas.sfdcescalation@emerson.com](mailto:mas.sfdcescalation@emerson.com)

**Note:** If the product is purchased through an Authorized Channel Partner, please contact the seller directly for any support.

Emerson reserves the right to modify or improve the designs or specifications of the products mentioned in this manual at any time without notice. Emerson does not assume responsibility for the selection, use or maintenance of any product. Responsibility for proper selection, use and maintenance of any Emerson product remains solely with the purchaser.

© 2021 Emerson. All rights reserved.

Emerson Terms and Conditions of Sale are available upon request. The Emerson logo is a trademark and service mark of Emerson Electric Co. All other marks are the property of their respective owners.

