

GE  
Intelligent Platforms

Programmable Control Products

# PACSystems\* RX7i & RX3i TCP/IP Ethernet Communications User Manual

GFK-2224M  
October 2014



## Warnings, Cautions, and 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 GE 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 GE 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 GE products referenced herein.

GE 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.

GE 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.

- \* Indicates a trademark of General Electric Company and/or its subsidiaries. All other trademarks are the property of their respective owners.

## Contact Information

---

If you purchased this product through an Authorized Channel Partner, please contact the seller directly.

### **General Contact Information**

Online technical support and GlobalCare	<a href="http://support.ge-ip.com">http://support.ge-ip.com</a>
Additional information	<a href="http://www.ge-ip.com/">http://www.ge-ip.com/</a>
Solution Provider	<a href="mailto:solutionprovider.ip@ge.com">solutionprovider.ip@ge.com</a>

### **Technical Support**

If you have technical problems that cannot be resolved with the information in this manual, please contact us by telephone or email, or on the web at <http://support.ge-ip.com>

### **Americas**

Online Technical Support	<a href="http://support.ge-ip.com">http://support.ge-ip.com</a>
Phone	1-800-433-2682
International Americas Direct Dial	1-780-420-2010 (if toll free 800 option is unavailable)
Technical Support Email	<a href="mailto:support.ip@ge.com">support.ip@ge.com</a>
Customer Care Email	<a href="mailto:customercare.ip@ge.com">customercare.ip@ge.com</a>
Primary language of support	English

### **Europe, the Middle East, and Africa**

Online Technical Support	<a href="http://support.ge-ip.com">http://support.ge-ip.com</a>
Phone	+800-1-433-2682
EMEA Direct Dial	+420 239015850 (if toll free 800 option is unavailable or if dialing from a mobile telephone)
Technical Support Email	<a href="mailto:support.emea.ip@ge.com">support.emea.ip@ge.com</a>
Customer Care Email	<a href="mailto:customercare.emea.ip@ge.com">customercare.emea.ip@ge.com</a>
Primary languages of support	English, French, German, Italian, Czech, Spanish

### **Asia Pacific**

Online Technical Support	<a href="http://support.ge-ip.com">http://support.ge-ip.com</a>
Phone	+86-400-820-8208 +86-21-3877-7006 (India, Indonesia, and Pakistan)
Technical Support Email	<a href="mailto:support.cn.ip@ge.com">support.cn.ip@ge.com</a> (China) <a href="mailto:support.jp.ip@ge.com">support.jp.ip@ge.com</a> (Japan) <a href="mailto:support.in.ip@ge.com">support.in.ip@ge.com</a> (remaining Asia customers)
Customer Care Email	<a href="mailto:customercare.apo.ip@ge.com">customercare.apo.ip@ge.com</a> <a href="mailto:customercare.cn.ip@ge.com">customercare.cn.ip@ge.com</a> (China)



# Table of Contents

---

Contact Information .....	ii
Table of Contents .....	v
Table of Figures.....	xiii
<b>Chapter 1</b> <b>Introduction .....</b>	<b>1</b>
<b>1.1</b> <b>Revisions in this Manual .....</b>	<b>2</b>
<b>1.2</b> <b>Other PACSystems Manuals.....</b>	<b>2</b>
<b>1.3</b> <b>Ethernet Interfaces for PACSystems Controllers.....</b>	<b>3</b>
1.3.1      Rack-based and RX7i Embedded Interfaces - Features .....	3
1.3.2      RX3i CPE305/CPE310 Embedded Ethernet Interface - Features .....	4
1.3.3      Ethernet Interface Specifications .....	5
1.3.4      Ethernet Interface Ports .....	5
1.3.5      Station Manager.....	6
1.3.6      Firmware Upgrades.....	6
1.3.7      Built-In Web Server .....	6
1.3.8      SRTP Client (Channels).....	6
1.3.9      Modbus TCP Client (Channels).....	6
1.3.10     Ethernet Global Data (EGD) <sup>1</sup> .....	7
1.3.11     SRTP Inactivity Timeout .....	7
<b>1.4</b> <b>Ethernet Redundancy Operation .....</b>	<b>8</b>
1.4.1      HSB CPU Redundancy .....	8
1.4.2      Non-HSB Redundancy .....	9
1.4.3      Effect of Redundancy Role Switching on Ethernet Communications.....	9
1.4.4      SRTP Server Operation in a Redundancy System.....	10
1.4.5      SRTP Client Operation in a Redundancy System.....	11
1.4.6      Modbus TCP Server Operation in a Redundancy System.....	11
1.4.7      Modbus TCP Client Operation in a Redundancy System .....	11
1.4.8      EGD Class 1 (Production & Consumption) in a Redundancy System.....	11
1.4.9      EGD Class 2 Commands in a Redundancy System .....	11
1.4.10     Web Server Operation in a Redundancy System.....	12
1.4.11     FTP Operation in a Redundancy System .....	12
1.4.12     SNMP Operation in a Redundancy System .....	12
1.4.13     Remote Station Manager Operation in a Redundancy System.....	12
1.4.14     IP Address Configuration in a Redundancy System .....	12
<b>Chapter 2</b> <b>Installation and Start-up: RX3i Embedded Interface .....</b>	<b>13</b>
<b>2.1</b> <b>RX3i Embedded Ethernet Interface Indicators .....</b>	<b>13</b>
2.1.1      Ethernet Port LEDs Operation .....	13
2.1.2      Module Installation .....	13

2.2	<b>Ethernet Port Connector</b> .....	<b>14</b>
2.2.1	Connection to a 10Base-T / 100Base Tx Network .....	14
2.2.2	10Base-T/100Base Tx Port Pinouts .....	14
2.3	<b>Pinging TCP/IP Ethernet Interfaces on the Network</b> .....	<b>15</b>
2.3.1	Pinging the Ethernet Interface from a UNIX Host or Computer Running TCP/IP Software .....	15
2.3.2	Determining if an IP Address is Already Being Used .....	15
<b>Chapter 3</b>	<b>Installation and Start-up: Rack-based and RX7i Embedded Interface</b> .....	<b>17</b>
3.1	<b>Ethernet Interface Controls and Indicators</b> .....	<b>18</b>
3.1.1	Ethernet LEDs.....	19
3.1.2	Ethernet Restart Pushbutton .....	20
3.2	<b>Module Installation</b> .....	<b>21</b>
3.2.1	Installing an RX7i CPU with Embedded Ethernet Interface .....	21
3.2.2	Installing an RX7i Ethernet Interface Module.....	21
3.2.3	Installing an RX3i Ethernet Interface Module.....	22
3.3	<b>Ethernet Port Connectors</b> .....	<b>23</b>
3.3.1	Embedded Switch.....	23
3.3.2	Connection to a 10Base-T / 100Base Tx Network .....	24
3.4	<b>Station Manager Port</b> .....	<b>26</b>
3.4.1	Port Settings .....	26
3.5	<b>Verifying Proper Power-Up of the Ethernet Interface after Configuration</b> .....	<b>27</b>
3.6	<b>Pinging TCP/IP Ethernet Interfaces on the Network</b> .....	<b>27</b>
3.6.1	Pinging the Ethernet Interface from a UNIX Host or Computer Running TCP/IP Software .....	27
3.6.2	Determining if an IP Address is Already Being Used .....	28
3.7	<b>Ethernet Plug-in Applications</b> .....	<b>28</b>
<b>Chapter 4</b>	<b>Configuration</b> .....	<b>29</b>
4.1	<b>RX3i Embedded Ethernet Interfaces</b> .....	<b>29</b>
4.1.1	Ethernet Configuration Data .....	29
4.1.2	Initial IP Address Assignment .....	30
4.1.3	Configuring the Ethernet Interface Parameters.....	31
4.2	<b>Rack-based and RX7i Embedded Interfaces</b> .....	<b>35</b>
4.2.1	Ethernet Configuration Data .....	36
4.2.2	Initial IP Address Assignment .....	36
4.2.3	Configuring Ethernet Interface Parameters .....	39
4.2.4	Configuring Ethernet Global Data.....	42
<b>Chapter 5</b>	<b>Ethernet Global Data</b> .....	<b>57</b>
5.1	<b>Ethernet Global Data Operation</b> .....	<b>58</b>
5.1.1	EGD Producer.....	58

5.1.2	EGD Consumers.....	58
<b>5.2</b>	<b>EGD Exchanges.....</b>	<b>59</b>
5.2.1	Content of an Ethernet Global Data Exchange.....	59
5.2.2	Data Ranges (Variables) in an Ethernet Global Data Exchange.....	59
5.2.3	Valid Memory Types for Ethernet Global Data.....	60
5.2.4	Planning Exchanges.....	60
5.2.5	Using Ethernet Global Data in a Redundancy System.....	61
<b>5.3</b>	<b>Sending an Ethernet Global Data Exchange to Multiple Consumers.....</b>	<b>61</b>
5.3.1	Multicasting Ethernet Global Data.....	61
5.3.2	Broadcasting Ethernet Global Data.....	62
5.3.3	Changing Group ID in Run Mode.....	62
<b>5.4</b>	<b>Ethernet Global Data Timing.....</b>	<b>63</b>
5.4.1	EGD Synchronization.....	63
5.4.2	Configurable Producer Period for an EGD Exchange.....	64
5.4.3	Consumer Update Timeout Period.....	64
<b>5.5</b>	<b>Time-Stamping of Ethernet Global Data Exchanges.....</b>	<b>65</b>
5.5.1	Obtaining Timestamps from the Ethernet Interface Clock.....	66
5.5.2	Obtaining Timestamps from Embedded Ethernet Interface for RX3i CPE305/CPE310.....	67
5.5.3	Obtaining Timestamps from the CPU TOD Clock.....	67
5.5.4	SNTP Operation.....	75
<b>5.6</b>	<b>Effect of PLC Modes and Actions on EGD Operations.....</b>	<b>76</b>
5.6.1	Run Mode Store of EGD.....	77
<b>5.7</b>	<b>Monitoring Ethernet Global Data Exchange Status.....</b>	<b>81</b>
5.7.1	Exchange Status Word Error Codes.....	82
<b>Chapter 6</b>	<b>Programming EGD Commands.....</b>	<b>83</b>
<b>6.1</b>	<b>General Use of EGD Commands.....</b>	<b>83</b>
<b>6.2</b>	<b>Using EGD Commands in a Redundancy System.....</b>	<b>83</b>
<b>6.3</b>	<b>COMMREQ Format for Programming EGD Commands.....</b>	<b>83</b>
<b>6.4</b>	<b>COMMREQ Status for the EGD Commands.....</b>	<b>84</b>
6.4.1	COMMREQ Status Values.....	84
<b>6.5</b>	<b>Read PLC Memory (4000).....</b>	<b>85</b>
6.5.1	Read PLC Memory Command Block.....	85
<b>6.6</b>	<b>Write PLC Memory (4001).....</b>	<b>88</b>
6.6.1	Write PLC Memory Command Block.....	88
<b>6.7</b>	<b>Read EGD Exchange (4002).....</b>	<b>90</b>
6.7.1	Read EGD Exchange Command Block.....	90

<b>6.8</b>	<b>Write EGD Exchange (4003)</b> .....	<b>93</b>
6.8.1	Write EGD Exchange Command Block.....	93
<b>6.9</b>	<b>Masked Write to EGD Exchange (4004)</b> .....	<b>95</b>
6.9.1	Masked Write EGD Exchange Command Block.....	95
<b>Chapter 7</b>	<b>Programming SRTP Channel Commands</b> .....	<b>99</b>
<b>7.1</b>	<b>SRTP Channel Commands</b> .....	<b>99</b>
7.1.1	Channel Operations.....	100
7.1.2	Aborting and Re-tasking a Channel.....	100
7.1.3	Monitoring the Channel Status.....	100
7.1.4	SRTP Channel Commands in a Redundant System.....	100
7.1.5	Executing a Channel Command .....	101
<b>7.2</b>	<b>COMMREQ Format for Programming Channel Commands</b> .....	<b>102</b>
7.2.1	The COMMREQ Command Block: General Description.....	102
7.2.2	Establish Read Channel (2003) .....	104
7.2.3	Establish Write Channel (2004).....	108
7.2.4	Send Information Report (2010).....	111
7.2.5	Abort Channel (2001) .....	113
7.2.6	Retrieve Detailed Channel Status (2002).....	114
<b>7.3</b>	<b>Programming for Channel Commands</b> .....	<b>115</b>
7.3.1	COMMREQ Sample Logic .....	116
7.3.2	Sequencing Communications Requests.....	118
7.3.3	Managing Channels and TCP Connections.....	118
7.3.4	Use “Channel Re-Tasking” To Avoid Using Up TCP Connections.....	119
7.3.5	Client Channels TCP Resource Management .....	119
7.3.6	SRTP Application Timeouts .....	120
<b>7.4</b>	<b>Monitoring Channel Status</b> .....	<b>120</b>
7.4.1	Format of the COMMREQ Status Word .....	120
<b>Chapter 8</b>	<b>Modbus/TCP Server</b> .....	<b>123</b>
<b>8.1</b>	<b>Modbus/TCP Server</b> .....	<b>123</b>
8.1.1	Modbus/TCP Server Connections.....	123
8.1.2	Modbus Conformance Classes.....	123
8.1.3	Server Protocol Services.....	123
8.1.4	Station Manager Support.....	123
<b>8.2</b>	<b>Reference Mapping</b> .....	<b>123</b>
8.2.1	Modbus Reference Tables.....	124
8.2.2	Address Configuration .....	125



8.3	<b>Modbus Function Codes</b> .....	<b>126</b>
<b>Chapter 9</b>	<b>Modbus/TCP Client</b> .....	<b>127</b>
9.1	<b>The Communications Request</b> .....	<b>127</b>
9.1.1	Structure of the Communications Request .....	128
9.1.2	COMMREQ Function Block .....	128
9.1.3	COMMREQ Command Block.....	128
9.1.4	Modbus/TCP Channel Commands.....	128
9.1.5	Status Data .....	129
9.1.6	Operation of the Communications Request.....	130
9.2	<b>COMMREQ Function Block and Command Block</b> .....	<b>131</b>
9.2.1	The COMMREQ Function Block .....	131
9.2.2	The COMMREQ Command Block .....	132
9.3	<b>Modbus/TCP Channel Commands</b> .....	<b>133</b>
9.3.1	Open a Modbus/TCP Client Connection (3000) .....	133
9.3.2	Close a Modbus/TCP Client Connection (3001).....	135
9.3.3	Read Data from a Modbus/TCP Device (3003).....	136
9.3.4	Write Data to a Modbus/TCP Device (3004).....	142
9.3.5	Mask Write Register Request to a Modbus Server Device (3009).....	146
9.3.6	Read/Write Multiple Registers to/from a Modbus Server Device (3005) .....	147
9.4	<b>Status Data</b> .....	<b>149</b>
9.4.1	Types of Status Data.....	149
9.5	<b>Controlling Communications in the Ladder Program</b> .....	<b>150</b>
9.5.1	Essential Elements of the Ladder Program.....	150
9.5.2	COMMREQ Ladder Logic Example .....	151
9.5.3	Troubleshooting a Ladder Program.....	157
9.5.4	Monitoring the Communications Channel .....	158
9.6	<b>Differences between Series 90 and PACSystems Modbus/TCP Channels</b> .....	<b>159</b>
<b>Chapter 10</b>	<b>OPC UA Server</b> .....	<b>161</b>
10.1	<b>Application Logic to Control the OPC UA Server</b> .....	<b>162</b>
10.1.1	OPC UA Server Service Request .....	162
10.1.2	OPC UA Server Subroutine .....	170
10.1.3	Connect OPC UA Client to OPC UA Server.....	172
10.1.4	OPC UA Client Authentication Settings .....	175
10.1.5	Anonymous Authentication.....	175
10.1.6	Username/Password Authentication.....	176
10.1.7	OPC UA Security Settings.....	178
10.1.8	OPC UA Address Space .....	178
10.1.9	Publish Application Variables to OPC UA Address Space .....	179
10.1.10	OPC UA Server Information in Address Space.....	180
10.1.11	OPC UA Server – Application Information.....	182

10.1.12	OPC UA Server – GE Device Information .....	183
10.1.13	OPC UA Automatic Restart Function .....	184
10.1.14	OPC UA Server Certificates .....	184
<b>Chapter 11</b>	<b>RX7i PLC Monitoring Via the Web .....</b>	<b>185</b>
11.1	<b>System Requirements .....</b>	<b>185</b>
11.2	<b>Disabling Pop-up Blocking .....</b>	<b>185</b>
11.3	<b>Web Server Operation in a Redundant System.....</b>	<b>185</b>
11.4	<b>Standard Web Pages .....</b>	<b>185</b>
11.4.1	RX7i Home Page.....	186
11.4.2	Factory Default Web Page.....	186
11.4.3	Reference Tables Viewer Page .....	186
11.4.4	PLC Fault Table Viewer Page .....	188
11.4.5	I/O Fault Table Viewer Page.....	190
11.5	<b>Downloading PLC Web Pages .....</b>	<b>190</b>
11.5.1	FTP Connect and Login .....	190
11.5.2	Changing the Password .....	191
11.5.3	Web Page File Transfer .....	191
11.6	<b>Viewing the RX7i PLC Web Pages.....</b>	<b>192</b>
<b>Chapter 12</b>	<b>Diagnostics.....</b>	<b>193</b>
12.1	<b>What to do if You Cannot Solve the Problem.....</b>	<b>193</b>
12.2	<b>Diagnostic Tools Available for Troubleshooting .....</b>	<b>194</b>
12.3	<b>States of the Ethernet Interface (Rack-based and RX7i Embedded Interfaces).....</b>	<b>195</b>
12.4	<b>EOK LED Blink Codes for Hardware Failures (Rack-based and RX7i Embedded Interfaces).....</b>	<b>197</b>
12.5	<b>Controller Fault Table .....</b>	<b>198</b>
12.5.1	Controller Fault Table Descriptions .....	198
12.6	<b>Monitoring the Ethernet Interface Status Bits .....</b>	<b>201</b>
12.6.1	LAN Interface Status (LIS) Bits .....	202
12.6.2	Channel Status Bits.....	203
12.7	<b>Monitoring the FT Output of the COMMREQ Function Block .....</b>	<b>204</b>
12.8	<b>Monitoring the COMMREQ Status Word.....</b>	<b>204</b>
12.8.1	Format of the COMMREQ Status Word .....	205
12.8.2	Major Error Codes in the COMMREQ Status Word.....	206
12.8.3	Minor Error Codes for Major Error Codes 05H (at Remote Server PLC) and 85H (at Client PLC) .....	207
12.8.4	Minor Error Codes for Major Error Code 11H (at Remote Server PLC).....	208
12.8.5	Minor Error Codes for Major Error Code 90H (at Client PLC) .....	210
12.8.6	Minor Error Codes for Major Error Code 91H (at Remote Modbus/TCP Server) .....	212

12.8.7	Minor Error Codes for Major Error Code A0H (at Client PLC).....	213
<b>12.9</b>	<b>Using the EGD Management Tool (Rack-based and RX7i Embedded).....</b>	<b>214</b>
12.9.1	Installing the EGD Management Tool .....	214
12.9.2	Launching the EGD Management Tool .....	214
12.9.3	Monitoring EGD Devices.....	215
12.9.4	Monitoring Status of Ethernet Global Data for a Device.....	216
<b>12.10</b>	<b>Troubleshooting Common Ethernet Difficulties .....</b>	<b>218</b>
12.10.1	COMMREQ Fault Errors .....	218
12.10.2	PLC Timeout Errors.....	219
12.10.3	Application Timeout Errors .....	220
12.10.4	EGD Configuration Mismatch Errors.....	220
12.10.5	Station Manager Lockout under Heavy Load.....	221
12.10.6	PING Restrictions.....	221
12.10.7	SRTP and Modbus/TCP Connection Timeout .....	221
12.10.8	Sluggish Programmer Response after Network Disruption .....	222
12.10.9	EGD Command Session Conflicts.....	222
12.10.10	SRTP Request Incompatibility with Existing Host Communications Toolkit Devices or Other SRTP Clients.....	222
12.10.11	COMMREQ Flooding Can Interrupt Normal Operation.....	222
12.10.12	Accelerated EGD Consumption Can Interfere with EGD Production.....	223
12.10.13	Channels Operation Depends Upon PLC Input Scanning .....	223
<b>Chapter 13</b>	<b>Network Administration .....</b>	<b>225</b>
<b>13.1</b>	<b>IP Addressing .....</b>	<b>225</b>
13.1.1	IP Address Format for Network Classes A, B, C.....	225
13.1.2	IP Addresses Reserved for Private Networks .....	226
13.1.3	Multicast IP Addresses .....	226
13.1.4	Loopback IP Addresses.....	226
<b>13.2</b>	<b>Gateways .....</b>	<b>226</b>
13.2.1	Networks Connected by a Gateway.....	227
<b>13.3</b>	<b>Subnets and Supernetns .....</b>	<b>227</b>
13.3.1	Subnet Addressing and Subnet Masks.....	227
<b>Appendix A</b>	<b>Configuring Advanced User Parameters .....</b>	<b>231</b>
<b>A-1</b>	<b>Format of the Advanced User Parameters File .....</b>	<b>232</b>
<b>A-2</b>	<b>Advanced User Parameter Definitions .....</b>	<b>233</b>
<b>A-3</b>	<b>AUPs Supported by RX3i CPE305/CPE310 Embedded Ethernet Interface .....</b>	<b>240</b>
<b>Index</b>	<b>241</b>	



# Table of Figures

---

Figure 1: Ethernet Connection System Diagram.....	3
Figure 2: Ethernet Operation in Redundancy Mode.....	8
Figure 3: Basic non-HSB System with Redundant IP.....	9
Figure 4: RJ-45 Connector .....	14
Figure 5: Ethernet Cable Routing.....	15
Figure 6: RX7i Faceplate .....	18
Figure 7: MAC Address on RX7i .....	21
Figure 8: MAC Address on RX3i ETM001 Module.....	22
Figure 9: Diagram of Embedded Ethernet Switch.....	23
Figure 10: System Diagram: Ethernet Routing Using Embedded Switch.....	23
Figure 11: Connection Using Hub/Switch/Repeater.....	25
Figure 12: Direct Connection to the Embedded Ethernet Ports.....	26
Figure 13: Expand CPU Slot to Display Ethernet Node .....	30
Figure 14: Expand RX3i CPU Node to Configure Embedded Ethernet Interface.....	31
Figure 15: Ethernet Settings Tab in Proficy Machine Edition .....	31
Figure 16: Terminals Tab Settings in Proficy Machine Edition .....	33
Figure 17: Adding Ethernet Global Data (EGD) to the Configuration .....	34
Figure 18: Defining EGD Produced Data Exchange .....	34
Figure 19: Defining EGD Consumed Data Exchange.....	35
Figure 20: Setting Temporary IP Address.....	37
Figure 21: Expand RX7i CPU Node to Configure Ethernet Daughterboard.....	39
Figure 22: Install ETM001 Module in Rack/Slot & Expand to Configure .....	40
Figure 23: Expand Node to View Ethernet Global Data .....	42
Figure 24: Local Producer ID.....	43
Figure 25: Configuring Redundancy for Ethernet Global Data.....	43
Figure 26: Exchange ID Offset in an Ethernet Redundancy System .....	44
Figure 27: Configuring Produce in Backup Mode Parameter.....	44
Figure 28: Configuring the EGD Configuration Server .....	46
Figure 29: Producing & Consuming Ethernet Global Data .....	58
Figure 30: Adding Symbolic Reference to Ethernet Global Data Exchange.....	60
Figure 31: Grouping of Devices for Ethernet Global Data Multicasting .....	61
Figure 32: Memory Sharing between PLC and Ethernet Interface .....	63
Figure 33: EGB Timing Example #1 .....	64
Figure 34: EGB Timing Example #2.....	65
Figure 35: Obtaining Timestamps from the Ethernet Interface Clock.....	66
Figure 36: Obtaining Timestamps from the PLC Time Clock .....	66
Figure 37: Obtaining Timestamps from the SNTP Server's Time Clock.....	67
Figure 38: Synchronizing CPU Time-of-Day Clock to an SNTP Server.....	68
Figure 39: Operating Sequence for CPU Clock Synchronization.....	69
Figure 40: COMMREQ to Control the CPU Time-of-Day Clock.....	70
Figure 41: COMMREQ Used to Program Ethernet Global Data .....	83
Figure 42: Example: Masked Write to EGD Exchange Bit Mask and Data Bits.....	97
Figure 43: COMMREQ Sequence for Establish Read Channel .....	101

Figure 44: COMMREQ for Programming Channel Commands.....	102
Figure 45: Interpreting Detailed Channel Status Words.....	115
Figure 46: Sample Ladder Logic for COMMREQ.....	117
Figure 47: Interpreting COMMREQ Status Word.....	120
Figure 48: Calculations for Modbus File and Record %W Memory Address.....	124
Figure 49: Phases of a COMMREQ Execution.....	128
Figure 50: Illustration of Phased Operation of a COMMREQ.....	130
Figure 51: The COMMREQ Function Block.....	131
Figure 52: Interpreting the COMMREQ Status Word.....	150
Figure 53: COMMREQ Ladder Logic Segment.....	151
Figure 54: COMMREQ Ladder Logic Segment (continued).....	152
Figure 55: COMMREQ Ladder Logic Segment (continued).....	153
Figure 56: COMMREQ Ladder Logic Segment (continued).....	154
Figure 57: COMMREQ Ladder Logic Segment (continued).....	155
Figure 58: COMMREQ Ladder Logic Segment (continued).....	155
Figure 59: COMMREQ Ladder Logic Segment (continued).....	156
Figure 60: SERVER_STATUS Word bit definitions.....	167
Figure 61: CONFIG_STATUS Word bit definitions.....	169
Figure 62: OPC UA Example Subroutine.....	171
Figure 63: Project Inspector/Ethernet Config Window.....	173
Figure 64: OPC UA Server Client Connection String.....	174
Figure 65: OPC UA Client Connection Dialog.....	174
Figure 66: Machine Edition Controller Hardware Configuration – Passwords Disabled.....	175
Figure 67: Machine Edition Controller Hardware Configuration – Passwords Enabled.....	176
Figure 68: Machine Edition Online Command to Set Passwords.....	177
Figure 69: OPC UA Connection Security Settings.....	178
Figure 70: Example OPC UA Address Space.....	178
Figure 71: Machine Edition Variable Inspector.....	179
Figure 72: Application Variable Address Space.....	180
Figure 73: OPC UA Address Space - Server Node.....	180
Figure 74: Server Specific Address Space.....	181
Figure 75: BuildInfo Subscription.....	181
Figure 76: OPC UA Address Space - Application Information.....	182
Figure 77: OPC UA Address Space – GE Device Information.....	183
Figure 78: PACSystems Factory Default Web Page.....	186
Figure 79: Selecting Display Format.....	186
Figure 80: PLC Fault Table Display.....	188
Figure 81: Fault Extra Data Display.....	189
Figure 82: I/O Fault Table Display.....	190
Figure 83: States of the Ethernet Interface.....	195
Figure 84: Fault Extra Data Example.....	198
Figure 85: Monitoring FT Output in COMMREQ Function Block.....	204
Figure 86: Decoding the COMMREQ Status Word.....	205
Figure 87: EGD Management Tool Screenshot.....	214
Figure 88: EGD Monitoring Tool Monitoring EGD Network.....	215
Figure 89: EGD Management Tool Displaying EGD Exchange Information.....	216
Figure 90: EGD Management Tool Displaying EGD Statistics.....	217

---

Figure 91: EGD Management Tool Displaying List of Variables for an Exchange.....	218
Figure 92: IP Address Format for Network Classes A, B, C .....	225
Figure 93: Gateway Connected to Two Networks .....	227
Figure 94: Class B Network netid and hostid Bit Formats .....	227
Figure 95: Use of Subnet Mask .....	228
Figure 96: Network 2 Divided into Subnets 2.1 and 2.2.....	228
Figure 97: Subnet Mask Used to Effect a Supernet .....	229
Figure 98: Resulting Supernet.....	229





# Chapter 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:

**Chapter 2, Installation and Startup: RX3i Embedded Interfaces** describes user features and basic installation procedures.

**Chapter 3, Installation and Startup: Rack-based and RX7i Embedded Interfaces** describes user features and basic installation procedures.

**Chapter 4, Configuration** describes assigning a temporary IP address and configuring the Ethernet interface parameters. For the rack-based and RX7i embedded interfaces, describes how to configure Ethernet Global Data (EGD) and set up the RS-232 port for Local Station Manager operation.

**Chapter 5, Ethernet Global Data** describes basic EGD operation for rack-based and RX7i embedded interfaces.

**Chapter 6, EGD Commands** describes a set of commands that can be used in the application program to read and write PLC data or Ethernet Global Data exchange data over the network.

**Chapter 7, Programming SRTP Channel Commands** explains how to implement PLC to PLC communications over the Ethernet network using Service Request Transfer Protocol (SRTP) Channel commands.

**Chapter 8, Modbus TCP Server** describes the implementation of the Modbus TCP Server feature for the PACSystems family of products.

**Chapter 9, Modbus TCP Client** explains how to program communications over the Ethernet network using Modbus TCP Channel commands.

**Chapter 10, OPC UA Server**, explains how to program communications for this protocol using the embedded Ethernet port.

**Chapter 11, RX7i PLC Monitoring Via the Web** describes the Web browser feature provided by a PACSystems RX7i CPU with Embedded Ethernet.

**Chapter 12, Diagnostics** describes diagnostic techniques for a PACSystems Ethernet Interface. This chapter also lists COMMREQ Status codes.

**Chapter 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 optional configuration of internal operating parameters used by the Ethernet interface. For most applications, the default Advanced User Parameters (AUPs) should not be changed.

**Note:** The RX3i CPE305/CPE310 **embedded** Ethernet interface provides a maximum of two programmer connections. It does not support the full set of features described in this manual. For a summary of RX3i embedded Ethernet interface features, refer to page 4.

## 1.1 Revisions in this Manual

**Note:** 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 RX7i* includes the following changes: Information about the following new features for the CPE305/CPE310 embedded Ethernet interface:

Rev	Date	Description
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> <li></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 Chapter 2 to Chapter 12. Configuration information has been moved to Chapter 4.</p> <p>Information about Channel Status bits has been removed from chapters 2, 7 and 9, and consolidated in Chapter 12.</p>

## 1.2 Other PACSystems Manuals

The manuals listed below provide more information about the PACSystems family of products.

- *PACSystems CPU Reference Manual*, GFK-2222
- *TCP/IP Ethernet Communications for PACSystems Station Manager Manual*, GFK-2225
- *PACSystems RX7i Installation Manual*, GFK-2223
- *PACSystems Hot Standby CPU Redundancy User's Guide*, GFK-2308
- *PACSystems RX3i System Manual*, GFK-2314
- *PACSystems RX3i Ethernet NIU User Manual*, GFK-2439
- *PACSystems RX3i and RX7i Controllers Battery Manual*, GFK-2741
- *Proficy\* Logic Developer-PLC Getting Started*, GFK-1918

In addition to these manuals, datasheets and Important Product Information documents describe individual modules and product revisions. The most recent PACSystems documentation is available online on the Support website.

## 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)<sup>1</sup> 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.

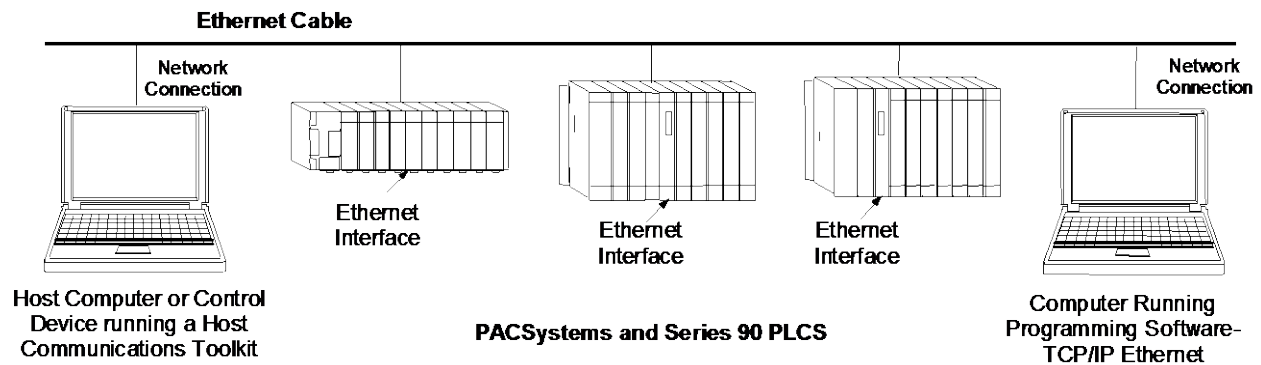


Figure 1: Ethernet Connection System Diagram

### 1.3.1 Rack-based and RX7i Embedded Interfaces - Features

**Note:** The RX3i CPE305/CPE310 embedded Ethernet interface supports a subset of these features. For a list of features provided by the RX3i embedded Ethernet, see page 4.

- Full RX3i Controller programming and configuration services with 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.
- Basic remote PLC monitoring from a web browser (RX7i CPU Ethernet interface only)
- Comprehensive station management and diagnostic tools
- Extended controller connectivity via IEEE 802.3 CSMA/CD 10Mbps and 100Mbps Ethernet LAN port connectors.
- Network switch that has Auto negotiate, Sense, Speed, and crossover detection.

<sup>1</sup> Effective with RX3i CPE305/CPE310 firmware version 8.30, EGD Class 1 is supported on the embedded Ethernet Interface. Earlier versions do not support EGD.

- Direct connection to BaseT (twisted pair) network switch, hub, or repeater without an external transceiver.
- 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.
- Internet access via web pages served up to standard web browsers, for the Ethernet interface embedded in the PACSystems RX7i CPU.

### **1.3.2      *RX3i CPE305/CPE310 Embedded Ethernet Interface - Features***

- Full RX3i controller programming and configuration services with 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.
- Comprehensive station management and diagnostic tools. For supported commands, refer to the *Station Manager Manual*, GFK-2225J or later.
- Extended controller connectivity via IEEE 802.3 CSMA/CD 10Mbps and 100Mbps Ethernet LAN port connectors.
- Network switch that has Auto negotiate, Sense, Speed, and crossover detection.
- Direct connection to BaseT (twisted pair) network switch, hub, or repeater without an external transceiver.
- Communications with remote PLCs and other nodes reachable through routers. The gateway IP address must be configured.

### 1.3.3 Ethernet Interface Specifications

#### All RX7i Ethernet Interface Modules and RX3i Rack-Based Ethernet Interface Modules

Connectors	- Two 10BaseT / 100BaseTX Ports: 8-pin female shielded RJ-45, autosensing - Station Manager (RS-232) Port: 9-pin female D-connector
LAN	IEEE 802.2 Logical Link Control Class I IEEE 802.3 CSMA/CD Medium Access Control 10/100 Mbps
Number of IP addresses	One
Maximum number of connections	48 SRTP Server connections. Includes: <ul style="list-style-type: none"> <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	Station Mgr Port: RS-232 DCE, 1200 - 115200 bps.
Station Manager	Access via local serial port or remote UDP. <i>Refer to the Station Manager Manual, GFK-2225J or later for supported commands.</i>
Maximum ETM001 modules per CPU rack	RX7i: seven RX3i: eight

#### RX3i Embedded Interface

Connector	One 10BaseT / 100BaseTX Port: 8-pin female shielded RJ-45, autosensing
LAN	IEEE 802.2 Logical Link Control Class I IEEE 802.3 CSMA/CD Medium Access Control 10/100 Mbps
Number of IP addresses	One
Maximum number of connections	32 SRTP Server connections. Includes: <ul style="list-style-type: none"> <li>▪ Up to 16 Modbus/TCP Server connections</li> <li>▪ Up to 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>
Station Manager	Access remote UDP <i>Refer to the Station Manager Manual, GFK-2225J or later for supported commands.</i>

### 1.3.4 Ethernet Interface Ports

The PACSystems Ethernet interface use auto-sensing 10Base T / 100Base TX RJ-45 shielded twisted pair Ethernet ports for connection to either a 10BaseT or 100BaseTX IEEE 802.3 network. The RX3i embedded Ethernet interface provides one such port; all other models provide two.

The port automatically senses the speed (10Mbps or 100Mbps), 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 10BaseT/100BaseTX media via its network ports.

**10BaseT:** 10BaseT 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 6 to 12 nodes connected in a star wiring topology.

**100BaseTX:** 100BaseTX 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.

### 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 addressing. 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 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.

### 1.3.7 Built-In Web Server

The embedded RX7i CPU Ethernet Interface provides Web Server capability. Each IC698 Ethernet interface supports Web access via FTP and HTTP to allow Web pages to be stored and maintained on the Ethernet interface and served up via the web to standard Web browsers. A standard API allows you to generate customized web pages that display desired PLC data in a desired format. You store the Web pages to the Ethernet interface via FTP. A basic set of predefined Web pages in English are provided; they include a home page, Reference Table data, Controller Fault Table, and I/O Fault Table. Rack-based Ethernet Interface modules do not provide Web Server capability.

### 1.3.8 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 "Ethernet Interface Specifications" on page 5.

### 1.3.9 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 “Ethernet Interface Specifications” on page 5.

### 1.3.10 Ethernet Global Data (EGD)

Each PACSystems CPU supports up to 255 simultaneous Ethernet Global Data (EGD) exchanges. EGD exchanges are configured using the programmer and stored into 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.

The PACSystems Ethernet interface can be configured to use SNTP (Simple Network Time Protocol) to synchronize the timestamps of produced EGD exchanges.

Beginning with PACSystems release 2.00, PACSystems Ethernet interfaces implement the capabilities of a Class 1 and Class 2 device. COMMREQ-driven EGD Commands can be used in the application program to read and write data into PACSystems PLCs or other EGD Class 2 devices.

PACSystems releases 5.5 and later support run mode store of EGD so that you can add, delete or modify EGD exchanges without stopping the controller. For details on using this feature, refer to “Run-Mode Store of EGD” in Chapter 5.

### 1.3.11 SRTP Inactivity Timeout

Starting with Release 6.00, the PACSystems Ethernet interface supports inactivity checking on SRTP server connections with any Proficy Machine Edition (PME) PLC programmer. With this feature, the Ethernet interface removes an abandoned SRTP server connection and all of its resources when there is no activity on the connection for a specified timeout interval. (For example, when communication with the programmer is lost.) Until the server connection is removed, other programmers cannot switch from Monitor to Programmer mode.

Without the SRTP inactivity timeout, an abandoned SRTP server connection persists until the underlying TCP connection eventually 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). Revision 6.00 and higher PME programmers can override the initial timeout value on a particular 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 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<sup>2</sup>

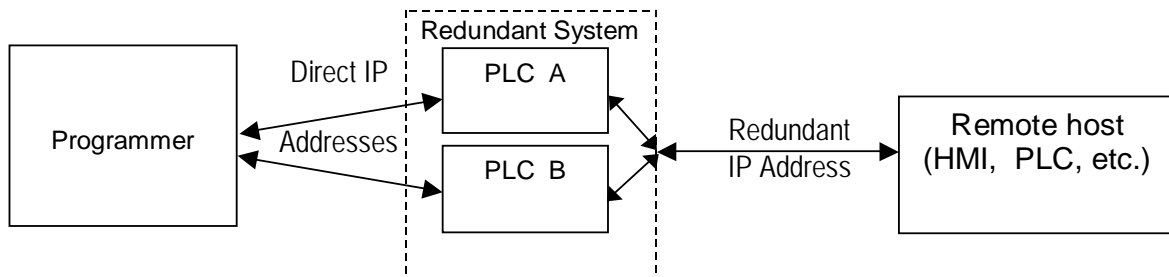
The Redundant IP feature of the Ethernet Interface allows a single IP address called the Redundant IP address to be assigned to two Ethernet modules. The two modules are in two different PLCs that are configured as a redundant system.

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 other unit (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 individual direct IP addresses, permitting programmer connection to the active or backup PLC at any time.



**Figure 2: Ethernet Operation in Redundancy Mode**

The Redundant IP feature is supported by Hot Standby (HSB) CPUs and non-HSB CPUs.

### 1.4.1 HSB CPU Redundancy

An HSB system uses redundancy CPUs that 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. Redundancy CPUs that include an embedded Ethernet Interface have a “CRE” designation, for example IC698CRE040. For information about HSB architectures, refer to the *PACSystems Hot Standby CPU Redundancy User's Guide*, GFK-2308.

<sup>2</sup> Not supported on the RX3i CPE305/CPE310 embedded Ethernet Interface.



## 1.4.2 Non-HSB Redundancy

Non-HSB redundancy systems use RX7i or 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. The figure below 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.

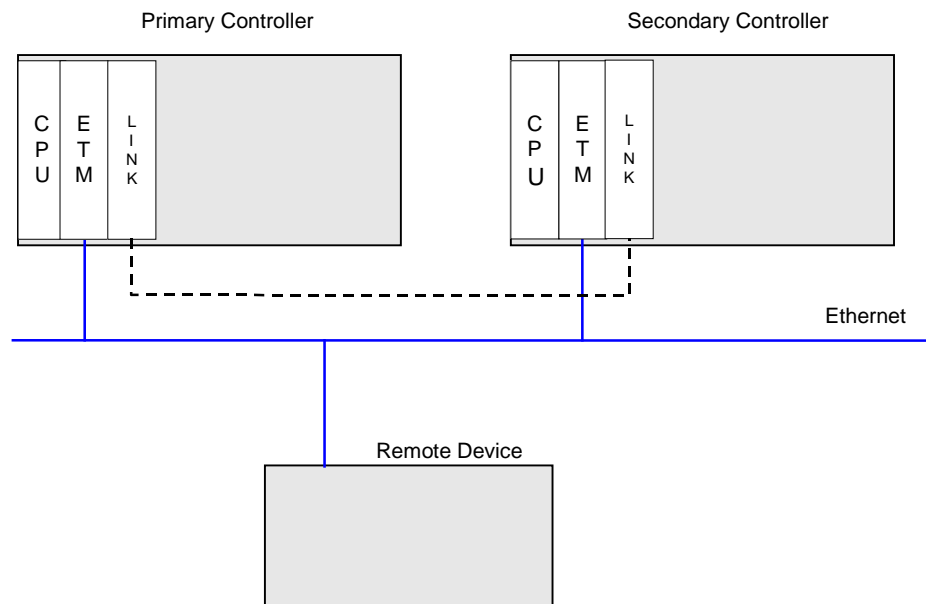


Figure 3: Basic non-HSB System with Redundant IP

## 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 has no knowledge of 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.

To remote hosts on the network, the redundant system is viewed as a single PLC with high reliability; the remote host neither knows nor cares which PLC is 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 redundancy CPU when the active unit detects a fatal fault, is placed in Stop mode, or is powered off. An HSB role switch can also be initiated manually or by the application logic. 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 "Monitoring the Ethernet Interface Status Bits" in Chapter 12

**Note:** The application must allow sufficient time for Redundant IP activation (at least 120msec) 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 a 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 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 CPU Reference Manual*, GFK-2222.

This function has an input parameter block with a length of one word.

<b>address</b>	0=Backup redundancy role
	1=Active redundancy role

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 has no effect on 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 is able to 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 is able to 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 interfaces changes to backup, it stops production of all EGD exchanges.

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. This permits 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 any unicast exchanges at the Redundant IP address.

### **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 backup, any EGD command currently in process over the Redundant IP address is ended.

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 interfaces changes to backup, any EGD commands in process are ended.

Although not recommend, EGD commands may be issued to the direct IP address. 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 backup, it takes down all Web server connections and their underlying TCP connections. The Web server maintains its underlying TCP connection only long enough to process each web page request; a new TCP connection is opened, used, and closed for each subsequent Web page display or update. So unless a Web page change or update is requested during the redundancy role switch, the operation of the Redundant IP address is transparent to the Web remote browser. 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**

FTP operations are used to transfer setup and configuration data to the Ethernet interface, not for communication with the actual PLC application. Therefore, 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, so it only receives 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.

### **1.4.13 Remote Station Manager Operation in a Redundancy System**

The remote Station Manager should respond to the direct IP address regardless of whether the unit is active or backup, or whether or not 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 actually 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).

## Chapter 2 Installation and Start-up: RX3i Embedded Interface

The RX3i CPE305 and CPE310 (CPE3xx) CPUs provide an embedded Ethernet interface for programmer communications. This chapter 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
- Pinging TCP/IP Ethernet Interfaces on the Network

**Note:** Effective with RX3i CPE310/CPE305 Firmware Release 8.30, the CPU itself also supports EGD<sup>3</sup> Class 1. Prior to that firmware release, EGD was only available in the RX3i via the RX3i Ethernet Interface module (ETM001).

**Note:** For features, installation and startup of the RX3i rack-based Ethernet module (ETM001), see Chapter 3.

### 2.1 RX3i Embedded Ethernet Interface Indicators

The Ethernet port has 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

LED	LED State ● On    ✚ Blinking    ○ Off	Ethernet Port State
<b>100</b>	● On, Green	Network data speed is 100 Mbps.
	○ Off	Network data speed is 10 Mbps.
<b>LINK</b>	● On, Amber	The link is physically connected.
	✚ Blinking, Amber	Traffic is detected at the port.
	○ Off	The Ethernet port is not physically connected.

#### 2.1.2 Module Installation

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

<sup>3</sup> Proficy Machine Edition Release 8.50 SIM 6 is required for EGD on CPE305/CPE310.

## 2.2 Ethernet Port Connector

The RX3i CPE305 and CPE310 CPUs provide a 10BaseT/100BaseTX Ethernet network port connector.

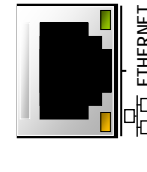


Figure 4: RJ-45 Connector

**Note:** Although the CPE310 can be configured as a CPU310 for backward compatibility, an Ethernet cable should not be connected to the device when it is configured as a CPU310. Ethernet is not supported when CPE310 is configured as a CPU310 and the Ethernet port should **not** be connected to any network as it may have adverse effects on the network and/or operation of the CPU.

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

Either shielded or unshielded twisted pair cable may be attached to a port. The 10Base-T/100Base Tx twisted pair cable must meet the applicable IEEE 802 standards. Category 5 cable is required for 100BaseTX operation. The Ethernet port automatically senses the speed (10Mbps or 100Mbps), 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>4</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.

<sup>4</sup> Pin 1 is at the bottom right of the Station Manager port connector as viewed from the front of the module.

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

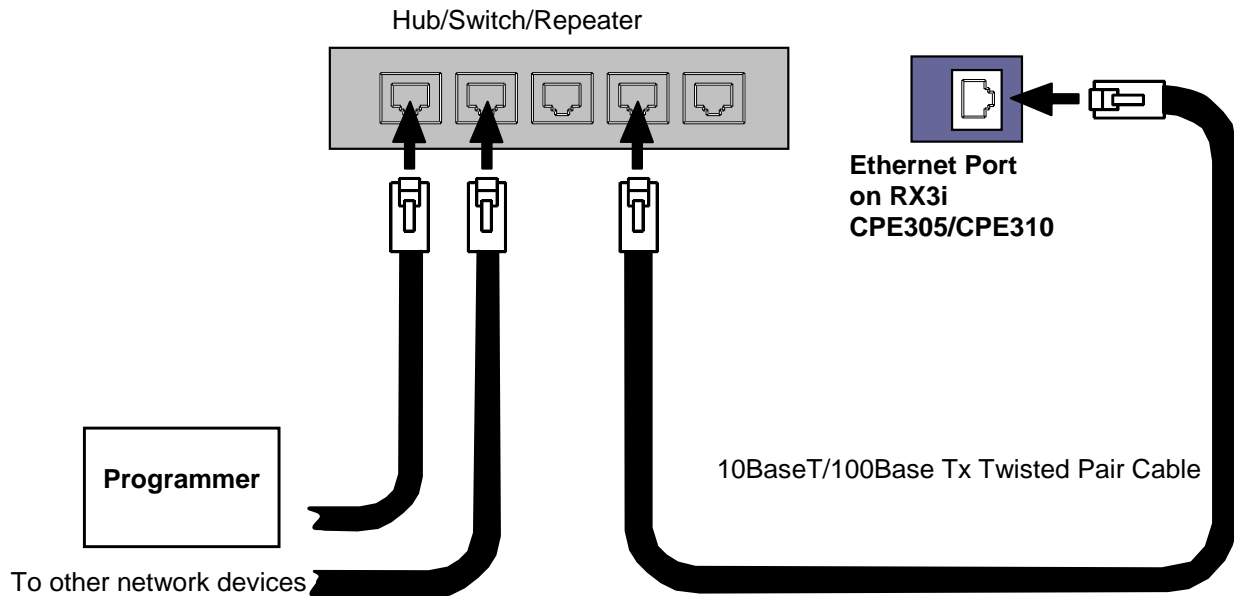


Figure 5: Ethernet Cable Routing

## 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 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 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 Chapter 4.

### 2.3.1 Pinging the Ethernet Interface from a UNIX Host or Computer Running TCP/IP Software

A *ping* command can be executed from a UNIX host or computer running TCP/IP (most TCP/IP communications software provides a *ping* command) or from another Ethernet interface. When using a computer or UNIX host, you can refer to the documentation for the *ping* command, but in general all that is required is the IP address of the remote host as a parameter to the *ping* command. For example, at the command prompt type:

```
ping 10.0.0.1
```

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

**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.

*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.





## Chapter 3 Installation and Start-up: Rack-based and RX7i Embedded Interface

---

This chapter describes the Ethernet Interface's user features and basic installation procedures.

- Ethernet Interface Controls and Indicators
  - Ethernet LEDs
  - Ethernet Restart Pushbutton
- Module Installation
  - RX7i CPU with Embedded Ethernet Interface
  - Rack-based Ethernet Interface Modules
- Ethernet Port Connectors
  - Embedded Switch
  - Connection to a 10Base T / 100Base Tx Network
- Station Manager Port
- Verifying Proper Power-Up of the Ethernet Interface After Configuration
- Pinging TCP/IP Ethernet Interfaces on the Network

Features of the embedded RX7i CPU Ethernet Interface and the rack-based RX3i/RX7i Ethernet interfaces are the same unless noted otherwise.

**Note:** For features, installation and startup of the RX3i embedded Ethernet interface, see Chapter 2.

### 3.1 Ethernet Interface Controls and Indicators

Features of the RX7i embedded CPU Ethernet Interface and the RX7i and RX3i rack-based Ethernet Interface modules are the same unless noted otherwise.

The Ethernet Interface provides:

1. Seven light emitting diode (LED) indicators
  - Ethernet Module OK (EOK)
  - LAN Online (LAN)
  - Status (STAT)
  - Two Ethernet network activity LEDs (LINK)
  - Two Ethernet network speed LEDs (100)
2. Ethernet Restart Pushbutton
3. Two 10BaseT/100BaseTX Ethernet network port connectors. There is only one interface to the network (only one Ethernet address and only one IP address).
4. Station Manager (RS-232) serial port

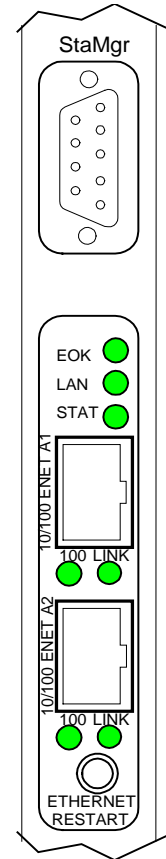


Figure 6: RX7i Faceplate

### 3.1.1 Ethernet LEDs

The LEDs indicate the state and status of the Ethernet Interface.

<i>RX7i Embedded and Rack-Based</i>	<i>RX3i Rack-Based</i>	<i>LED State</i>			<i>Indicates</i>
		● On	✚ Blinking	○ Off	
EOK LAN STAT	ETHERNET OK LAN OK LOG EMPTY	✚ ○ ○	Fast Blink Off Off	Performing Diagnostics	
EOK LAN STAT	ETHERNET OK LAN OK LOG EMPTY	✚ ○ ○	Slow Blink Off Off	Waiting for Ethernet configuration from CPU	
EOK LAN STAT	ETHERNET OK LAN OK LOG EMPTY	✚ ● ✚ ○ ✚	Slow Blink <sup>5</sup> On/Traffic/Off Slow Blink <sup>5</sup>	Waiting for IP Address	
EOK LAN STAT	ETHERNET OK LAN OK LOG EMPTY	● ● ✚ ○ ● ○	On On/Traffic/Off On/Off	Operational	
EOK LAN STAT	ETHERNET OK LAN OK LOG EMPTY	✚ ○ ○	Blink error code Off Off	Hardware failure. See Chapter 12 for blink code definitions.	
EOK LAN STAT	ETHERNET OK LAN OK LOG EMPTY	✚ ✚ ✚	Slow Blink <sup>6</sup> Slow Blink <sup>6</sup> Slow Blink <sup>6</sup>	Firmware Update	

#### LAN LED Operation

The LAN LED (LAN OK on the RX3i Ethernet module) indicates access to the Ethernet network. During normal operation and while waiting for an IP address, the LAN 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 firmware update is occurring.

#### STAT LED Operation

The STAT LED (LOG EMPTY on the RX3i Ethernet module) indicates the condition of the Ethernet interface in normal operational mode. If the STAT LED is off, an event has been entered into the exception log and is available for viewing via the Station Manager interface. The STAT LED is on during normal operation when no events are logged.

In the other states, the STAT LED is either off or blinking and helps define the operational state of the module.

<sup>5</sup> EOK and STAT blink in unison.

<sup>6</sup> All LEDs blink in unison; pattern same for awaiting or performing load.

### EOK LED Operation

The EOK LED (ETHERNET OK on the RX3i Ethernet module) indicates whether the module is able to perform normal operation. This LED is on for normal operation and flashing for all other operations. When a hardware or unrecoverable runtime failure occurs, the EOK LED blinks a two-digit error code identifying the failure. For a list of blink codes and their meanings, see Chapter 12.

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

Each of the two Ethernet ports (Ports 1A and 1B) has 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. Traffic at the port does not necessarily mean that traffic is present at the Ethernet interface, since the traffic may be going between ports of the switch.

### 3.1.2 Ethernet Restart Pushbutton

The Ethernet Restart pushbutton is used to manually restart the Ethernet firmware without power cycling the entire system. It is recessed to prevent accidental operation.

#### Restart Pushbutton Operation for Version 3.6 and Later

For PACSystems Ethernet interfaces version 3.6 and later, an Ethernet restart occurs when the Restart pushbutton is released. The duration that the Restart pushbutton is pressed determines the operation after the restart occurs. In all cases, the EOK, LAN and STAT LEDs briefly turn on in unison as an LED test. The Ethernet port LEDs are not affected by a manual restart of the Ethernet firmware.

To restart the Ethernet interface normally, press the Ethernet Restart pushbutton for less than 5 seconds.

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 pushbutton between 5 and 10 seconds.

To restart the Ethernet interface 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.

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

<b>Restart Operation</b>	<b>Depress Ethernet Restart pushbutton for</b>	<b>Ethernet LEDs Illuminated</b>
Restart the Ethernet interface normally, and start any optional Ethernet plug-in applications that are being used.	Less than 5 seconds	EOK, LAN, STAT
Restart the Ethernet interface without starting any Ethernet plug-in applications.	5 to 10 seconds	LAN, STAT
Restart the Ethernet interface into firmware update operation.	More than 10 seconds	STAT

When forced into firmware update operation, but before the firmware update actually begins, pressing the Ethernet Restart pushbutton again exits the firmware update mode and restarts with the existing firmware. Once the firmware update actually begins, the existing firmware is erased and the Ethernet Restart pushbutton is disabled until the firmware update is complete.

#### Restart Pushbutton Operation Prior to Version 3.6

For PACSystems Ethernet interfaces earlier than version 3.6, pressing the Ethernet Restart pushbutton restarts the module immediately. The EOK, LAN and STAT LEDs briefly turn on in unison as an LED test. These three LEDs are turned on for ½ second and are then turned off when the firmware is restarted. The Ethernet port LEDs are not affected by a manual restart of the Ethernet firmware.

## 3.2 Module Installation

For general information about module and system installation, or if the installation requires CE Mark compliance, refer to the *PACSystems RX7i Hardware Installation Manual*, GFK-2223 or the *PACSystems RX3i System Manual*, GFK-2314.

### 3.2.1 Installing an RX7i CPU with Embedded Ethernet Interface



#### Warning

Do not insert or remove the CPU module with power applied. This could cause the CPU to stop, damage the module, or result in personal injury.

1. Record the 12-digit hexadecimal MAC Address from the printed label located on the rear wall of CPU battery compartment. The label is visible when the battery is removed from its compartment. (The battery does not need to be disconnected to temporarily remove it from the compartment.) For compatible batteries and battery installation procedures for specific CPUs, refer to the *PACSystems RX3i and RX7i Controllers Battery Manual*, GFK-2741.
2. Install the CPU in the rack. Refer to *PACSystems RX7i Hardware Installation Manual*, GFK-2223 for installation instructions.
3. Set the PLC to Stop mode via the Run/Stop switch or the programming software.

### 3.2.2 Installing an RX7i Ethernet Interface Module

1. Record the 12-digit hexadecimal MAC Address from the printed label on the Ethernet Interface. The label is visible only with module out of the rack.
2. Be sure the rack power is OFF.
3. Slide the module into the slot for which it was configured in the system. (Must go into main rack.)
4. Press the module firmly in place, but do not force the module. Tighten the screws on the top and bottom tabs.
5. Connect one or both of the network ports on the Ethernet Interface to the Ethernet network.
6. Turn on power to the PACSystems rack.
7. Set the PLC to Stop mode via the Run/Stop switch or the programming software.

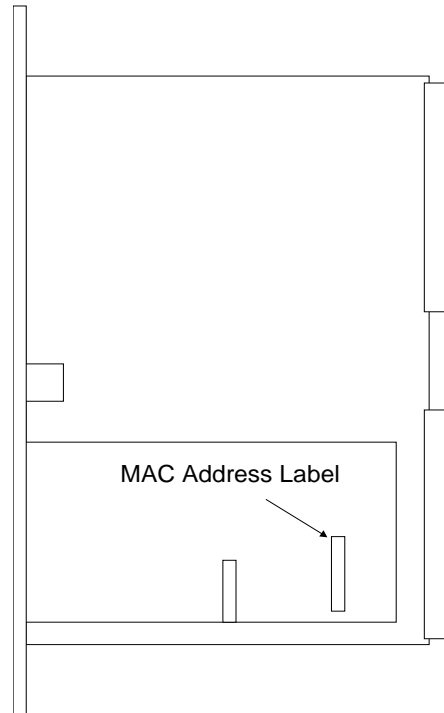


Figure 7: MAC Address on RX7i

### 3.2.3 Installing an RX3i Ethernet Interface Module

1. Record the 12-digit hexadecimal MAC Address from the printed label located on the front of the Ethernet Module.
2. PLC rack power may be off or on (“hot insertion”). For hot insertion, be sure that all cables are disconnected from the Ethernet module
3. Slide the module into the slot for which it was configured in the system. (Must go into main rack.)
4. Press the module firmly in place, but do not force.
5. Connect one or both of the network ports on the Ethernet Interface to the Ethernet network.
6. Unless this is a hot insertion, turn on power to the PACSystems rack.

Set the PLC to Stop mode via the Run/Stop switch or the programming software

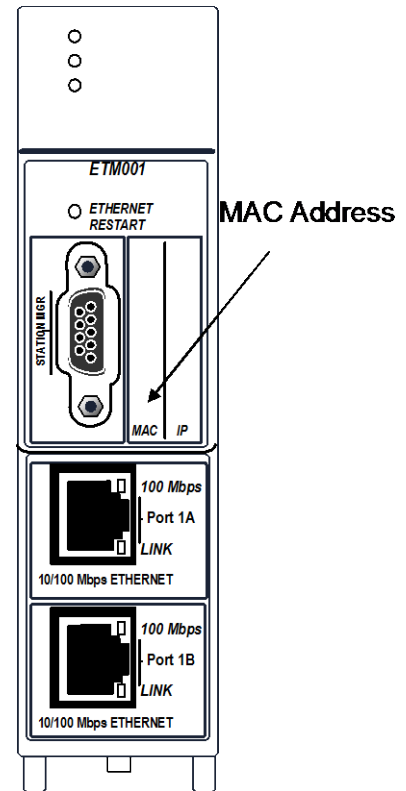


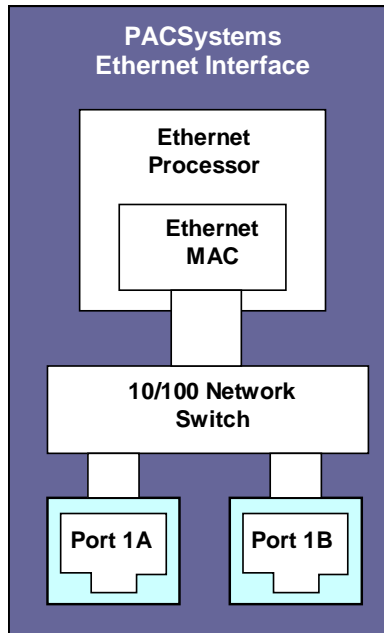
Figure 8: MAC Address on RX3i ETM001 Module

### 3.3 Ethernet Port Connectors

The Ethernet Interface has two Ethernet port connectors, each of which supports both 10Base-T and 100Base-Tx operation using either full-duplex or half-duplex operation. These 8-pin RJ-45 connectors are used to connect the Ethernet Interface to a hub, repeater, switch, or other Ethernet device.

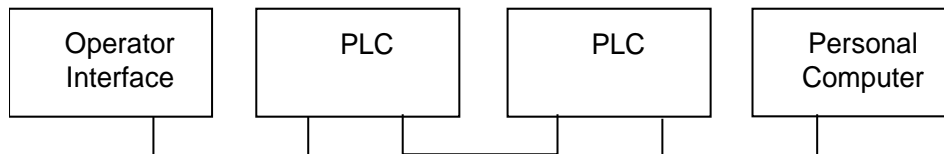
#### 3.3.1 Embedded Switch

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 9: Diagram of Embedded Ethernet Switch**

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



**Figure 10: System Diagram: Ethernet Routing Using Embedded Switch**

It is possible to daisy-chain PLCs together without additional components, but that should be done with great care. Power loss or reset at an Ethernet interface causes loss of communication to any devices downstream from that Ethernet interface in the daisy chain. Restarting the Ethernet interface (via the Ethernet Restart pushbutton, for example) disrupts daisy chain communication.

Each switch port auto-negotiates (by default) to the correct link speed and duplex mode for the device connected to the other end of the link. Each port operates independently, so devices at two different speeds and/or duplex modes may be attached to the two ports. Each port also automatically detects the attached cable and will work properly with either straight-through or crossover cables (by default).



### **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, otherwise duplication of packets and network overload may occur.

---



### **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. In the words of 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 are 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 Network**

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

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



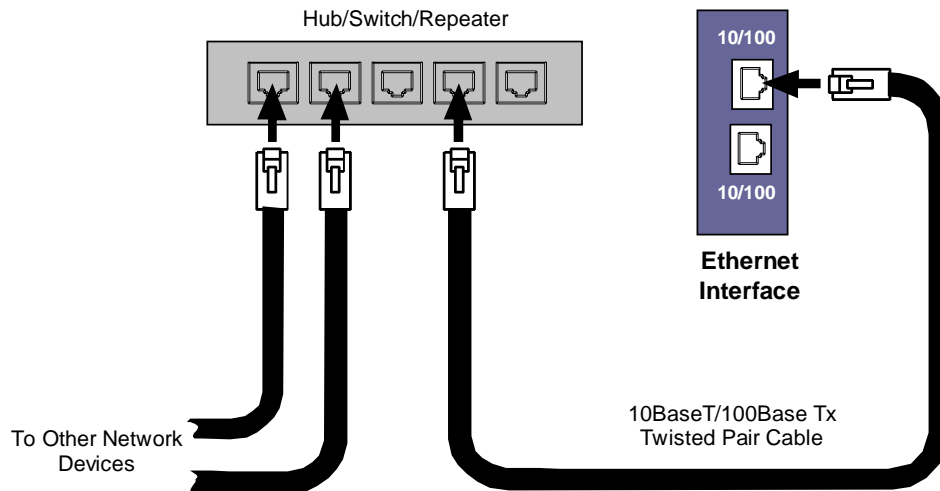
### 10Base-T/100Base Tx Port Pinouts

Pin Number <sup>4</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.

### Connection Using a Hub/Switch/Repeater

Connection of the Ethernet Interface to a 10Base-T or 100Base-Tx network is shown below.



**Figure 11: 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 in regards to 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:

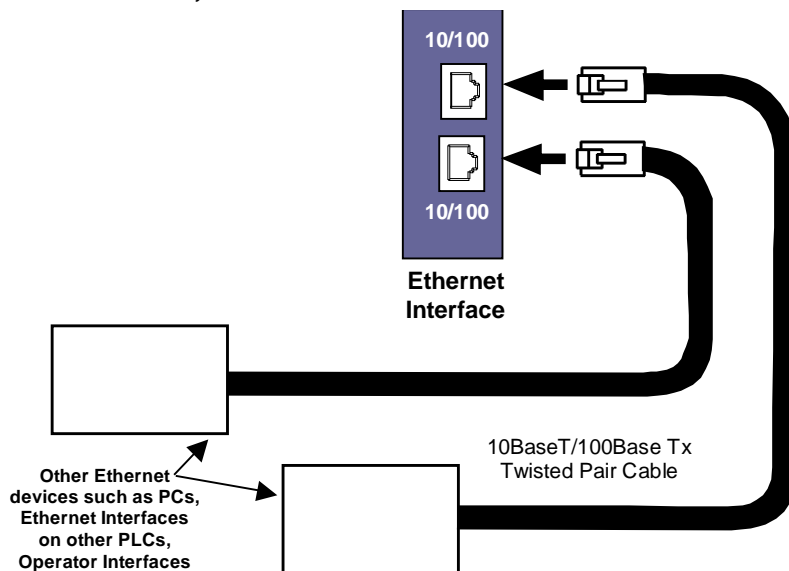


Figure 12: Direct Connection to the Embedded Ethernet Ports

## 3.4 Station Manager Port

The RX7i and rack-based RX3i Ethernet interfaces provide a dedicated RS-232 serial port for local Station Manager use. This nine-pin D 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:

IC200CBL001 Cable, CPU Programming

### 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>4</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 Chapter 4, turn power OFF to the CPU for 3–5 seconds, then turn the power back ON. This starts a series of diagnostic tests. The EOK LED will blink indicating the progress of power-up.

The Ethernet LEDs will have the following pattern upon successful power-up. At this time the Ethernet Interface is fully operational and on-line.

LED	Ethernet Interface Online
EOK	 On
LAN	   On, Off, or blinking, depending on network activity
STAT	 On

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 “Diagnostics,” Chapter 12 for corrective action.

## 3.6 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 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 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 Chapter 4.

### 3.6.1 Pinging the Ethernet Interface from a UNIX Host or Computer Running TCP/IP Software

A *ping* command can be executed from a UNIX host or computer running TCP/IP (most TCP/IP communications software provides a *ping* command) or from another Ethernet Interface. When using a computer or UNIX host, you can refer to the documentation for the *ping* command, but in general all that is required is the IP address of the remote host as a parameter to the *ping* command. For example, at the command prompt type:

```
ping 10.0.0.1
```

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

**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.

*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.

## 3.7 Ethernet Plug-in Applications

Ethernet interface versions 3.6 and later support 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 overwritten by WinLoading another Ethernet plug-in application with the same number, or it is explicitly deleted via the *pluginapp* Station Manager command (see *TCP/IP Ethernet Communications for PACSystems Station Manager Manual*, GFK-2225).

All Ethernet plug-in applications are started during normal Ethernet power-up or restart. During troubleshooting, the Ethernet Restart pushbutton may be used to startup the Ethernet interface without the plug-in applications (see "Ethernet Restart Pushbutton").

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

## Chapter 4 Configuration

---

Before you can use the Ethernet Interface, you must configure it using Machine Edition Logic Developer-PLC software.

This chapter includes configuration information for:

RX3i Embedded Ethernet Interface	page	29
Rack-based and RX7i Embedded Ethernet Interfaces		35

### 4.1 *RX3i Embedded Ethernet Interfaces*

#### 4.1.1 *Ethernet Configuration Data*

The PACSystems PLC is configured exclusively by the Machine Edition Logic Developer-PLC programmer. For 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 auto-configuration.

##### ***Generating / Storing / Loading the Configuration***

The RX3i embedded Ethernet interface is configured as a sub-module of the CPE CPU module. The RX3i embedded Ethernet Interface uses Ethernet Configuration and optional Advanced User Parameter (AUP) Configuration. Both are generated at the Programmer, stored from the Programmer to the PLC as part of the hardware configuration Store sequence, and 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 Appendix A for details.) Once stored to the PLC, the CPU maintains the Ethernet configuration data in non-volatile memory over power cycles.

##### ***Backup Configuration Data***

The RX3i 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***

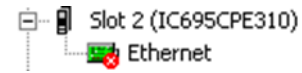
The embedded Ethernet configuration and AUP configuration cannot be locally edited via Station Manager. All configuration changes must be performed via the programmer.

## 4.1.2 Initial IP Address Assignment

The RX3i embedded Ethernet Interface comes from the factory with a default IP default IP address (192.168.0.100). This address is intended only for initial connection in order 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.

1. Using Proficy Machine Edition software, configure the CPE3xx CPU in an RX3i 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 13: Expand CPU Slot to Display Ethernet Node**

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.

**Note:** These CPUs do **not** support the alternate methods of setting a temporary IP address: the Set Temporary IP Address tool in PME, BOOTP or the Station Manager CHSOSW command.

2. Go online with the target and download the configuration. You can use one of the following methods for the initial connection to the CPE3xx:
  - Through the embedded Ethernet port, using the factory-loaded default IP address (192.168.0.100). To set the IP address that PME will 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.

- Through the Ethernet connection of an ETM001 in the same rack with a known IP address configuration.
- Through the RS-232 COM1 serial port – This is a DCE (data communications equipment) port that allows a simple straight-through cable to connect with a standard nine-pin AT-style RS-232 port.
- CPE310: Through the RS-485 COM2 serial port – Use SNP programming cable IC690ACC901

### 4.1.3 Configuring the Ethernet Interface Parameters

#### Configuring an RX3i 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".
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.

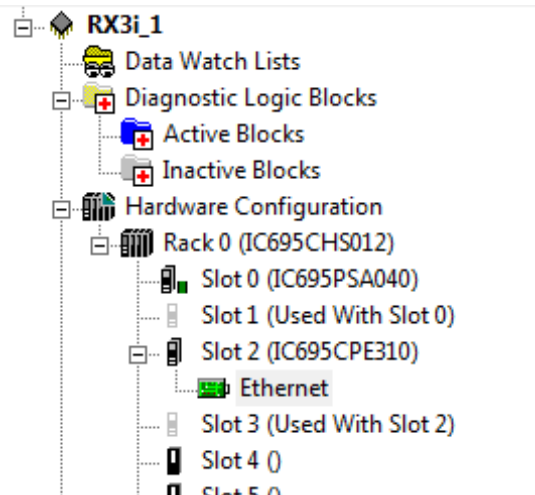


Figure 14: Expand RX3i CPU Node to Configure Embedded Ethernet Interface

#### Ethernet Parameters (Settings Tab)

The screenshot shows the 'Settings' tab for the Ethernet interface. The Navigator on the left is expanded to 'Ethernet'. The InfoViewer on the right shows the following parameters:

Parameters	
Configuration Mode	TCP/IP
Adapter Name	0.2.0
IP Address	10.10.0.12
Subnet Mask	255.255.255.0
Gateway IP Address	10.10.0.1
Network Time Sync	SNTP
Status Address	%I00001
Length	80
I/O Scan Set	1

Figure 15: Ethernet Settings Tab in Proficy Machine Edition

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

**Adapter Name:** This is automatically generated based upon the rack/slot location of the Ethernet interface.

**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, otherwise the Ethernet Interface may be unable to communicate on the network and/or network operation 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:** Key in the desired mask in the format indicated. Learn more about subnet mask usage at Subnet Addressing and Subnet Masks on page 227.

**Gateway IP Address:** Key in the desired Gateway IP Address in the format indicated. Learn more about Gateways on page 226.

**Network Time Sync:** Options are “None” and “SNTP”. Select SNTP (Simple Network Time Protocol) 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 LAN 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 “Monitoring the Ethernet Interface Status Bits” in Chapter 12, “Diagnostics.”

**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).

**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.



## Terminals Tab

This configuration tab is displayed (Figure 15) 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.

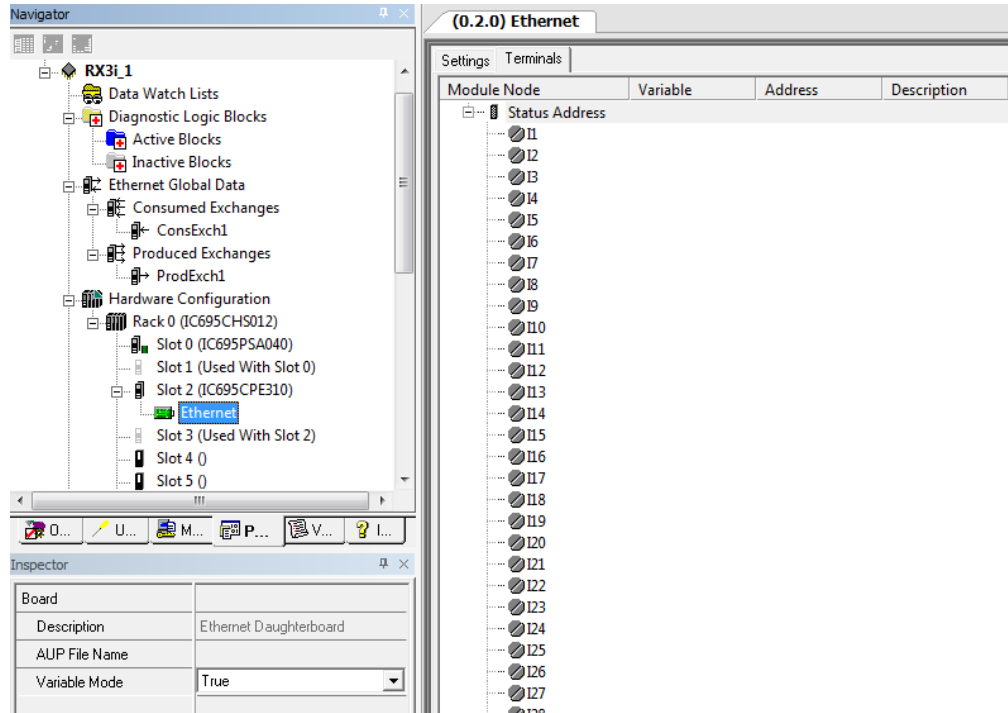


Figure 16: Terminals Tab Settings in Proficy Machine Edition

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 “I/O Variables” in the *PACSystems 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* (page 42) for a fuller discussion.

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 17:

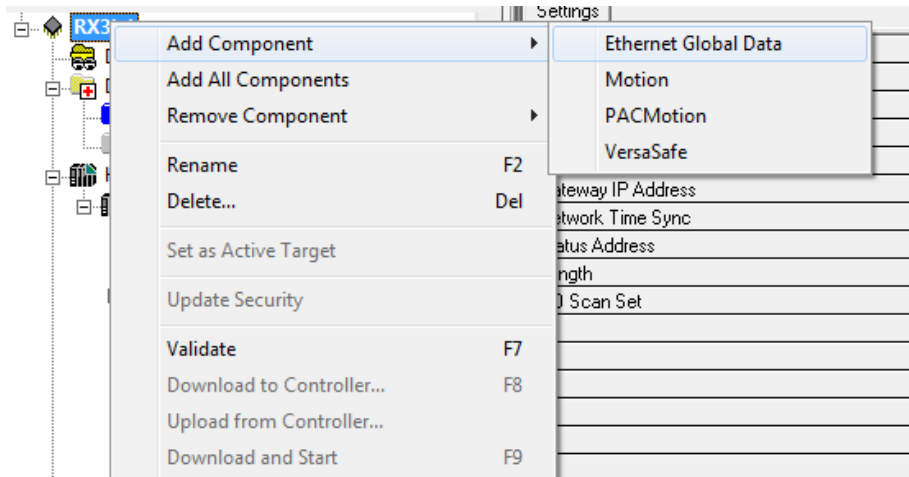


Figure 17: Adding Ethernet Global Data (EGD) to the Configuration

Once the EGD component has been added, it is possible to define the EGD data to be produced (Figure 18) and the EGD data to be consumed (Figure 19) by the embedded Ethernet Interface, per the following screen-shots.

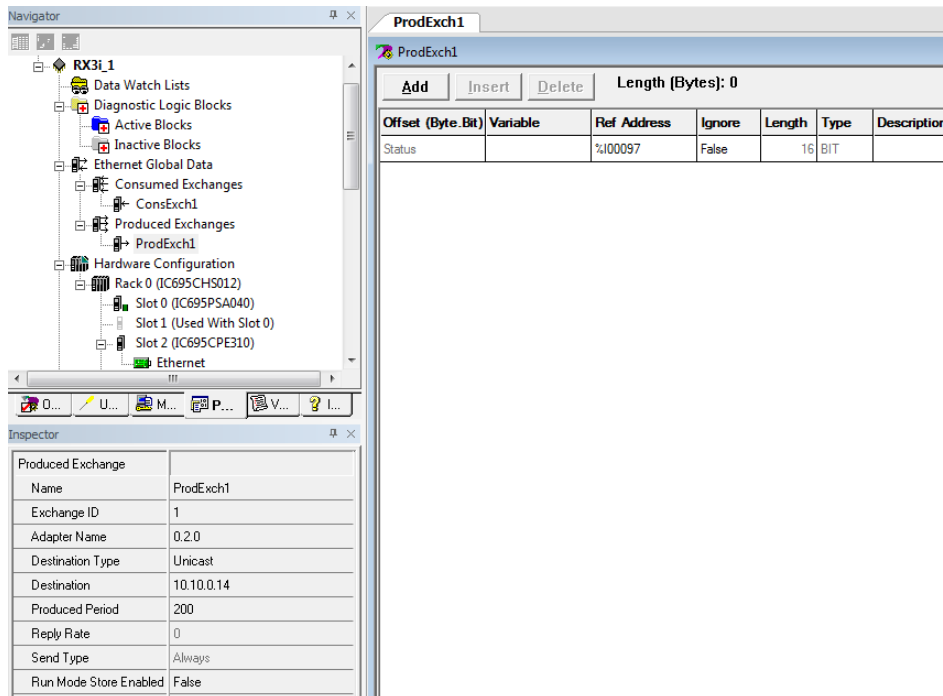
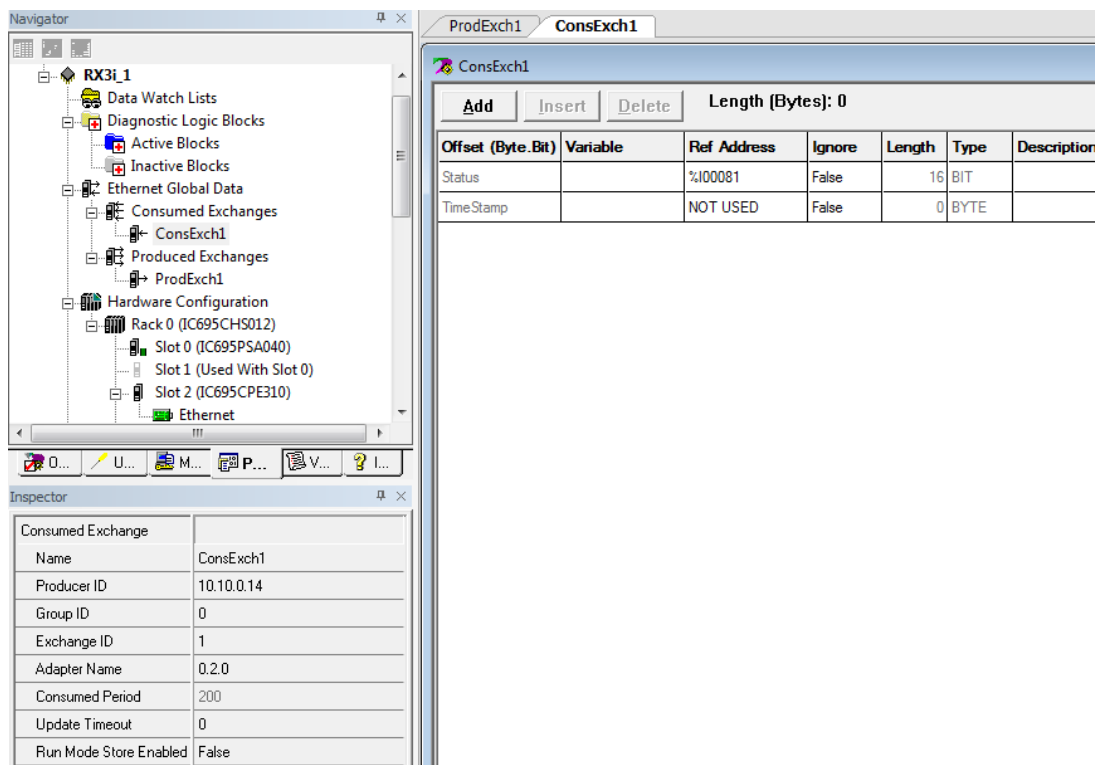


Figure 18: Defining EGD Produced Data Exchange



**Figure 19: Defining EGD Consumed Data Exchange**

The parameters to be entered and their relevance is discussed in the sections (below) entitled *Configuring an Ethernet Global Data Exchange for a Producer* (page 47) and *Configuring an Ethernet Global Data Exchange for a Consumer* (page 48).

See also *Ethernet Global Data Operation* in Chapter 5 for a fuller understanding.

## 4.2 Rack-based and RX7i Embedded Interfaces

The configuration process for the rack-based and RX7i embedded Ethernet interfaces includes:

- 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). Configuring advanced parameters. This requires creating a separate ASCII parameter file that is stored to the PLC with the hardware configuration. The Ethernet Interface has a set of default Advanced User Parameter values that should only be changed in exceptional circumstances by experienced users. The Advanced User Parameters definitions and configuration are described in Appendix A.
- (Optional) Setting up the PLC for Modbus/TCP Server operation. See Chapter 8 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 Machine Edition 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 if valid. If that data is invalid or has never been configured, factory default configuration values are used.

### **Locally Edited Configuration Data**

If the PLC configuration was not stored from the programmer, the CHSOSW and CHPARAM Station Manager commands can be used to locally edit Ethernet configuration or AUP configuration data. These Station Manager commands are not active if the PLC configuration has been stored from 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). Because this 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 using the "Set Temporary IP" method if the PLC is not in a RUN state, even if the Ethernet interface already has a valid configured IP Address. If the Ethernet interface 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.

Alternatively, an initial IP address can be set via the CHSOSW command from a local serially connected Station Manager terminal. See *PACSystems TCP/IP Communications Station Manager Manual*, GFK-2225, for details.

A third way of setting the IP address is to configure the IP address in Hardware Configuration and store the configuration over a serial connection.

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

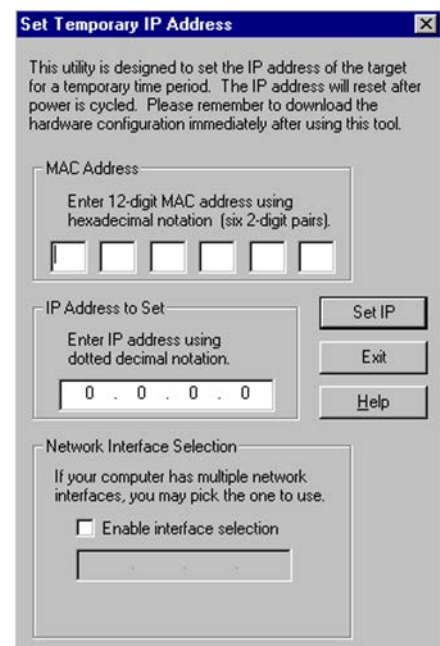
To initiate Ethernet communications with the programmer, you first need to set up a temporary IP address. After the programmer is connected, the actual IP address for the Ethernet interface (as set up in the hardware

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 if 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 address of the Ethernet Interface. The MAC address is located on a label on the module, as shown in Chapter 2, "Installation." Connect the PACSystems Ethernet Interface to the Ethernet network.

1. 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.
2. 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.
3. When the fields are properly configured, click the Set IP button.
4. 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 20: 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

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. The Ethernet Interface acts as a BOOTP client. The Ethernet Interface issues a BOOT Request to the network. If any 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 Chapter 2, 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.



### 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 Chapter 2, "Installation."
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.

3. 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.

## 4.2.3 Configuring Ethernet Interface Parameters

This section describes how to configure the parameters of an RX7i embedded or rack-based PACSystems Ethernet Interface.

### Configuring an RX7i 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 1). The Ethernet Interface daughterboard is displayed as "Ethernet".
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, then click in the appropriate Values field.

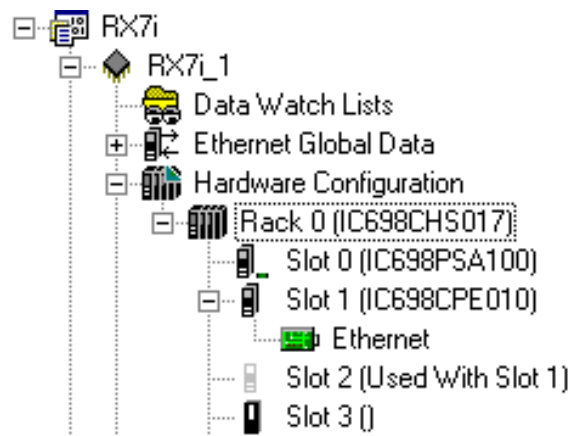


Figure 21: Expand RX7i CPU Node to Configure Ethernet Daughterboard



### Configuring a Rack-based Ethernet Interface Module

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 IC698ETM001 module (for RX7) or IC695ETM001 module (for RX3i) and click OK. The Ethernet module is placed in the rack and its parameters are displayed in the Parameter Editor window.

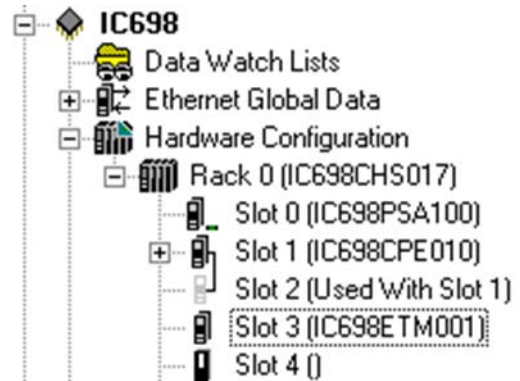


Figure 22: Install ETM001 Module in Rack/Slot & Expand to Configure

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, then click in the appropriate Values field. (To edit parameters of a module that is already configured in the rack, right click the slot containing the module and choose Configure.)

### Ethernet Parameters (Settings Tab)

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

**Adapter Name:** This is automatically generated based upon the rack/slot location of the Ethernet interface.

**Use BOOTP for IP Address:** This selection specifies whether the Ethernet must obtain its working IP address over the network via BOOTP. When set to False (= do not use BOOTP), the IP Address value must be configured (see IP Address parameter, below). 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 in order 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, otherwise the Ethernet Interface may be unable to communicate on the network and/or network operation 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.



**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. Valid range is 0 through 16. Default is 2.

**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. Valid range is 0 through 16. Default is 2.

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

**Network Time Sync:** Selection of the method used to synchronize the real-time clocks over the network. The choices are None (for no network time synchronization) and 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. Time stamps of exchanges produced by a PLC with this setting are not synchronized with the time stamps of exchanges produced by other PLCs.

See “Time-stamping of Ethernet Global Data Exchanges” in Chapter 5 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 Chapter 12, “Diagnostics,” for definitions of the LAN Interface Status (LIS) portion of the Ethernet Status data.

The meaning of the Channel Status portion of the Ethernet Status depends upon the type of operation for each channel.

For details of the status bits and their operation, refer to “Monitoring the Ethernet Interface Status Bits” in Chapter 12, “Diagnostics.”

**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 Terminals Tab, below).

**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:** 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, below. The default value is False.

**Redundant IP Address:** 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 (above) 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 Chapter 1, “Introduction” for more information about Ethernet redundancy. This IP address is assigned in addition to the device’s primary IP address.

**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.

**Note:** The Ethernet interface delivers its Ethernet Status (including Channel Status bits) during its input scan. Each channels data transfer updates the Channels Status bits, so channels performance may be reduced if the Ethernet interface is configured to use an I/O Scan Set than 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. These 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

This configuration tab is displayed only when the Ethernet interface's Variable Mode property is set to True. When Variable Mode is selected, 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 "I/O Variables" in the *PACSystems CPU Reference Manual*, GFK-2222.

## 4.2.4 Configuring Ethernet Global Data

For more information about Ethernet Global Data, see Chapter 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 Manual*, GFK-2439.

### Basic EGD Configuration

Whether or not the EGD Configuration Server is used, certain steps will need to be taken to use EGD. These steps are described below.

If Ethernet Global Data does not appear as shown, right-click the PLC icon (**PLC1** in this example). Select 'Add Component' and then select 'Ethernet Global Data'.

For each PLC:

1. In the PLC programming software, open the Project folder and expand the target node for the PLC.

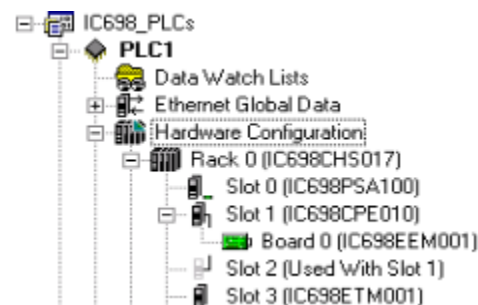


Figure 23: Expand Node to View Ethernet Global Data

- To configure the Local Producer ID, right-click the Ethernet Global Data node and choose Properties. The Local Producer ID is shown in the properties Inspector window. This parameter must be unique on the network.

The **Local Producer ID** 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.

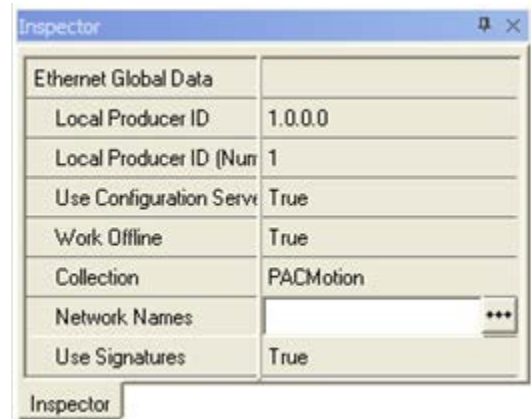


Figure 24: Local Producer ID

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 Chapter 5, "Ethernet Global Data," for more information on how the Producer ID is used.

### 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. The use of the offset is illustrated below.

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.

**Note:** It is the user's responsibility to ensure that the same offset value is specified in both the primary and secondary target projects.

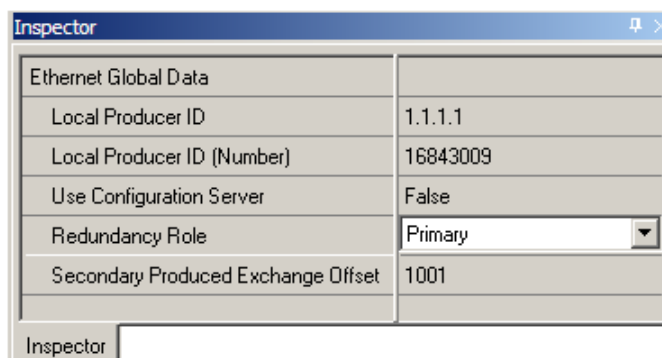
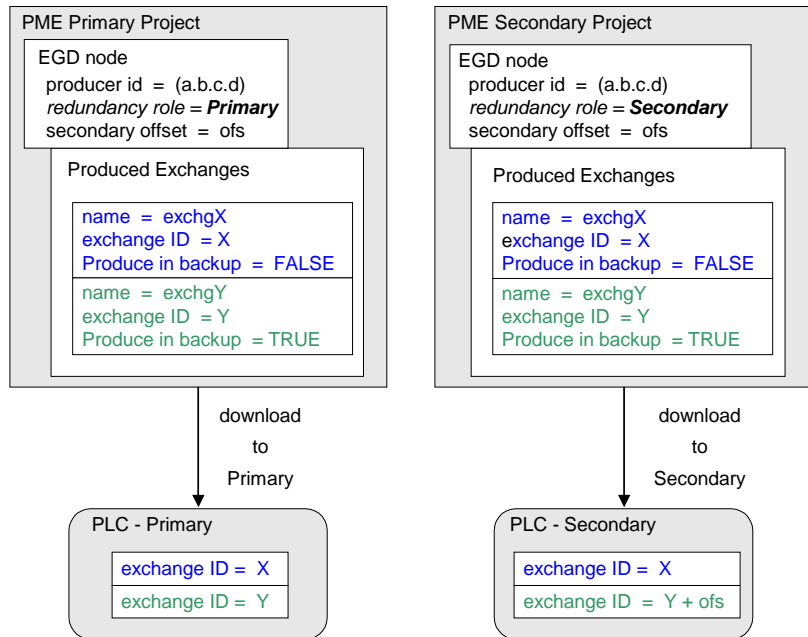


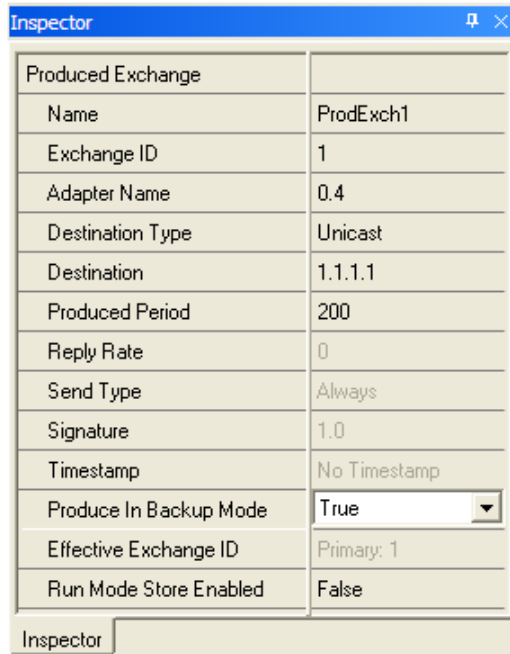
Figure 25: Configuring Redundancy for Ethernet Global Data

**Exchange ID Offset in an Ethernet Redundancy System**



**Figure 26: Exchange ID Offset in an Ethernet Redundancy System**

The “Produce in Backup Mode” parameter appears in the properties for each produced exchange.



**Figure 27: Configuring Produce in Backup Mode Parameter**

**Using Signatures in Ethernet Global Data**

EGD signatures can be used to make sure that the format of the data from the producer matches that expected by the consumer.

The EGD signature is a numeric value that has two parts: the major number and the minor number. The major number reflects the “primary format” of the data. The minor number reflects backward-compatible changes

made to the EGD exchange (such as adding data to the end of the exchange). An EGD signature has the format *maj.min*, where *maj* is the major value and *min* is the minor value.

The primary format of the data is first established when the EGD exchange is defined. At that time 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. 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.

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. After checking the signature major number, the consumer checks the signature minor number. 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 can 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.

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.

Only the PACSystems RX7i and RX3i support non-zero signatures. All other targets force the signature for both produced and consumed exchanges to be zero.

Use of signatures is enabled by default for new RX7i or RX3i projects and is disabled for other targets and for existing projects.

### **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 0 because a value of 0 effectively disables the signature checking function.

For information about the use of signatures with run mode stores of EGD, refer to “Run Mode Store of EGD” in Chapter 5.

### **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 that both the producer and consumer must have signatures enabled, otherwise signatures are ignored and only the exchange size is used to determine compatibility.

### **Configuring Ethernet Global Data Using the EGD Configuration Server**

The EGD Configuration Server is supplied with the Machine Edition software, but it is not automatically installed with Machine Edition. 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, go to the directory where the machine Edition software is installed and open the folder named “EGD Installs”. Select the file “EgdCfgServerSetup.msi”. Double-click on the file to install the EGD Configuration Server.

#### **Configuring the EGD Configuration Server**

To configure the Ethernet Global Data server in Machine Edition, click on the Options tab in the Navigator window. In the Machine Edition folder, select the EGD item to display the configuration options for the configuration server. For example:

EGD Preferences	
Local Server Cache Path	C:\Program Files\COMPLIXITY Machine Editor
Base Path	/EGD
Host Name	172.131.1.25
Server Port	7938
Timeout	20000
Configuration Server	Located

**Figure 28: Configuring the EGD Configuration Server**

**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.

**Host Name** : The host name 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 typically is left at the default of 7938. If changed, it must be changed on both the programming software and on the server. This value is not stored in the project but is stored in the computer. 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 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. The Use Configuration Server option is displayed in the properties Inspector window. This parameter may be set to True to enable using the configuration server for the device or to False to not use the server.

In some cases you may want to work offline from the configuration server for a while, for example when you want to work disconnected from the network and the configuration server is located on another computer. In this case, you can 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. Setting this path to a location on the local machine and selecting 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 will synchronize the server with the data from the cache.

#### **Network Names and Collections**

In order to perform validation between producers and consumers, it is necessary to know 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.

### 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 Chapter 12.

The EGD Management Tool can look at subsets of EGD devices, called collections. 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 assigned for this exchange. Defaults to "ProdExchX" where X is a sequential number.
<b>Exchange ID</b>	A number that identifies a specific exchange to be sent by the producing device.
<b>Adapter Name</b>	The specific Ethernet Interface, identified by its rack and slot location within the producing 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). 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>	Identifies the data's consuming device, based on the Destination Type selected above: <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 0 or 2-3,600,000 (2 milliseconds to 1 hour). The value zero means data will be produced at the end of each PLC scan, but not less than 2 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 2 milliseconds.
<b>Reply Rate</b>	Not used.
<b>Send Type</b>	Fixed at "always." 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>Run Mode Store Enabled</b>	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.
-------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Configuring the Exchange Variables

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 to 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 <b>Byte.Bit</b> , 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 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 that 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 data type.
<b>Type</b>	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, 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 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." 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 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 assigned for this exchange. Defaults to the target name of the producer, an underscore, and the exchange ID in the producer. Changing this name may make resynchronization of the variable with the server impossible.
<b>Producer ID</b>	The ID of the PLC producing the exchange. Producer ID is defined by the producer; changing here it may make resynchronization with the server impossible.
<b>Group ID</b>	Used only if the produced exchange has been configured with a Destination Type of Multicast. Group ID is defined by the producer; changing it here may make it impossible to consume the data from the producer.
<b>Exchange ID</b>	Identifies a specific data exchange to be received by the consuming device. Exchange ID is defined by the producer; changing it here may make resynchronization with the server impossible.
<b>Adapter Name</b>	The specific Ethernet Interface, identified by its rack and slot location within the consuming PLC.
<b>Consumed Period</b>	Not used. (Always displayed as 200 milliseconds; not editable.)
<b>Update Timeout</b>	A value in the range 0 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 producer period, and should allow for transient network delays. The default is 0 indicates no timeout. Resolution is in 2ms increments.
<b>Run Mode Store Enabled</b>	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.

### 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 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 in 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 and any discrepancies are noted (see above for how to correct any errors detected in the consumer).

It is also possible, by selecting the menu item “Unconsumed Data Report”, to generate a report listing 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; 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. However, 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 separate frame. It allows you to visualize, analyze and debug an EGD network. See Chapter 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 not automatically installed when you install the Programmer. To install the GDE, 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 “EgdGenericEditorSetup.msi”. Double-click on this file to install the EGD Generic Device Editor.

### **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.

### **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.
2. 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 assigned for this exchange. Defaults to “ProdExchX” where X is a sequential number.
<b>Exchange ID</b>	A number that identifies a specific exchange to be sent by the producing device.
<b>Adapter Name</b>	The specific Ethernet Interface, identified by its rack and slot location within the producing 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 0 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 2 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 to the nearest 2 milliseconds.
<b>Send Type</b>	Fixed at “always.” 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, 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.

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 to 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 <b>Byte.Bit</b> , 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 defined for this variable.
<b>Ref Address</b>	The PLC memory reference address that 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>	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 assigned for this exchange. Defaults to "ConsExchX" where X is a sequential number.
<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 specific Ethernet Interface, 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 0 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 producer period, and should allow for transient network delays. The default is 0 indicates no timeout. Resolution is in 2ms increments.
<b>Run Mode Store Enabled</b>	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.

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 <b>Byte.Bit</b> , 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 defined for this variable.
<b>Ref Address</b>	The PLC memory reference address that 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 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>	Size of the data for this variable, expressed in units of the selected PLC reference memory type.
<b>Type</b>	Data type of the selected PLC reference memory type. (Automatically setup 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 2-byte integer, followed by a 2-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 that 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 being transferred for the exchange.





## Chapter 5 Ethernet Global Data

---

This chapter describes basic Ethernet Global Data (EGD) features, which are supported on all RX7i Ethernet interfaces and by the rack-based RX3i Ethernet interface (ETM001). Effective with RX3i CPE310/CPE305 Firmware Release 8.30, the RX3i CPU itself also supports EGD Class 1<sup>3</sup>. 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 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.

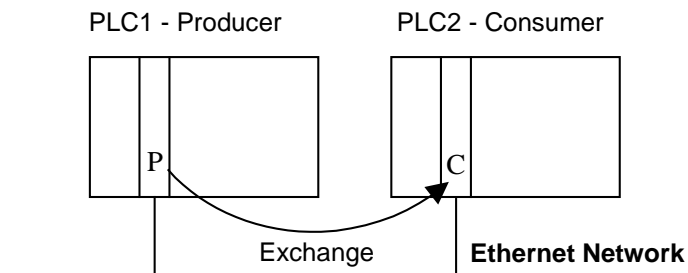


Figure 29: Producing & Consuming Ethernet Global Data

### 5.1.1 EGD Producer

The producer of an exchange periodically sends new samples of data from its local internal memory. The producer of an 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 particular PLC on the network. Since the Producer ID identifies only the PLC producing the exchange, it doesn't matter how many Ethernet Interfaces are installed in that 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.

### 5.1.2 EGD Consumers

A consumer is a device that will update its local internal memory based on the data in an 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. The consumer takes in the whole exchange from the network but does not need to send all of the exchange to the PLC memory. This feature is called Selective Consumption. A Consumed Exchange can be set to ignore the data ranges (variables) that are not needed.

## 5.2 EGD Exchanges

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.

Different produced exchanges can include some or all of the same data even though the exchanges are produced at different rates and 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 the concept of an EGD “page”. 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. Machine Edition does not currently show information about EGD pages; you will instead see the constituent exchanges for each page.

### 5.2.1 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:

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

The same exchange can be configured at each consumer to suit the needs of the application.

### 5.2.2 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 of 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 of the same data ranges even if the exchanges are produced at different rates. A consumer does not have to consume all of the data from a produced exchange. A consumed exchange may be configured to ignore specified data ranges. (See “Selective Consumption” in Chapter 4.)

### 5.2.3 Valid Memory Types for Ethernet Global Data

The PLC memory types listed below can be included in EGD exchanges.

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

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.

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.

Add   Insert   Delete   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

Figure 30: Adding Symbolic Reference to Ethernet Global Data Exchange

### 5.2.4 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 of 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.

### 5.2.5 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 interfaces changes to backup, it stops 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. This permits 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 to “Ethernet Redundancy Operation” in Chapter 1.

## 5.3 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 of 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 Broadcast or Multicast, the same exchange must be configured at the producer and at each consumer. Each consumer can use all of the data or just a selected portion, as configured for the consumed exchanges.

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

### 5.3.1 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 from Group 1.

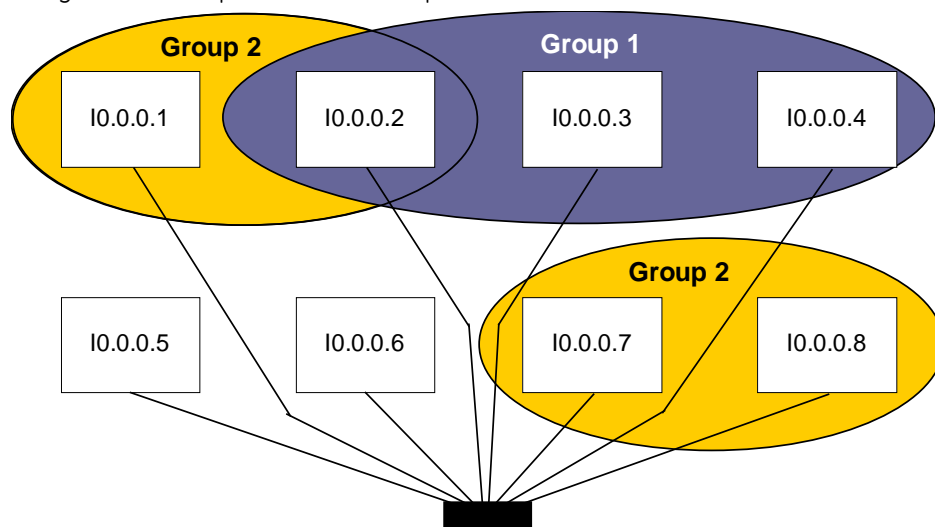


Figure 31: Grouping of Devices for Ethernet Global Data Multicasting

Each device in a group responds to the group’s assigned ID number from 1 to 32.

**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
1	224.0.7.1
2	224.0.7.2
.	.
.	.
.	.
32	224.0.7.32

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.

### 5.3.2 Broadcasting Ethernet Global Data

The same Ethernet Global Data exchange can be sent to all of 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.

### 5.3.3 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 have 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.

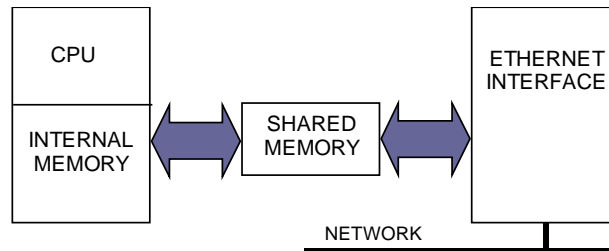
This means that modifying a multicast exchange so that it produces to a different Group ID may or may not affect its consumption. If the remote device has any exchanges configured to consume from the new producer ID, consumption will not be interrupted. However, consumption will be affected if the remote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

## 5.4 Ethernet Global Data Timing

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



**Figure 32: 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 particular exchange; it has 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.

### 5.4.1 EGD Synchronization

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

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 on 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.)

As soon as a sample for a consumed exchange is received, it is transferred to the CPU during the next input scan portion of the CPU sweep.

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 not useful or necessary to configure the production period to be less than the CPU sweep time. 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 fresh CPU data becomes available at the end of the sweep, the Ethernet interface will immediately send another sample with the fresh 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.

### 5.4.2 Configurable Producer Period for an EGD Exchange

The Producer period for an EGD exchange can be 2 milliseconds to one hour. In the PLC, the Ethernet Interface attempts to produce the data at this interval. As explained above, the exchange production may vary from the configured interval by up to one production period or one producer CPU sweep period, whichever is smaller.

Producer period is configurable in increments of 2 milliseconds. If the Producer Period is set to zero, production is scheduled every scan or every 2ms, 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 being unable to produce all the required data, and will also degrade SRTP communication.

### 5.4.3 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 on the devices, providing capacity for other transfers.

The following illustrations show the relationship among 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.

#### Producer Period = 1.5 Times CPU Sweep

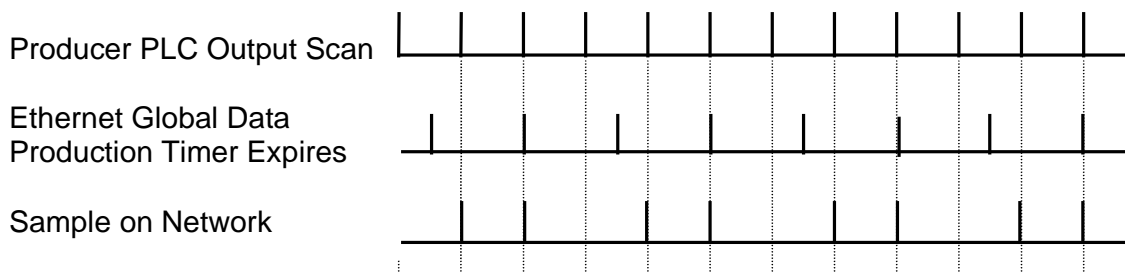


Figure 33: EGB Timing Example #1



**Timing Example 2**

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

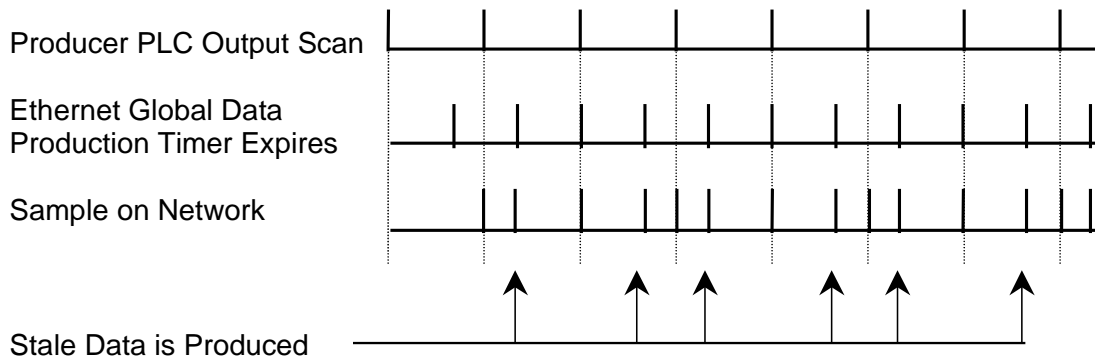
**Producer Period = 2/3 Time of CPU Sweep**

Figure 34: EGB Timing Example #2

## 5.5 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.

In its default operating mode for SNTP synchronization, the PLC CPU obtains the timestamp data from the time clock in the Ethernet interface, which can be synchronized to either the clock in the CPU or an external SNTP server on the network. For details on SNTP operation, see page 75.

Alternatively, the timestamp data can be obtained from the CPU TOD clock when the CPU TOD clock is synchronized with an SNTP server. Synchronizing the CPU TOD clock to an SNTP server allows you to set a consistent PLC time across multiple systems. This operating mode must be configured by an Advanced User Parameter and enabled from the application logic. For additional information, see "Obtaining Timestamps from the CPU TOD Clock" on page 67.

### 5.5.1 Obtaining Timestamps from the Ethernet Interface Clock

In this operating mode, 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.

If time synchronization between the CPU and ETM is lost, as when the CHTIME Station Manager command is used to change the ETM time, the CPU uses its own clock for the time stamp.

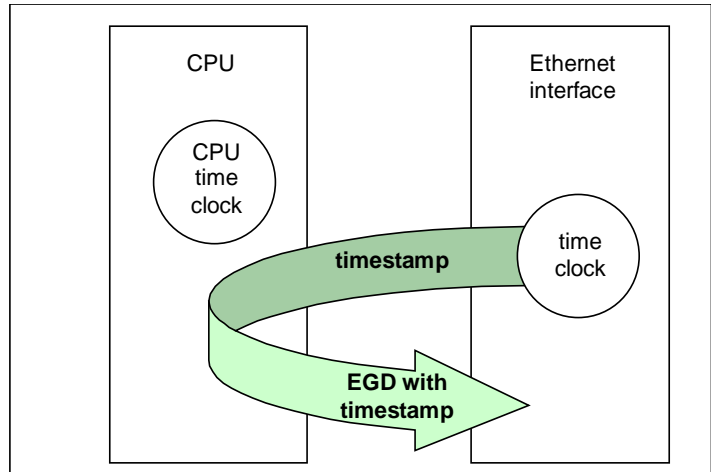


Figure 35: Obtaining Timestamps from the Ethernet Interface Clock

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 at 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.

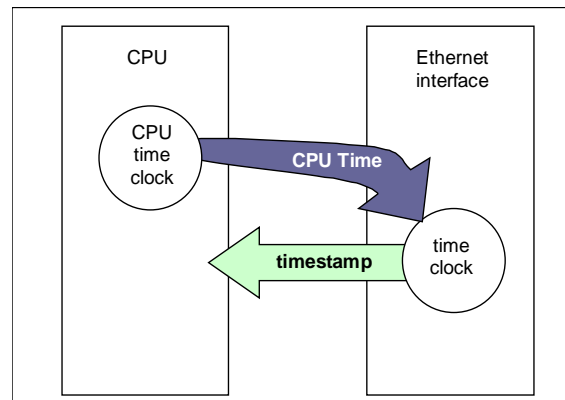


Figure 36: Obtaining Timestamps from the PLC Time Clock

**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.

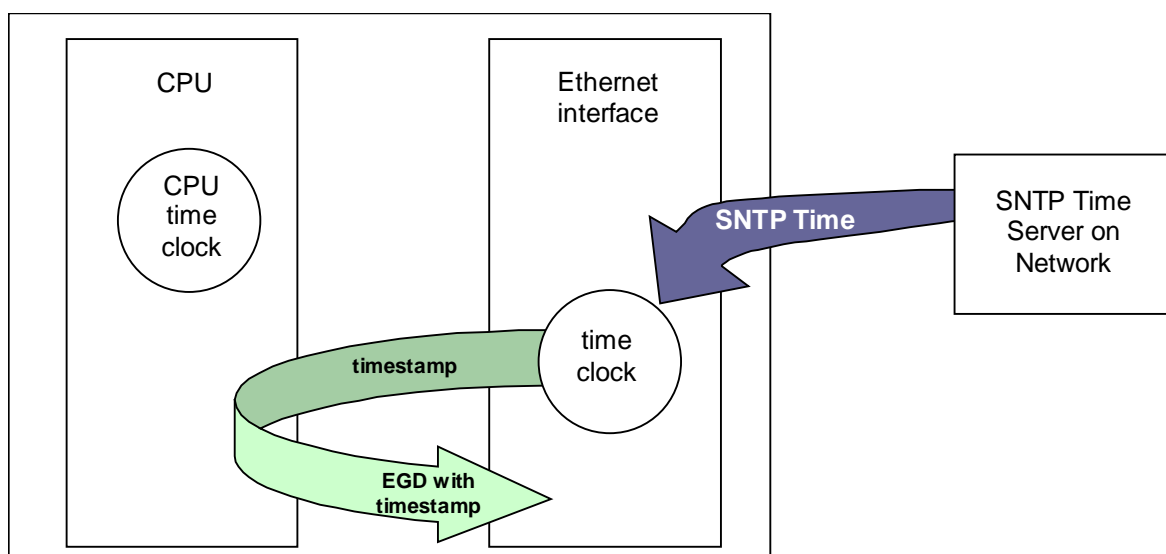


Figure 37: Obtaining Timestamps from the SNTP Server's Time Clock

### 5.5.2 Obtaining Timestamps from Embedded Ethernet Interface for RX3i CPE305/CPE310

In this operating mode, the PLC CPU obtains the timestamp data from its own Time-of-Day (TOD) Clock. The CPU uses this timestamp for Ethernet Global Data exchanges produced by its Embedded Ethernet Interface.

The following two scenarios exist, depending on the configuration chosen for the Time Synchronization feature:

1. **PLC's Time Clock:** If this source is selected (Time synchronization = None) in PME configuration, then the CPU Time Clock is not synchronized to the SNTP time server, and Ethernet Global Data exchanges produced by the Embedded Ethernet Interface are time-stamped using the CPU Time Clock.
2. **SNTP Server's Time Clock:** If this source is selected (Time synchronization = SNTP) in PME configuration, the CPU TOD clock will be periodically synchronized to the reference clock on SNTP server over the Ethernet network and via the Embedded Ethernet Interface.

In the event the RX3i CPU is hosting one or more ETM001 modules in its backplane, scenario (2) above may have the following two sub-scenarios:

- a. If the installed ETM001 module has Time synchronization configured to "None", the RX3i CPU will also use the SNTP Server timestamp for the Ethernet Global Data exchanges produced by that ETM001 module. This will enable this CPU system to apply a consistent timestamp to all produced EGD exchanges, regardless of the physical source within that CPU system.
- b. If the ETM001 has Time synchronization configured to "SNTP" and is actively synchronized to the reference clock of the SNTP server on network, then the EGD exchanges produced by that particular ETM001 will use the timestamp generated by that particular ETM001 clock. This scenario can result in more variance of the applied timestamp, especially if communications with the SNTP Server is lost for a prolonged period.

Note that each ETM001 discussed in (a) and (b) above is individually configured.

### 5.5.3 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, all produced EGD exchanges will use the CPU's TOD for the time stamp.

Synchronizing the CPU TOD clock to a network timeserver requires CPU firmware version 5.00 or greater. Each participating Ethernet interface must use firmware version 5.00 or greater. Older firmware versions do not support the necessary COMMREQ commands.

### Synchronizing the CPU TOD Clock to an SNTP Server

The CPU TOD clock is set with accuracy within  $\pm 2\text{ms}$  of the SNTP time stamp.

CPU TOD clock synchronization is enabled on an Ethernet module by setting the Advanced User Parameter (AUP) `ncpu_sync` to 1. For details on configuring an AUP file, refer to Appendix A.

Within a PLC, only one Ethernet interface at a time 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 69.

**Note:** With the AUP parameter `ncpu_sync = 1`, the Ethernet modules get their time from the SNTP network server regardless of the Network Time Sync setting in the Ethernet module's hardware configuration.

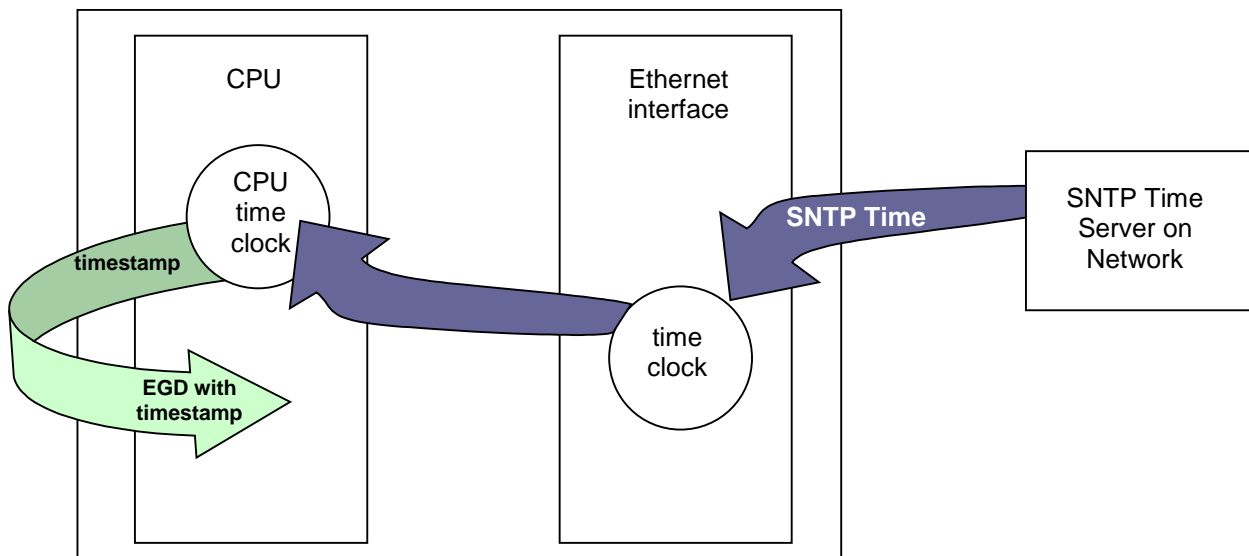


Figure 38: Synchronizing CPU Time-of-Day Clock to an SNTP Server

### Operating Sequence for CPU Clock Synchronization

The following diagram illustrates the sequence of events for setup and operation of a system that uses clock synchronization.

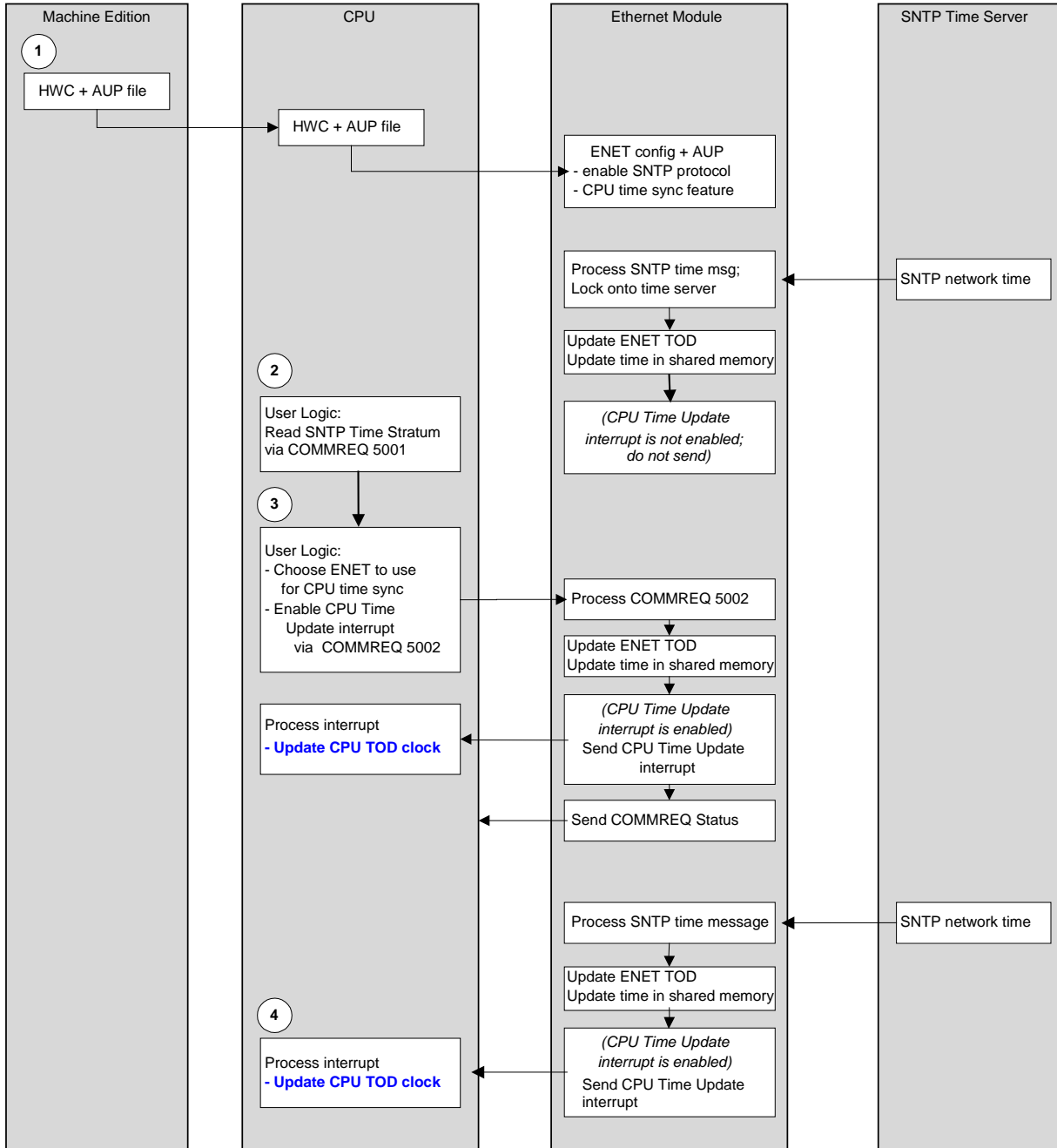


Figure 39: Operating Sequence for CPU Clock Synchronization

### Steps to Synchronize the CPU TOD Clock to an SNTP Server

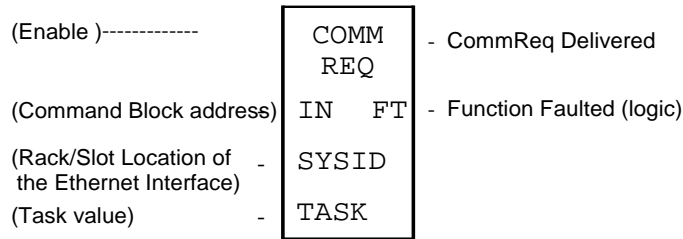
These steps correspond to the numbers in the operating sequence illustrated on page 69.

1. The user configures an AUP file to enable the CPU Time Sync feature and imports AUP file(s) into the PLC configuration. The user stores HWC containing AUP file(s) to PLC.
2. 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.
3. 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 timeserver on the network, the CPU immediately updates its TOD clock.
4. At every subsequent periodic network time message from the locked SNTP timeserver, the CPU receives the network time and immediately updates its TOD clock.

**Note:** In a PLC with only one Ethernet interface, the logic program may skip step 2. There is no need to select between multiple Ethernet interfaces.

### 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 40: COMMREQ to Control the CPU Time-of-Day Clock**

The parameters of the COMMREQ are:

**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.

Rack	Slot	Hex Word Value
0	4	0004H
3	4	0304H
2	9	0209H
4	2	0402H

**TASK:** For the PACSystems Ethernet module, 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.



### Caution

Entering an incorrect TASK value may cause the Ethernet Interface to fail.

**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.

#### 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.

#### Command Block for Read Ethernet Clock Status and Stratum COMMREQ

Word Offset	Value	Description
Word 1	Length of 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 "COMMREQ Status for the EGD Commands" in Chapter 6.	Memory type of the COMMREQ status word.
Word 4	0-based	Offset of COMMREQ status word. For CRS word values, refer to page 74.
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 "COMMREQ Status for the EGD Commands" in Chapter 6.	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

The Ethernet clock status and stratum values from the locked time server (if any) are returned as two consecutive words.

**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 timeserver, the Ethernet clock stratum value indicates the accuracy of the time value provided by the server. A stratum value of 1 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**

<b>Value</b>	<b>Description</b>
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 timeserver on the network. The logic application can periodically examine this bit to determine when an Ethernet module has lost its lock with a network timeserver. For details of the LIS block, refer to “Monitoring the Ethernet Interface Status Bits” in Chapter 12.

**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**

<b>Word Offset</b>	<b>Value</b>	<b>Description</b>
Word 1	Length of 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 "COMMREQ Status for the EGD Commands" in Chapter 6.	Memory type of the COMMREQ status word.
Word 4	0-based	Offset of COMMREQ status word. For CRS word values, refer to page 74.
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 “Communication Request” in the *PACSystems 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.

<b>Minor (Hex)</b>	<b>Major (Hex)</b>	<b>Description</b>
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 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 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 not allowed because SNTP Time Transfer feature was not configured.
16	0D	COMMREQ data block length (COMMREQ word 1) is too long.

### **CPE305/310 TOD Clock Synchronization with Time from ETM001 Modules**

Whenever one or more ETM001 modules are mounted in the rack of a host CPU, the CPU TOD clock can be synchronized to the time clock within a selected ETM001 module. To synchronize in this fashion, use AUP parameter (ncpu\_sync=1) and time synchronization COMMREQs (5001 and 5002).

1. CPU TOD clock synchronization - This is enabled on an Ethernet module by setting the Advanced User Parameter (AUP) ncpu\_sync to 1. When applied to an RX3i CPU, the effect is to cause the embedded Ethernet interface within the CPU to look to the specified ETM001 module for time synchronization.
2. Read Ethernet Clock Status and Stratum COMMREQ (5001) for Embedded Ethernet Interface. The effect is to read this information from the targeted ETM001 module.
3. Enable or Disable PLC Time Update COMMREQ (5002) for Embedded Ethernet Interface. Enabling causes the embedded Ethernet interface within the CPU to synchronize its TOD clock to the clock within the targeted ETM001 module.

There are some complications that should be avoided. For example, if the CPU TOD Clock is synchronized over the backplane to the time clock of an ETM001 module, as discussed above, and the CPU Time synchronization is also set to 'SNTP', then this will result in the produced Ethernet Global data exchanges from that PLC system having alternate timestamps cyclically. In this scenario, the CPU TOD clock will alternately be set using the SNTP time (a) from its Embedded Ethernet Interface and (b) from the designated ETM001 module. This is not desirable.

#### **5.5.4 SNTP Operation**

In an SNTP system, a computer on the network (called an SNTP server) sends out a periodic timing message to all of the SNTP-capable Ethernet Interfaces on the network, which keep their internal clocks synchronized with this SNTP timing message.

In a redundancy system, SNTP operation is unaffected by the current Ethernet redundancy state or by redundancy role switches.

SNTP server dates before January 1, 1989 are not supported.

##### **Normal SNTP Operation**

If SNTP is configured, the default mode of operation is Broadcast and Multicast. For Unicast mode of communication, you will need to configure the necessary parameters as defined in Appendix A, "Configuring Advanced User Parameters."

##### **SNTP Broadcast and Multicast Operation Mode**

The Ethernet Interface will synchronize to a remote SNTP timeserver after receiving two broadcast clock values within a 150-second period. The Station Manager can be used to view server status information.

##### **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. For parameter definitions refer to Appendix A, "Configuring Advanced User Parameters."

##### **Multiple SNTP Servers (Applies only to SNTP Broadcast and Multicast Mode)**

To guard against loss of SNTP timing messages, multiple SNTP timeservers 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 a 150-second period. A server is considered "lost" if more than 150 seconds elapse between timing messages.

### **Local Time and Daylight Saving Time Corrections**

Versions 6.20 and later of the Ethernet interface support the ability to specify an offset to the Coordinated Universal Time (UTC) to correct for local time zone and daylight saving 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. The specified correction is applied to all modes of SNTP communications (Broadcast, Multicast and Unicast).

The default SNTP operation is *no correction* for local time or DST. Local time and DST corrections must be enabled via AUP. For local time correction and DST parameters, refer to Appendix A, "Configuring Advanced User Parameters."

### **Loss or Absence of SNTP Timing Signals**

If an Ethernet Interface is configured for SNTP, but does not receive two timing messages from an SNTP network timeserver within a 150-second period, 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 particular 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 the Ethernet Interface, after its last power-up/restart cycle, has never received an SNTP server's timing message, it will continue to use the PLC CPU's local clock value that it received at power-up/restart for its time base.
- If the Ethernet Interface has been synchronized to an SNTP server but 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 time-stamping, while it "listens" for SNTP timing messages from the network. If SNTP messages are received later, the Ethernet Interface will then synchronize to them.

## **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 stop. 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 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>7</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>8</sup>	<sup>8</sup>
STOP-Clear Logic	YES	YES	<sup>8</sup>	<sup>8</sup>
STOP-Config Store	Replaced <sup>9</sup>	Replaced <sup>7</sup>	NO <sup>9</sup>	NO <sup>9</sup>
STOP-Clear Config	NO	NO	NO	NO
PLC Power Cycle	YES	YES	<sup>8,10</sup>	<sup>8,10</sup>
Ethernet Interface Restart	YES	YES	<sup>8,10</sup>	<sup>8,10</sup>

### 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.

PACSystems versions 5.5 and later allow you to 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 begin consumption/production shortly after the activation of any logic that is part of the run mode store sequence.

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

**Modified** exchanges will be offline for a short time during the activation of new logic that is part of the RMS. This amount of time depends on factors, such as sweep mode and sweep time. All variables associated with a

<sup>7</sup> RUN-SUSPEND I/O refers to the SUSIO logic function. (The DOIO logic function does not affect EGD production or consumption.)

<sup>8</sup> Production and consumption is controlled by the PLC Mode as described above.

<sup>9</sup> Producer ID and exchange definitions are replaced.

<sup>10</sup> Producer ID and exchange states depend on the PLC mode and configuration prior to the action.

modified exchange will hold their last state during the pause in consumption. The consumption timeout is restarted for each modified consumed exchange.

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 alter the operation of an existing exchange and do not affect compatibility.

For details on the use of signatures to determine compatibility, refer to "Using Signatures in Ethernet Global Data" in Chapter 5.

### **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.

<b>Parameter</b>	<b>Behavior</b>
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.

<b>Parameter</b>	<b>Behavior</b>
Producer ID	Redefines the exchange. Causes a major signature change in the producer. Exchange will be incompatible.
Group ID	Determines the producer of the exchange and may affect compatibility. For details, refer to "Sending an Ethernet Global Data Exchange to Multiple Consumers" on page 61.
Exchange ID	Redefines the exchange. Causes a major signature change in the producer. Exchange will be incompatible.
Adapter Name	Deletes an exchange from one Ethernet module and adds an exchange to another. Assuming no other parameters change, this will not affect compatibility 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.
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**

<b>Parameter</b>	<b>Behavior</b>
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 "Sending an Ethernet Global Data Exchange to Multiple Consumers" on page 61.
Destination	Determines the consumer(s) of the exchange. Affects compatibility.
Produced Period	Modifies the existing exchange. Does not affect compatibility.
Produce In Backup Mode	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. If the primary unit is the active unit, modifying this parameter will have no immediate effect. If the secondary unit is the active unit, modifying this parameter will cause an incompatibility because it changes the exchange ID. <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.

### 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.

<i>Type of Change</i>	<i>Resulting Signature Change</i>
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 an RMS to a producer or consumer to cause an incompatibility that cannot be detected by the 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 is 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 an RMS 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 RMS, 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, this effectively disables signature checking.

## **5.7 Monitoring Ethernet Global Data Exchange Status**

The Exchange Status word is used to store status information about an EGD exchange. A unique Exchange Status word location must be configured for each 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.

The program should also monitor the "LAN Interface OK" Status bit (see Chapter 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 in order to identify the exchange. In this case, the Exchange Status Word cannot be updated.

### 5.7.1 Exchange Status Word Error Codes

The following table shows the error codes that can be written to the Exchange Status word in the Producer (P) and Consumer. The Exchange Status Word value for each exchange may be displayed via the STAT G Station Manager command.

<b>Value (Dec.)</b>	<b>P/C</b>	<b>Error</b>	<b>Description</b>
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>11</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>11</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>11</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.

## Chapter 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:

- Read PLC Memory
- Write PLC Memory
- Read EGD Exchange
- Write EGD Exchange
- Masked Write to EGD Exchange

### 6.1 General Use of EGD Commands

COMMREQ-driven EGD Commands can be used in the application program to read and write data into PACSystems PLCs or other EGD Class 2 devices.

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 Redundant IP operation (see Chapter 1, "Introduction", for more information), only the active unit sends or responds to EGD commands. 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 recommend, EGD commands may be issued to the direct IP address. Both 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.

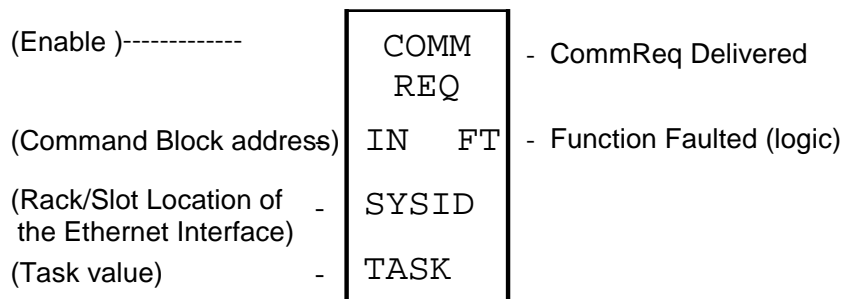


Figure 41: COMMREQ Used to Program Ethernet Global Data

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

**Enable:** Control logic for activating the COMMREQ Function Block.

**IN:** 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.

**SYSID:** 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.

**TASK:** 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, Task must be set to zero.

**FT Output:** 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 Chapter 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 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 previous page)	Memory type of COMMREQ Status Word
Word 4	0-based <sup>12</sup>	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)	
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)	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. May not be a group IP address.
Word 22	Reserved	Always 0

<sup>12</sup> Word 4 (COMMREQ status word address) is the only zero-based address in the Command Block. This value alone requires that 1 be subtracted from the intended address.

**(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. 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>13</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>14</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)

**(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. (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 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.

<sup>13</sup> %W memory is supported on PACSystems clients and servers only.

<sup>14</sup> Read-only memory, cannot be written to.

**(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 (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. For example, if Word 12=16 and Word 15=4, then 4 bytes (32 bits) of %I memory will be transferred. 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.

## 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 84)	Memory type of COMMREQ Status Word
Word 4	0-based <sup>12</sup>	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)	
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

**(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).

**(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 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. For example, if Word 12=16 and Word 15=4, then 4 bytes (32 bits) of %I memory will be transferred. For Write 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.

## 6.7 Read EGD Exchange (4002)

The Read EGD Exchange command can be used to read some or all of a configured Ethernet Global Data exchange from either the producer or the consumer. 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 to 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 84)	Memory type of COMMREQ Status Word
Word 4	0-based <sup>12</sup>	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
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 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	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

**(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 Data Length (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 12) Remote EGD exchange – Exchange Signature:** Word 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 be executed only 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.

<b>EGD Signatures Enabled (Y/N)</b>	<b>Signature Comparison Desired</b>	<b>Recommended with RMS of EGD</b>	<b>User Specified Signature</b>
No	No	No	0 (Default - no check)
Yes	No	No	0 (Default - no check)
Yes	Yes	Yes	Current EGD signature

**(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.

## 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 to 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 84)	Memory type of COMMREQ Status Word
Word 4	0-based <sup>12</sup>	Offset of COMMREQ Status Word
Word 5	0	Reserved
Word 6	0	Reserved
Word 7	4003 (fa3H)	Write 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)	
Word 12	Remote signature	EGD Exchange signature. This should be 0 for PLCs when not using signatures. If 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	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

**(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 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.

<b>EGD Signatures Enabled (Y/N)</b>	<b>Signature Comparison Desired</b>	<b>Recommended with RMS of EGD</b>	<b>User Specified Signature</b>
No	No	No	0 (Default - no check)
Yes	No	No	0 (Default - no check)
Yes	Yes	Yes	Current EGD signature

**(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 to 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 84)	Memory type of COMMREQ Status Word
Word 4	0-based <sup>12</sup>	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	Bit mask, set bit to be written to 1, rest to 0	The bit mask 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 selected bit.	Value to set the bit selected by the bit mask in Word 9.
Word 11	Remote signature	EGD Exchange signature. This should be 0 for PLCs when not using signatures. If 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.

**(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 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.

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</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

**(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 Bit Mask and Data Bits**

Word 9 of the Masked Write command contains the bit mask. 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.

For example:



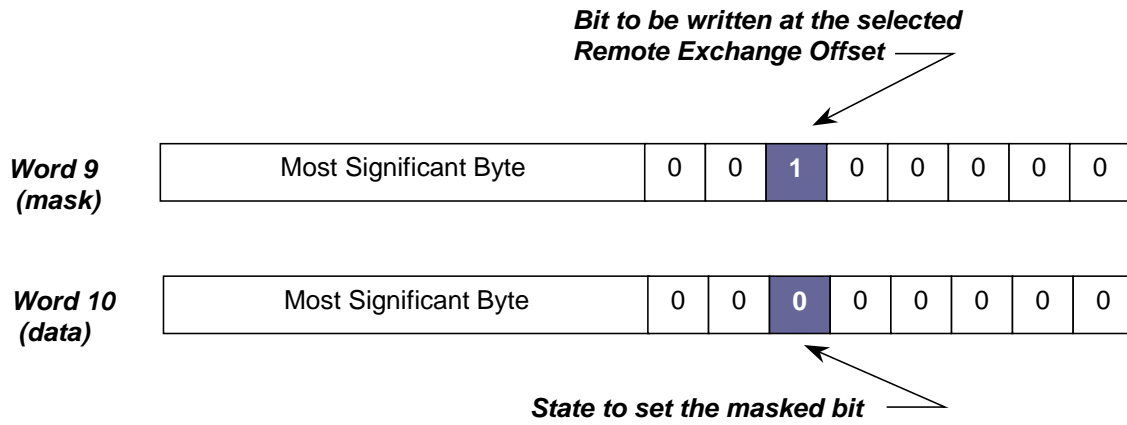


Figure 42: Example: Masked Write to EGD Exchange Bit Mask and Data Bits



## Chapter 7 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
- Differences between Series 90 and PACSystems SRTP Channels

### 7.1 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

Up to 32<sup>15</sup> channels can be established by a PACSystems Ethernet Interface. Channels can be individually monitored from the application program.

### 7.1.1 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.

### 7.1.2 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 establish 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.

### 7.1.3 Monitoring the Channel Status

The Ethernet Interface status bits occupy a single block of memory, which is specified during 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 “Monitoring the Ethernet Interface Status Bits” in Chapter 12, “Diagnostics.”

### 7.1.4 SRTP Channel Commands in a Redundant System

When configured for Redundant IP operation (see Chapter 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.

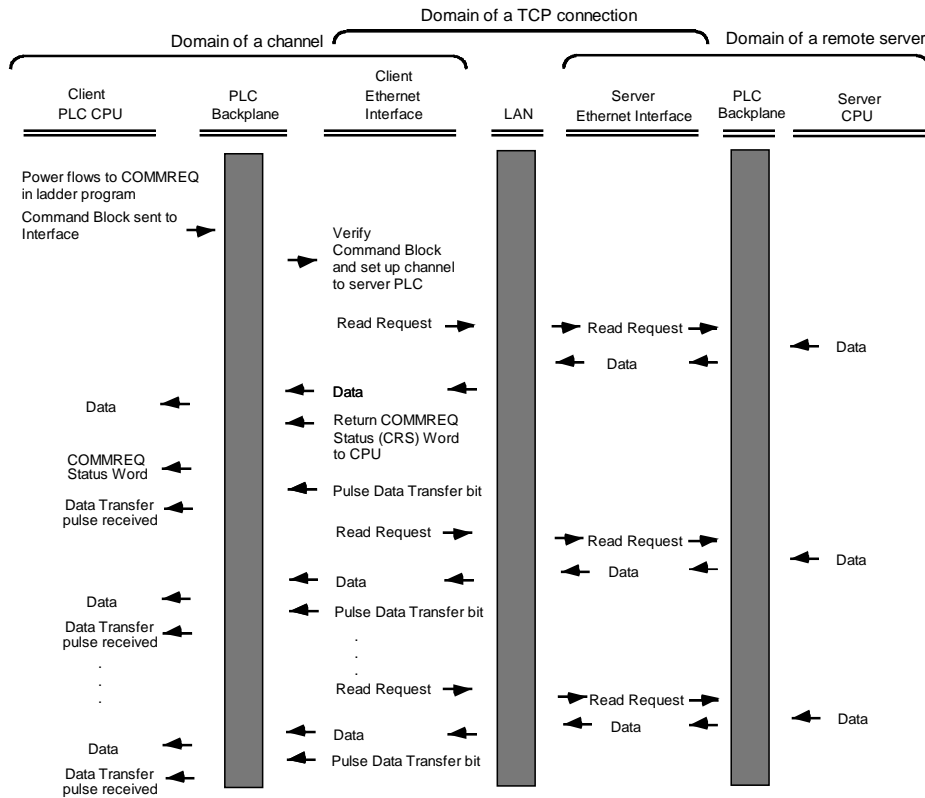
Because it can take some time to take down a TCP connection, the Redundant system should reserve a spare SRTP Client connection for each connection using the Redundant IP address. That will prevent temporary resource problems when establishing new SRTP Client connections to the new active unit while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

---

<sup>15</sup> The RX3i Embedded Ethernet interface supports a maximum of 16 channels.

### 7.1.5 Executing a Channel Command

The figure below shows how a Communications Request carries out a Channel command, in this case, Establish Read Channel.



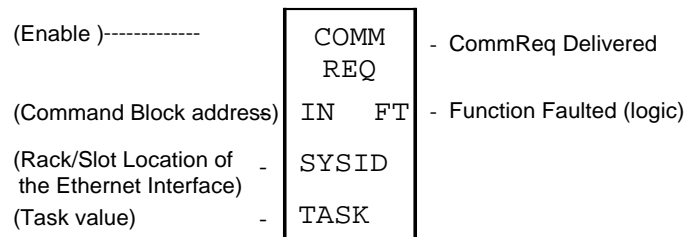
**Figure 43: COMMREQ Sequence for Establish Read Channel**

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 in a timely manner 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.

## 7.2 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 44: COMMREQ for Programming Channel Commands**

For the Channel Commands, the parameters of the COMMREQ are:

**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.

Rack	Slot	Hex Word Value
0	4	0004H
3	4	0304H
2	9	0209H
4	2	0402H

**TASK:** For the PACSystems Ethernet module, Task must be set to zero.

For the PACSystems CPU embedded Ethernet interface, Task must be set to the value 65536 (10000H) to address the CPU's Ethernet daughterboard.



### Caution

Entering an incorrect TASK value may cause the Ethernet Interface to fail.

**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.

### 7.2.1 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. It 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 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)

**(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. (COMMREQ Status Word values are described in Chapter 12.)

**(Word 3) 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)

**(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.

### 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.

## 7.2.2 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 server (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.

	Dec	(Hex)	
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 <sup>12</sup>	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)
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)
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)

The term **local PLC** is used here to identify the **client PLC**—the PLC that initiates the communications request.

The term **remote PLC** is used here to identify the **server PLC**—the PLC that responds to the communications request.

**(Word 7) Channel Command Number:** Word 7 requests that a read channel be set up. If the command is processed successfully, it will result in attempting 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

**Note:** If Time Unit Value is 5 (hours), then the maximum usable value of Number of Time Units is 5965.

**(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. Note: The COMMREQ Status is issued only once for each COMMREQ; for more information, see “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.

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. Note: The COMMREQ Status is issued only once for each COMMREQ; for more information, see “Using COMMREQs for Channel Commands.” The transfer can still complete even after a timeout occurs. You can determine when the pending transfer completes by monitoring the Channel Error and Data Transfer bits. 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 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 taking action.

**(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. 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>16</sup>	0	Program Block Local register memory (word mode)
%P <sup>16</sup>	4	Program register memory (word mode)
%W <sup>17</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>14</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)

**(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 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

<sup>16</sup> Can only be accessed in the Remote PLC

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

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.

**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.

### 7.2.3 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	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 <sup>12</sup>	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)
The term <b>local PLC</b> is used here to identify the <b>client PLC</b> —the PLC that initiates the communications request.	Word 9	00000	(0000)	Number of write repetitions (write indefinitely)
	Word 10	00003	(0003)	Time unit for write period (3=seconds)
	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)
The term <b>remote PLC</b> is used here to identify the <b>server PLC</b> —the PLC that responds to the communications request.	Word 14	00050	(0032)	Local PLC - Starting address from which to write data (%R50)
	Word 15	00008	(0008)	Remote PLC - Memory type at which to store data (%R)
	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)

**(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 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. Note: The COMMREQ Status is issued only once for each COMMREQ; for more information, see “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 writes, the next write transfer will be issued only after the pending write transfer completes.

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. Note: The COMMREQ Status is issued only once for each COMMREQ; for more information, see “Using COMMREQs for Channel Commands”. The transfer can still complete even after a timeout occurs. You can determine when the pending transfer completes by monitoring the Channel Error and Data Transfer bits. 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 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 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.



## 7.2.4 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 an Ethernet Interface to a host application SRTP server. 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 remote host. Repeat the send 10 times, once every 7 seconds, with a timeout of 500ms for each transfer.

	Dec	Hex)	
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)	Memory type of COMMREQ status word (%R)
Word 4 <sup>12</sup>	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)
The term <i>local PLC</i> is used here to identify the <i>client PLC</i> —the PLC that initiates the communications request.	Word 10	00003	(0003) Time unit for send period (3=seconds)
	Word 11	00007	(0007) Minimum interval between host accesses (every 7 seconds)
	Word 12	00050	(0032) Timeout on each individual 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)
	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)
The term <i>Remote Host</i> is used here to identify the <i>SRTP Host server</i> .	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)

**(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 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 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 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 example, if Word 13=16 and Word 14=2, the starting address will be %I9). Valid ranges of values depend on the PLC's memory ranges.

**(Word 15) Local 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 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.

### 7.2.5 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.

	Dec	Hex	
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)	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

**(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.

## 7.2.6 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 has no effect on 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.

	Dec	Hex)	
	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)
The term <b>local PLC</b> is used here to identify the <b>client PLC</b> —the PLC that initiates the communications request.	Word 4 <sup>12</sup>	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)

**(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 Chapter 12.

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.

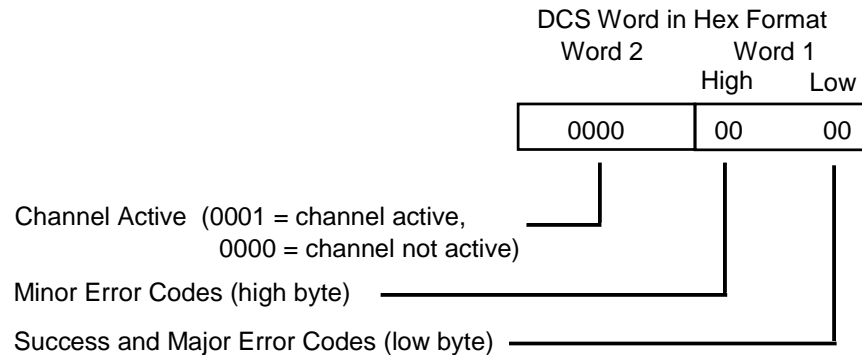
The DCS words location is specified in the Retrieve Detailed Channel Status Command. The contents of these status words are defined below.

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 hexadecimal form to differentiate the high and low bytes.



**Figure 45: Interpreting Detailed Channel Status Words**

### **7.3 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 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 interlock 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.

### 7.3.1 **COMMREQ Sample Logic**

In the sample logic that follows, the input values for the Block Move Functions are taken from the *Establish Read Channel (2003)* 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.

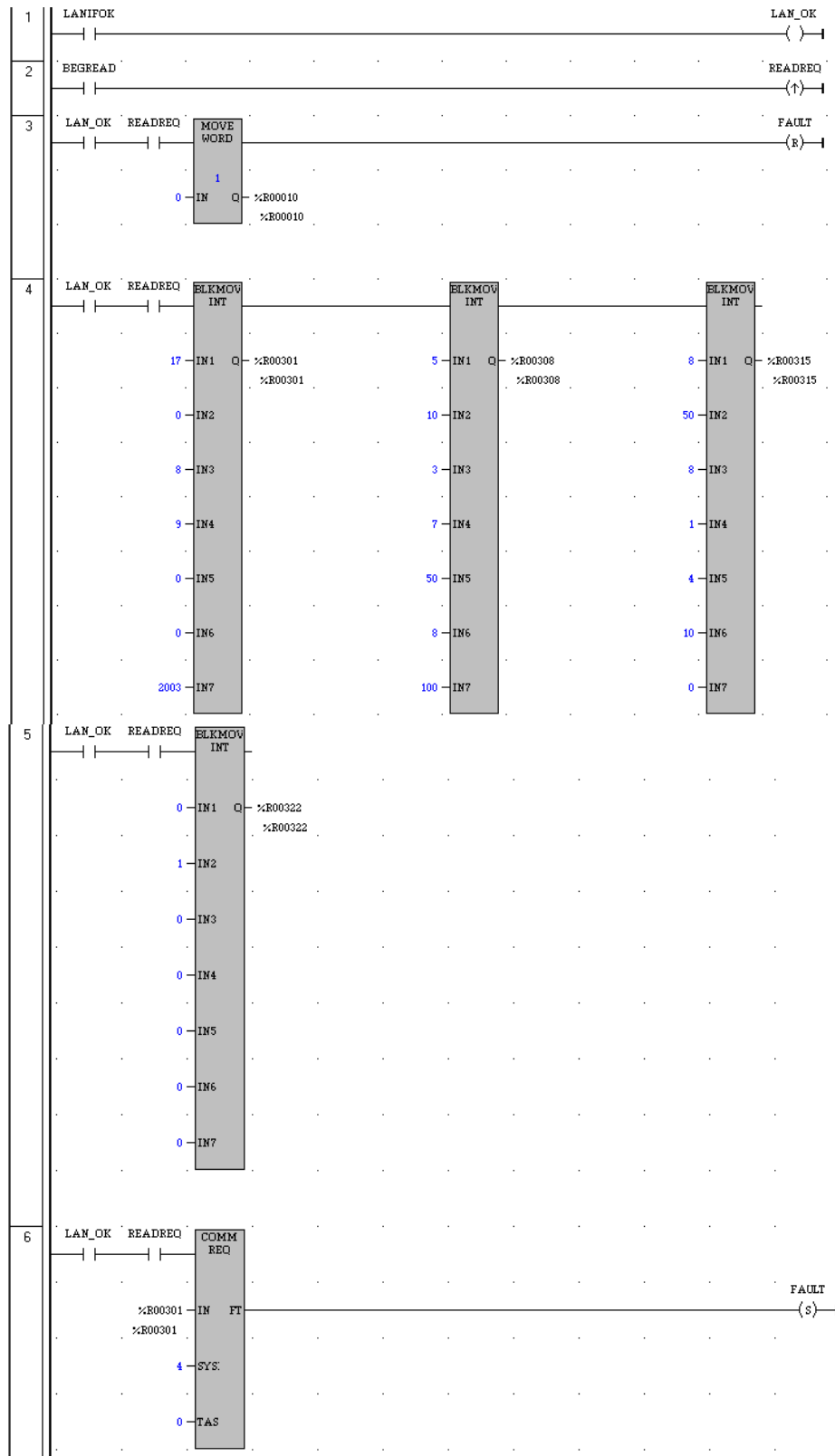


Figure 46: Sample Ladder Logic for COMMREQ

**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 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 rack-based Ethernet module, Task must be set to 0. For a PACSystems CPU embedded Ethernet interface, Task must be set to 65536 (10000H).
- The FT output (energizes the FAULT coil in this example) is turned ON (set to 1) if there were problems preventing the delivery of the Command Block to the Ethernet Interface. In this case, the other status indicators are not updated for this COMMREQ

### 7.3.2 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 – Comm\_req aborted”

The PLC logic program should send retry the COMMREQ after a short delay.

### 7.3.3 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.

#### 7.3.4 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 to 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.

#### 7.3.5 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 in order to establish another client channel (a COMMREQ Status of 0xA890 is returned if no TCP resources are currently available; 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 a number of iterations, or run forever. Additionally, SRTP Channels allows 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, time 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.

### 7.3.6 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.

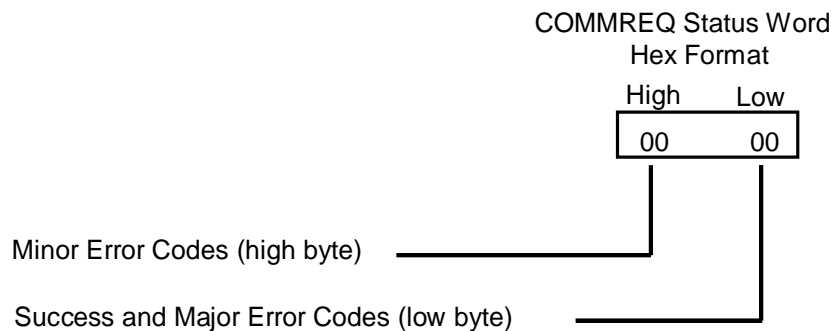
If the application is seeing timeouts during channel startup, there are a few different options:

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.

## 7.4 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 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.

### 7.4.1 Format of the COMMREQ Status Word



**Figure 47: 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.

Chapter 11 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.

### **Differences between Series 90 and PACSystems SRTP Channels**

This section lists differences between the Series 90 implementation of SRTP Channels and the PACSystems implementation.

1. The TCP Connect Timeout for an SRTP Channel on the Series 90 was 90 seconds. For PACSystems, a new SRTP AUP parameter, "SRTP Channel TCP Connect Timeout", will be added that specifies the amount of time to wait for a TCP connection to be established: `hconn_tout`. The default value will be set to 75 seconds, and its maximum value is 75 seconds, which is the maximum value we can specify to the current OS. Minimum value is 10 milliseconds.
2. PACSystems has a TCP Linger Period, which is the period of time the OS Network stack hangs onto the TCP resources associated with a connection after it is closed. The TCP resources from a channel that was stopped will become available again after the 60 second TCP linger period has expired. The Series 90 had no linger period.
3. The Series 90 SRTP Channel implementation performed a normal stopping of the channel on a Run-to-Stop transition. On PACSystems, a Run-to-Stop transition causes an Abrupt Shutdown, avoiding the TCP Linger period and reducing the chance of exhausting TCP resources when quickly transitioning between Run->Stop and Stop->Run.
4. On the Series 90, if an Abort/Abort All Channel COMMREQ is issued, followed by an Establish Read/Write/Send Info Report Channel COMMREQ before the COMMREQ Status Word for the Abort/Abort All has been updated, the Establish Read/Write/Send Information Report was dropped and the COMMREQ Status Word was not updated (it remained zero). For PACSystems, the Establish Read/Write/Send Information Report COMMREQ is discarded and its COMMREQ Status Word is set to a failure value (A990). That indicates it was discarded because the application logic issued the command while an Abort was in progress.
5. For PACSystems, new COMMREQ Status Codes are defined. See Chapter 11 for details.
6. The PACSystems implementation supports Re-tasking to a different remote device (different IP Address).
7. The Series 90-70 limited the total number of TCP connections shared between SRTP Client Channels and SRTP Server to 48. TCP connections not shared between SRTP Server and Client, and the maximum TCP Connections allowed for PACSystems are increased as follows:
  - a. maximum of 48 Server TCP connections for Rack-based and RX7i Embedded<sup>18</sup>
  - b. maximum of 32 Client Channel TCP connections<sup>19</sup>

<sup>18</sup> 32 SRTP server connections for RX3i Embedded Ethernet interface

<sup>19</sup> 16 Client Channel connections for RX3i Embedded Ethernet interface



## Chapter 8 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

### 8.1 Modbus/TCP Server

The PACSystems products listed below support Modbus/TCP Server functionality:

- CPU010 and CPU020 with primary firmware version 3.0 or later.
- CRE020 with Ethernet firmware version 3.0 or later.
- RX7i: IC698ETM001 and RX3i IC695ETM001 with firmware version 3.0 or later.
- CRE305 and CRE310 with primary firmware version 7.30 or later

#### 8.1.1 Modbus/TCP Server Connections

The Modbus/TCP Server supports up to 16 simultaneous 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.

#### 8.1.2 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.

#### 8.1.3 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.

#### 8.1.4 Station Manager Support

The Modbus/TCP Server supports the standard Station Manager commands: STAT, TALLY, and TRACE, plus the Modbus/TCP server-specific KILLMS command. The Modbus/TCP Server task letter is "o".

### 8.2 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.

## 8.2.1 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)	---	%I1 – 32768 (bits)
---	---	1 – 32640 (16-bit words)	---	---	%AI1 – 32640 (16-bit words)
---	---	---	---	1 – 32768 (bits)	%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)

### 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:

$$\text{File} = \frac{W-1}{10,000} + 1 \quad \left( \text{Discard any fractional portion; round the result downward to the next integer value.} \right)$$

$$\text{Record} = W - (10,000 (F - 1))$$

Figure 48: 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

## **8.2.2 Address Configuration**

Address mapping is done in the Machine Edition 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.

<b>Number</b>	<b>Modbus Register</b>	<b>Start Address</b>	<b>End Address</b>	<b>PLC Memory Address</b>	<b>Length</b>
1	0xxxx - Coil Table	1	32768	%Q00001	32768
2	1xxxx Discrete Table	1	32768	%I00001	32768
3	3xxxx Input Registers	1	64	%AI00001	64
4	4xxxx - Register Table	1	1024	%R00001	1024
5	6xxxx - Internal Table	0	0	%W0001	0

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 Machine Edition, 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.

### 8.3 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 5 Write Single Coil 15 Write Multiple Coils	0xxxx	1	32768	%Q00001	32768
2 Read Discrete Inputs	1xxxx	1	32768	%I00001	32768
3 Read Holding Registers 6 Write Single Register 16 Write Multiple Registers 22 Mask Write Register 23 Read/Write Multiple Registers	4xxxx	1	1024	%R00001	1024
4 Read Input Registers	3xxxx	1	64	%AI00001	64
7 Read Exception Status 8 Diagnostics	n/a	N/A	N/A	N/A	N/A
20 Read File Record 21 Write File Record	6yxxxx	1	0	%W00001	0

## Chapter 9 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
- Differences between Series 90 and PACSystems Modbus/TCP Channels

### 9.1 *The 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

### 9.1.1 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 execution of the COMMREQ Function Block

The figure below illustrates the relationship of these elements:

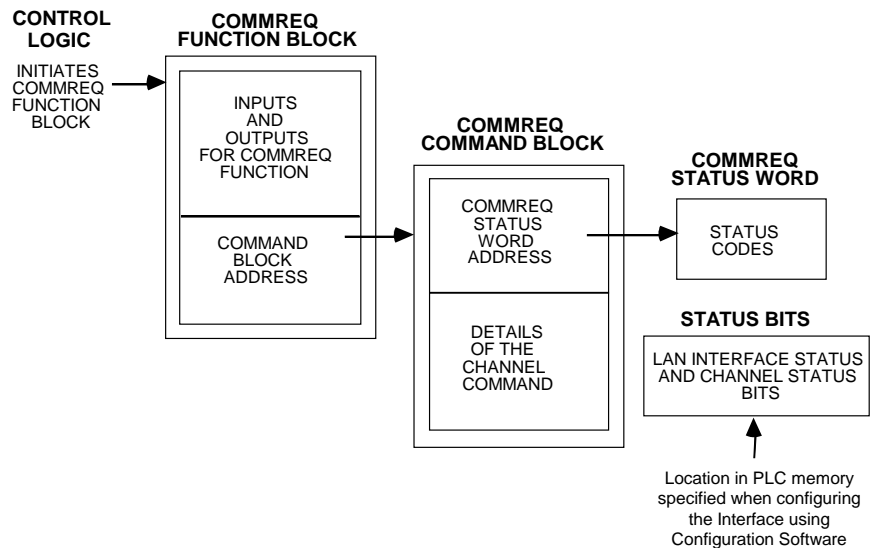


Figure 49: Phases of a COMMREQ Execution

### 9.1.2 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.

### 9.1.3 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.

### 9.1.4 Modbus/TCP Channel Commands

The Channel commands are a set of client commands used to communicate with a server. Up to 32<sup>15</sup> 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.



### 9.1.5 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 execution of the COMMREQ function when the Ethernet interface is not operational. Following 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.

### 9.1.6 Operation of the Communications Request

Figure 50 below 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.

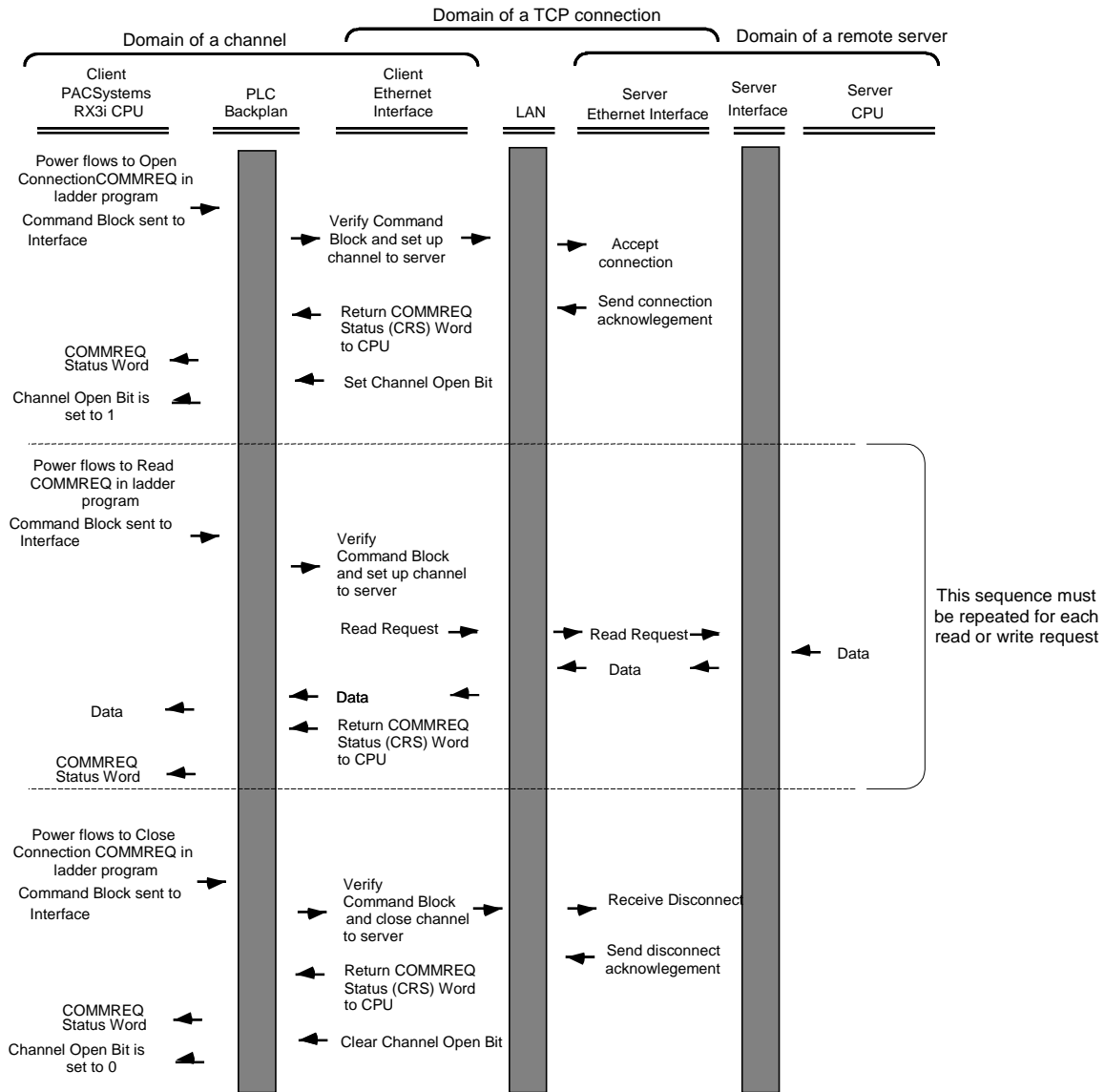


Figure 50: Illustration of Phased Operation of a COMMREQ

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.

## 9.2 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.

### 9.2.1 The COMMREQ Function Block

The Communications Request is triggered when the logic program passes power to the COMMREQ Function Block.

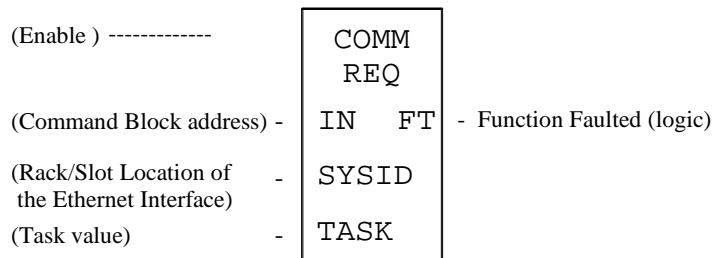


Figure 51: The COMMREQ Function Block

Each of the inputs and outputs are discussed in detail below. It is important to understand that the Command Block address points to the location in memory you have setup 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.

Examples:

Rack	Slot	Hex Word Value	Notes
0	1	16#0001	Slot 1 is used for the Ethernet daughterboard on an RX7i CPU.
0	4	16#0004	
3	4	16#0304	
2	10	16#020A	
4	2	16#0402	

**TASK:** For the RX3i and Rx7i ETM001 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.



### Caution

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

**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.

## 9.2.2 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 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)

When entering information for the Command Block, refer to these definitions:

**(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. 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) 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.

## 9.3 Modbus/TCP Channel Commands

This section describes the operation of the Channel commands. A detailed description and example of each Channel command is 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.

### 9.3.1 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 the section "Status Data" later in this document 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.

	Dec	(Hex)	
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 <sup>12</sup>	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

**(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 10 specifies the value of the first octet of the IP Address.

**(Word 12) IP Address 2<sup>nd</sup> Octet:** Word 11 specifies the value of the second octet of the IP Address.

**(Word 13) IP Address 3<sup>rd</sup> Octet:** Word 12 specifies the value of the third octet of the IP Address.

**(Word 14) IP Address 4<sup>th</sup> Octet:** Word 13 specifies the value of the fourth octet of the IP Address.

### 9.3.2 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.

	Dec	(Hex)	
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 <sup>12</sup>	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)

**(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.

### 9.3.3 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. Refer to **Local PLC Memory Type** on page 138 for the list of data areas and identification codes for the PLC. Note that the first item referred to in each data area is 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. See the section “Status Data” later in this chapter for detailed CRS failure codes.

### 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.

	Dec	(Hex)	
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 <sup>12</sup>	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

**(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 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>14</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)

**(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. Note: The function code determines the Modbus server address area, Word 12 is the address within this area.

**(Word 13) Number Registers in Remote Device:** Words 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.

### 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.

#### Dec (Hex)

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 <sup>12</sup>	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

**(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 138.

**(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.

	Dec	(Hex)	
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 <sup>12</sup>	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

**(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 138.**(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.

	<b>Dec</b>	<b>(Hex)</b>	
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 <sup>12</sup>	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	00024	(0018)	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

**(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 138.

**(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) 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.

### 9.3.4 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. The source of data can be any of the PLC data areas (see **Local PLC Memory Type** on page 138).

**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 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. See the section "Status Data" later in this document for detailed CRS failure codes.

**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.

	Dec	(Hex)	
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 <sup>12</sup>	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

**(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 138.

**(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:** 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.

	Dec	(Hex)	
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 <sup>12</sup>	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

**(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 138.

**(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.

	<b>Dec</b>	<b>(Hex)</b>	
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 <sup>12</sup>	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

**(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 16, Write Multiple Registers

**(Word 10) Local PLC Memory Type:** Words 10–11 specify the location in the local PLC where the Ethernet interface will get the data to be written to the remote PLC. Values for Word 10 are listed on page 138. The value 16 specifies Discrete Input Memory %I (byte mode).

**(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 register in the remote Modbus/TCP device.

**(Word 13) Number Registers in Remote Device:** Words 13 specifies the quantity of registers to write to 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.

### 9.3.5 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:

$$\text{Register value} = ((\text{Current register value}) \text{ AND } (\text{And Mask Value})) \text{ OR } ((\text{OR Mask Value}) \text{ AND } (\text{NOT}(\text{And Mask Value})))$$

#### 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.

	Dec	(Hex)	
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 <sup>12</sup>	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.

### 9.3.6 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. Note, 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.

	Dec	(Hex)	
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 <sup>12</sup>	00009	(0009)	CRS word address minus 1 (%R10)
Word 5	00000	(0000)	Reserved
Word 6	00000	(0000)	Reserved
Word 7	03005	(0BBD)	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
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

**(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. Values for Word 10 are listed on page 138. 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 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 138. 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.

## 9.4 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.

### 9.4.1 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 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 “Monitoring the Ethernet Interface Status Bits” in Chapter 12, “Diagnostics.”

#### **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 is 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.

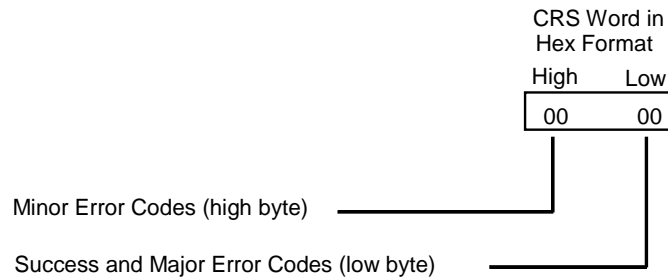
The FT Output passes power upon the following errors:

- 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 52: Interpreting the COMMREQ Status Word**

The Ethernet Interface reports the status of the COMMREQ back to the status location. See Chapter 11, “Diagnostics” for COMMREQ major and minor error codes that may be reported in the CRS words for Modbus/TCP commands.

## 9.5 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

### 9.5.1 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 interlock 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.

**Note:** When using a Write Data or Read/Write COMMREQ, data is not read from the local PLC synchronously with 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.

The example ladder program segment starting on the next page illustrates how to incorporate these important points in your program.

## 9.5.2 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. All other nicknames may be assigned as you choose.

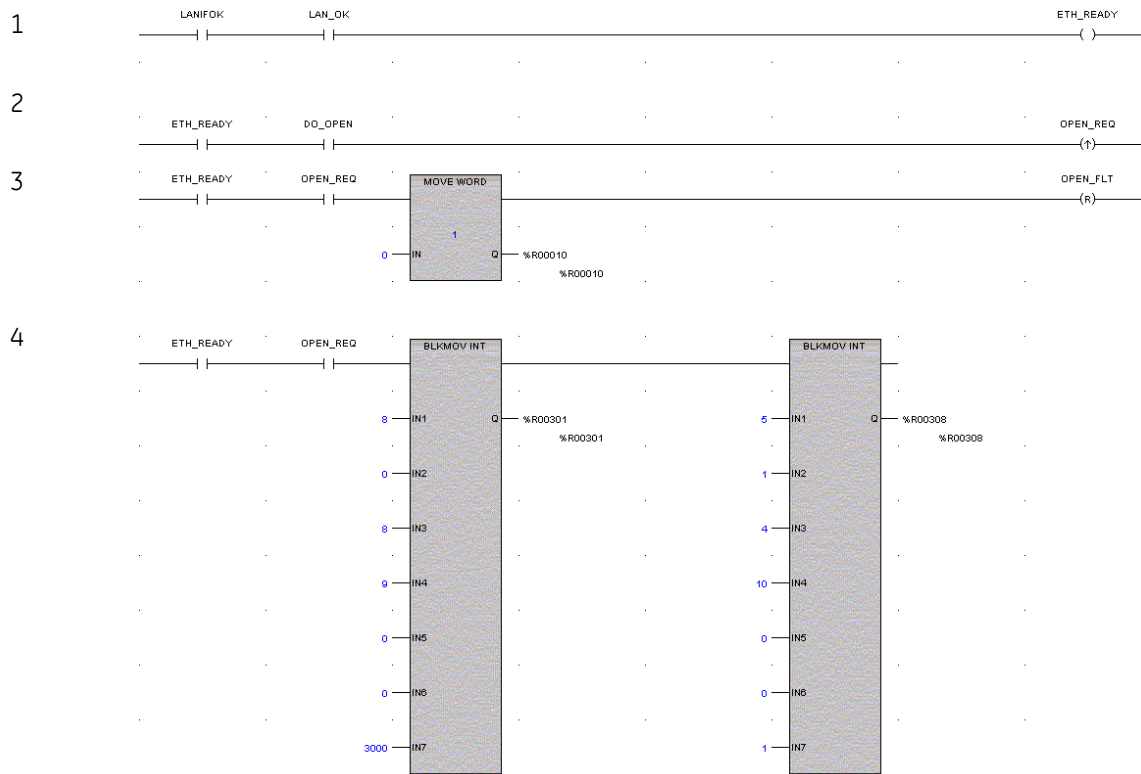


Figure 53: COMMREQ Ladder Logic Segment

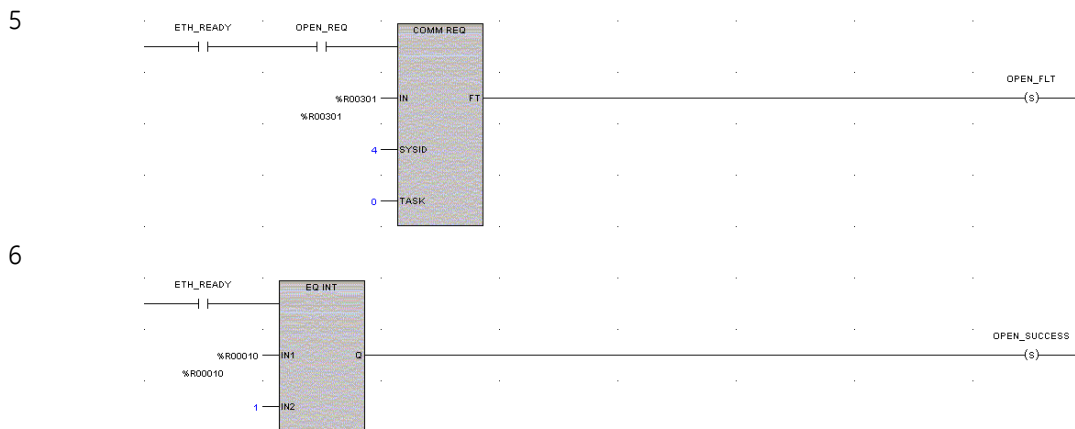
**Rung # 1:** Input LANIFOK (bit 16 of the LAN Interface Status bits) monitors the health of the Ethernet interface. Input LAN\_OK (bit 13 of the LAN Interface Status bits) monitors the online/offline status of the Ethernet interface. If both bits are set it is OK to send a COMMREQ and the ETH\_READY coil is ON. ETH\_READY is used as an interlock for Rungs 2-16.

**Rung # 2:** When ETH\_READY is 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 be cleared each time before initiating a COMMREQ function.

**Rung # 4:** The BLKMOV INT functions 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.



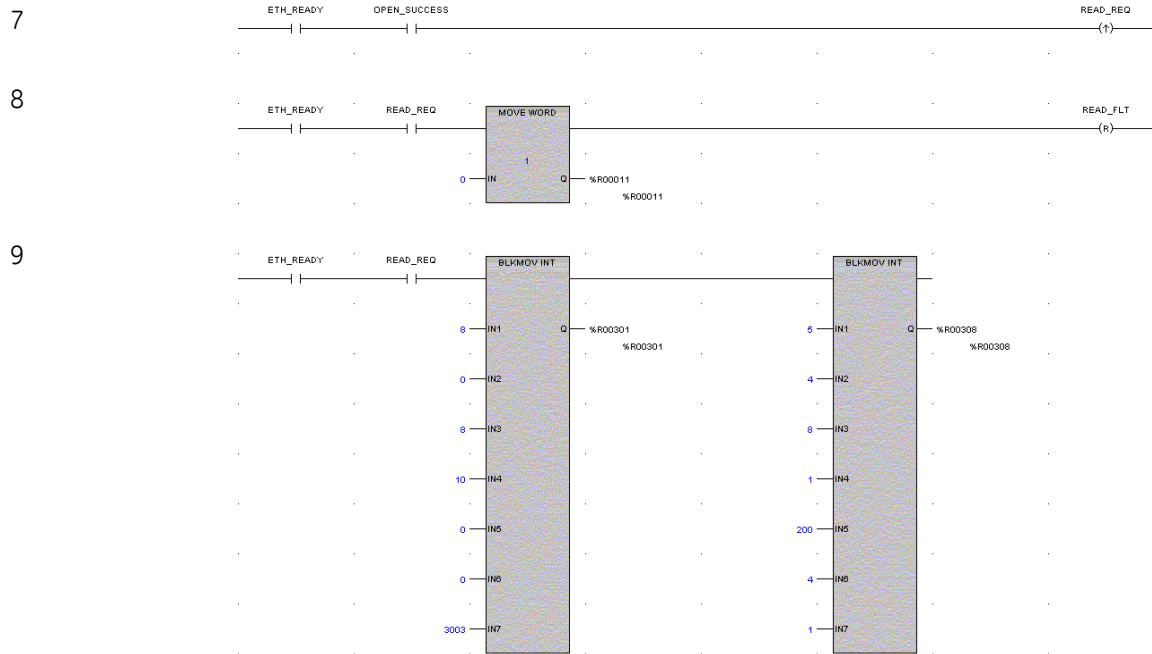
**Figure 54: COMMREQ Ladder Logic Segment (continued)**

**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 and Rx7i ETM001 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 there were problems 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. The CRS word change to 1 sets coil OPEN\_SUCCESS.





**Figure 55: COMMREQ Ladder Logic Segment (continued)**

**Rung # 7:** When OPEN\_SUCCESS is 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 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.

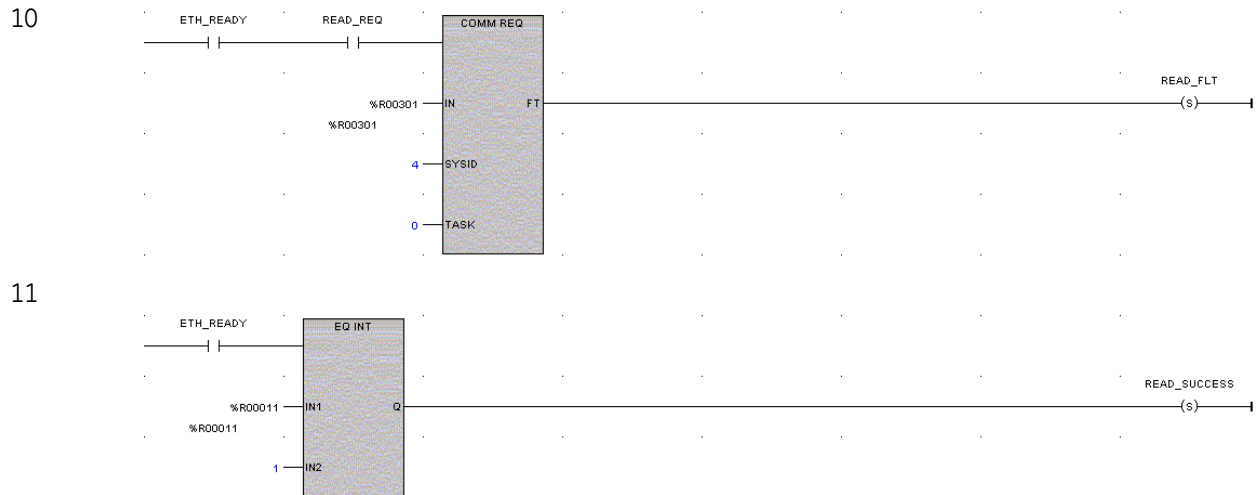


Figure 56: COMMREQ Ladder Logic Segment (continued)

**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 and Rx7i ETM001 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 there were problems 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 sets coil READ\_SUCCESS.

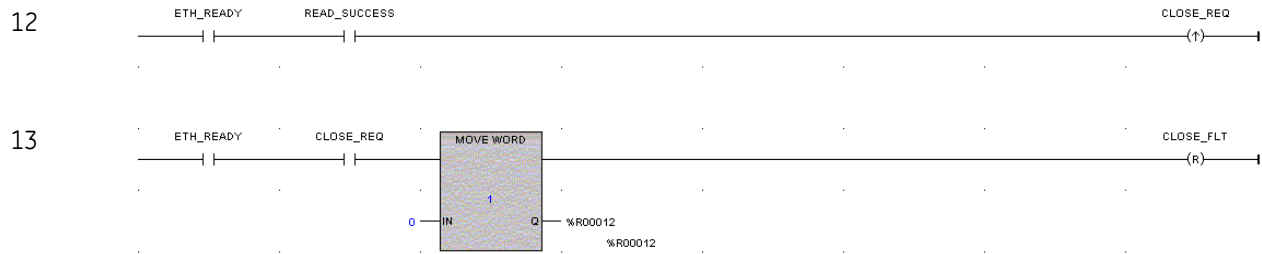


Figure 57: COMMREQ Ladder Logic Segment (continued)

**Rung # 12:** When READ\_SUCCESS is set 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.

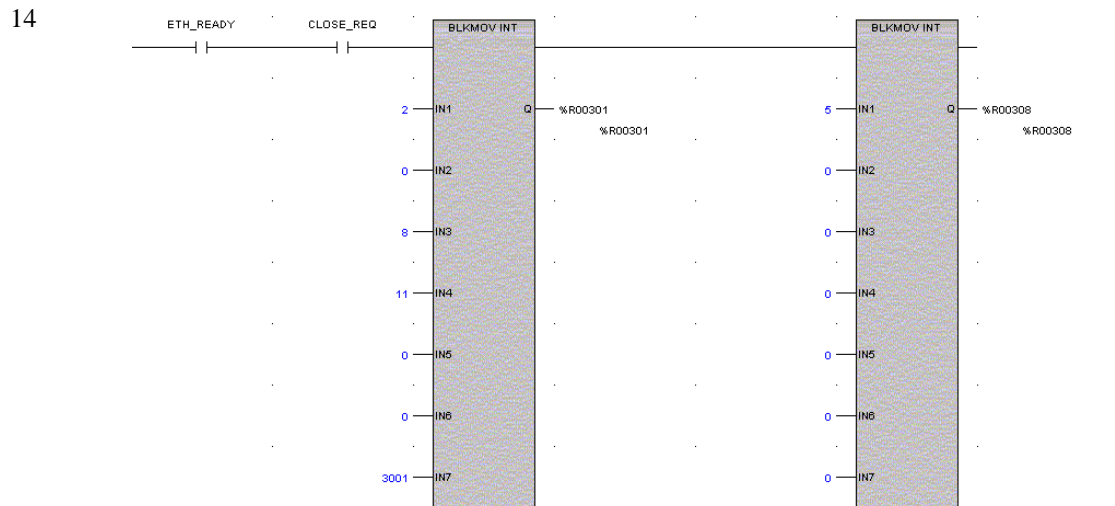


Figure 58: COMMREQ Ladder Logic Segment (continued)

**Rung # 14:** The BLKMOV INT functions 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.

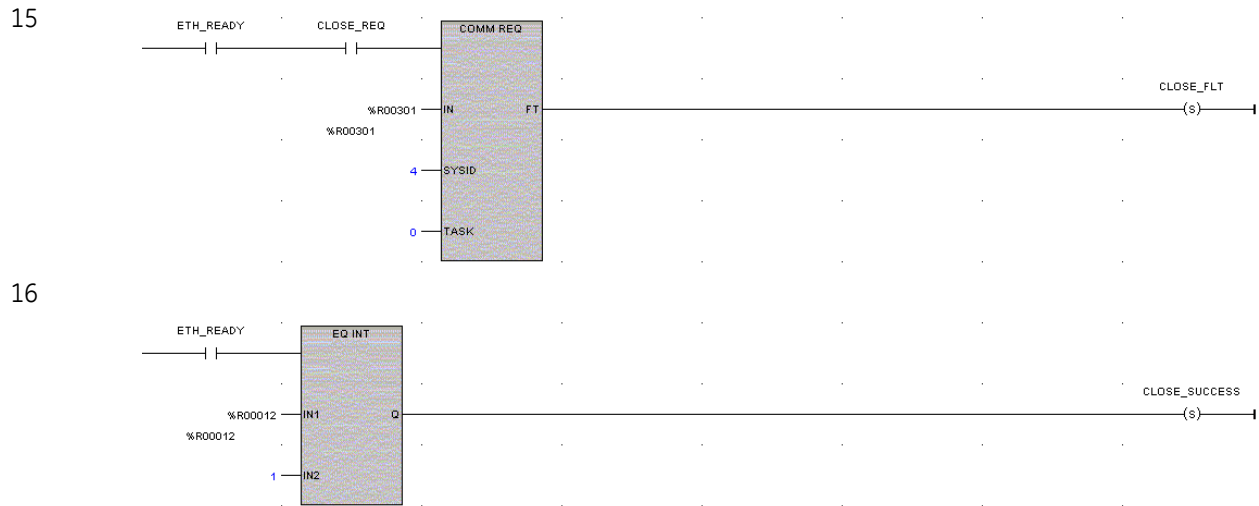


Figure 59: COMMREQ Ladder Logic Segment (continued)

**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 and Rx7i ETM001 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 there were problems 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. The CRS word change to 1 sets coil CLOSE\_SUCCESS.

### 9.5.3 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 and Rx7i ETM001 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.

## 9.5.4 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.

## 9.6 Differences between Series 90 and PACSystems Modbus/TCP Channels

This section lists the known differences between the Series 90 implementation of Modbus/TCP Channels and the PACSystems implementation.

1. On the 90-30 CMM321 if a Modbus error response is received for a Modbus/TCP channel, the Ethernet interface closes the TCP connection and updates the CRSW with an appropriate error code. For PACSystems Ethernet, the Modbus error response results in an updated CRSW with an appropriate error code but the TCP connection is NOT closed.
2. A CRSW of 0x8390 (Invalid Server Memory Type) is returned when an invalid Modbus Function code is specified for the CMM321. For PACSystems Ethernet, an improved CRSW of 0xB690 (Invalid/Unsupported Modbus Function Code) is returned.
3. The TCP connect timeout (i.e. the amount of time to wait for the Remote server or Gateway to establish a TCP connection with a Modbus/TCP Channel) is 90 seconds on the Series 90 and 75 seconds on PACSystems. An error is returned in the CRSW for the Open Modbus/TCP Connection COMMREQ when this timeout occurs.
4. The station manager command "stat m" on the Series 90 results in displaying "Closed" for specific Closed channels while PACSystems Modbus/TCP Channels results in displaying nothing for a specific Closed channel.
5. When sending a Close Modbus/TCP Connection COMMREQ, the PACSystems Modbus/TCP Client will return a success CRSW (0x0001) while the CMM321 module returns an error CRSW.
6. The rules for Endian conversion when transferring between Word and Bit types of memory are different in order to make these types of conversions consistent.

### 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.

<i>Memory Location / Type</i>	<i>Memory value example</i>	<i>Transfer Direction</i>	<i>Memory Location / Type</i>	<i>Resulting Value After Transfer</i>	<i>Notes</i>
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:

<i>Memory Location / Type</i>	<i>Memory value example</i>	<i>Transfer Direction</i>	<i>Memory Location / Type</i>	<i>Resulting Value After Transfer</i>	<i>Notes</i>
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>



## Chapter 10 OPC UA Server

---

OPC Unified Architecture, or OPC UA, is a communication standard published by the OPC<sup>20</sup> 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 (Proficy 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.

- 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

The sections that follow provide details for each of these topics.

---

<sup>20</sup> OPC originally meant "Object Linking and Embedding (OLE) for Process Control", but is now said to stand for "Open Productivity and Connectivity".

## 10.1 Application Logic to Control the OPC UA Server

The OPC UA server is controlled by means of a service request (SVC REQ function block). The service request allows you to start, stop, restart and clear OPC specific information.

### 10.1.1 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

**Note:** All other protocol codes are reserved, and if used, the SVC\_REQ function will not pass power.

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

**Note:** All other sub-functions are reserved; if used, the SVC\_REQ function does not pass power.

### 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:** This request can only be successfully performed when the OPC UA server is stopped.

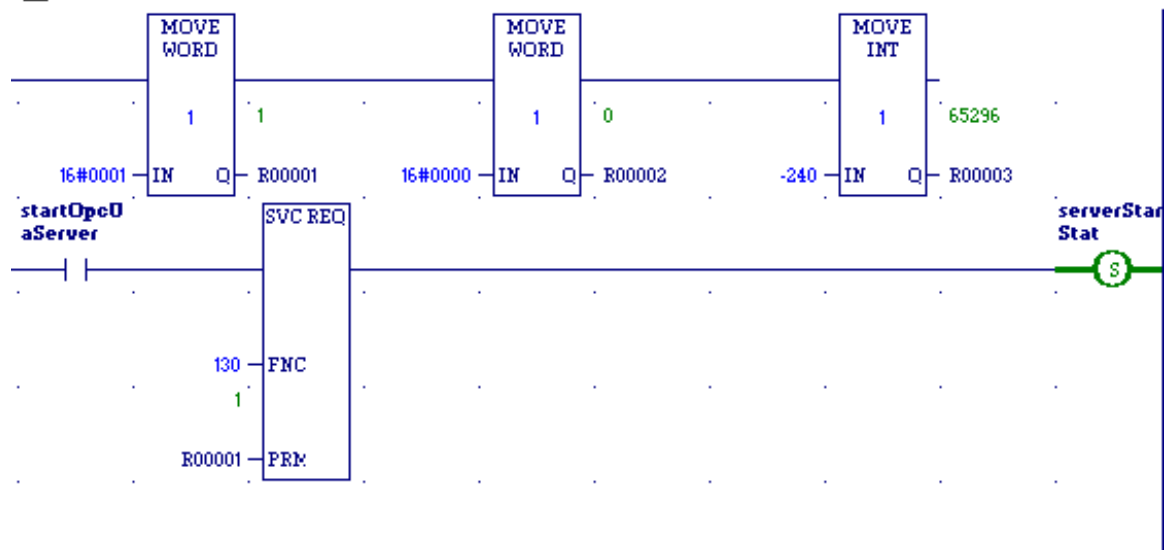
Parameters for the START function service request function block are:

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 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).

#### Example:

 Start OPC UA Server Service Request Example



**Note:** In this example, a Time Zone Offset of -240 was used, meaning local time is UTC time minus 240 minutes (4 hours).

### 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.

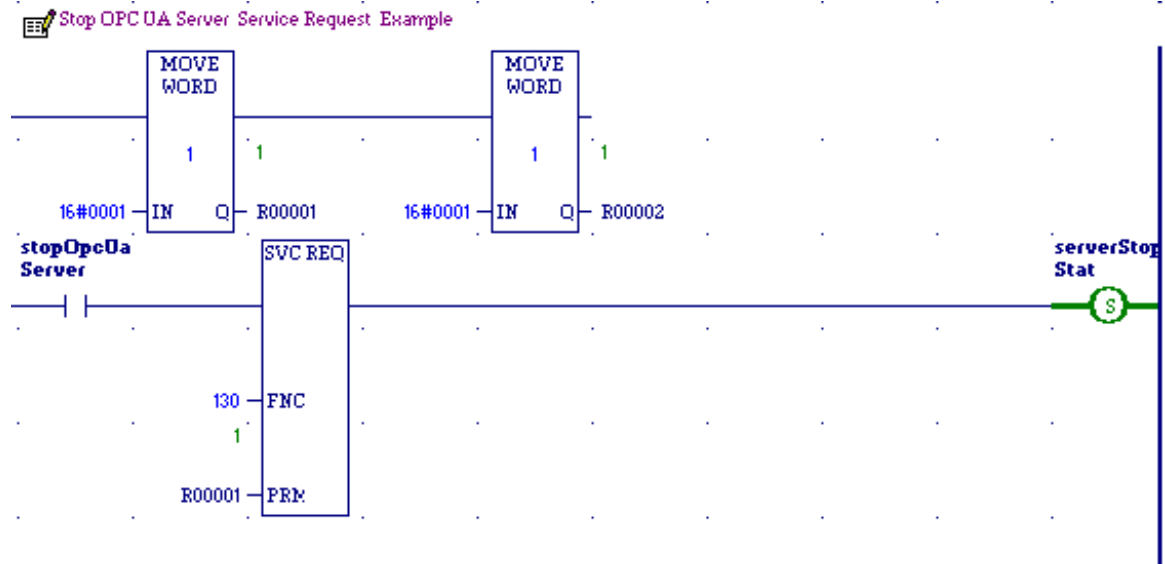
**Note:** This request can only be successfully performed when the OPC UA server is started.

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.

**Example:**



### OPC UA Server – Service Request – CLEAR

This function clears the configuration files and certificates used by the OPC UA Server on the controller.

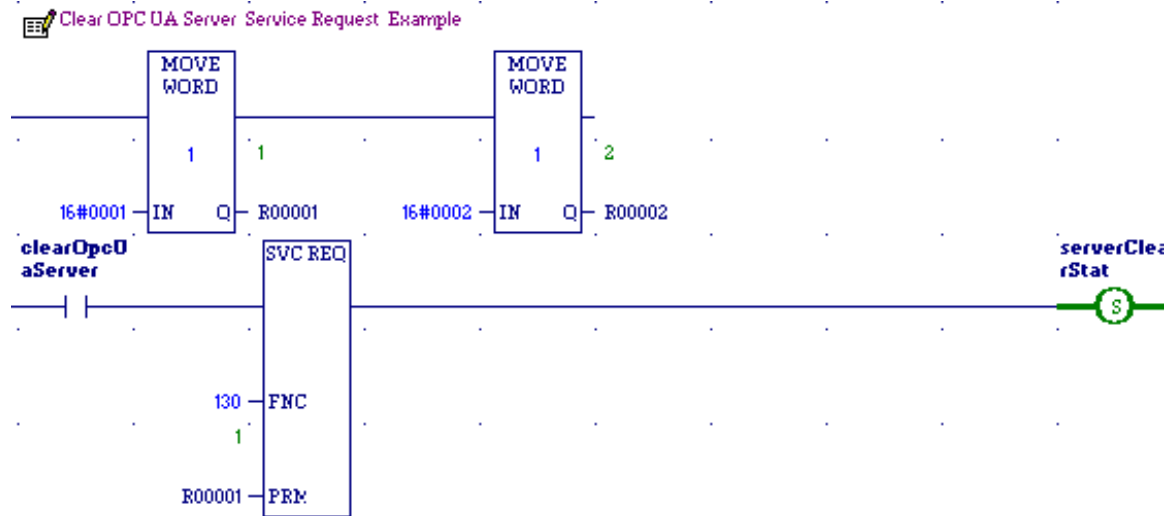
**Note:** This request can only be successfully performed when the server is stopped.

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

If the SVC\_REQ does not pass power, the operation did not complete.

#### Example:



### OPC UA Server – Service Request – RESTART

This function stops and then restarts the OPC UA server on a target.

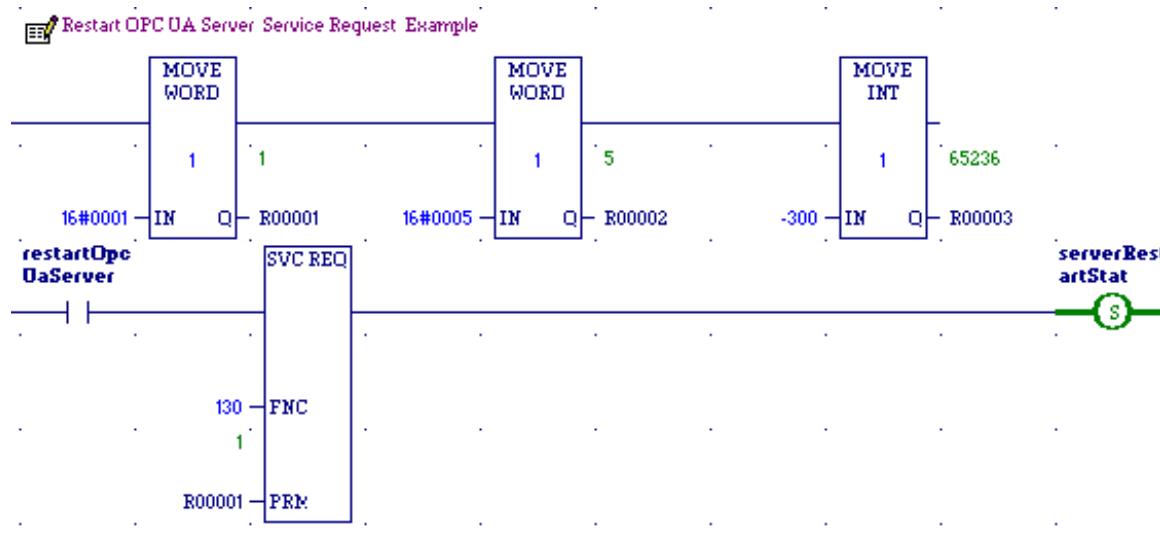
**Note:** This request can only be successfully performed when the OPC UA server is started.

Parameters for the START function service request are:

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.

**Example:**



**Note:** In this example a Time Zone Offset of -300 was used, meaning local time is UTC minus 300 minutes (5 hours).

### OPC UA Server – Service Request – SERVER\_STATUS

The SERVER\_STATUS sub-function code can be used to obtain info about the server status. The sub-function uses the following bitmask

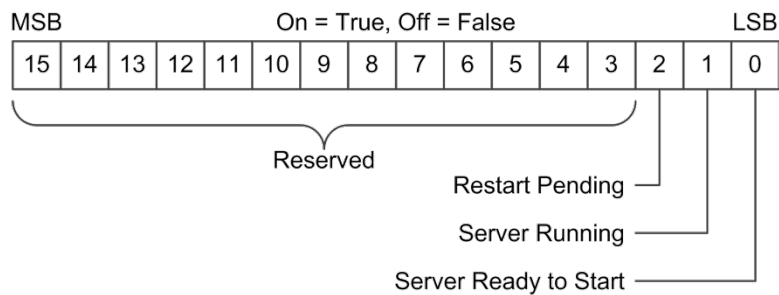
OPC_UA_SERVER_STAT_READY_TO_START_BITMASK	0x0001
OPC_UA_SERVER_STAT_RUNNING_BITMASK	0x0002
OPC_UA_SERVER_STAT_RESTARTS_PENDING_BITMASK	0x0004

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.

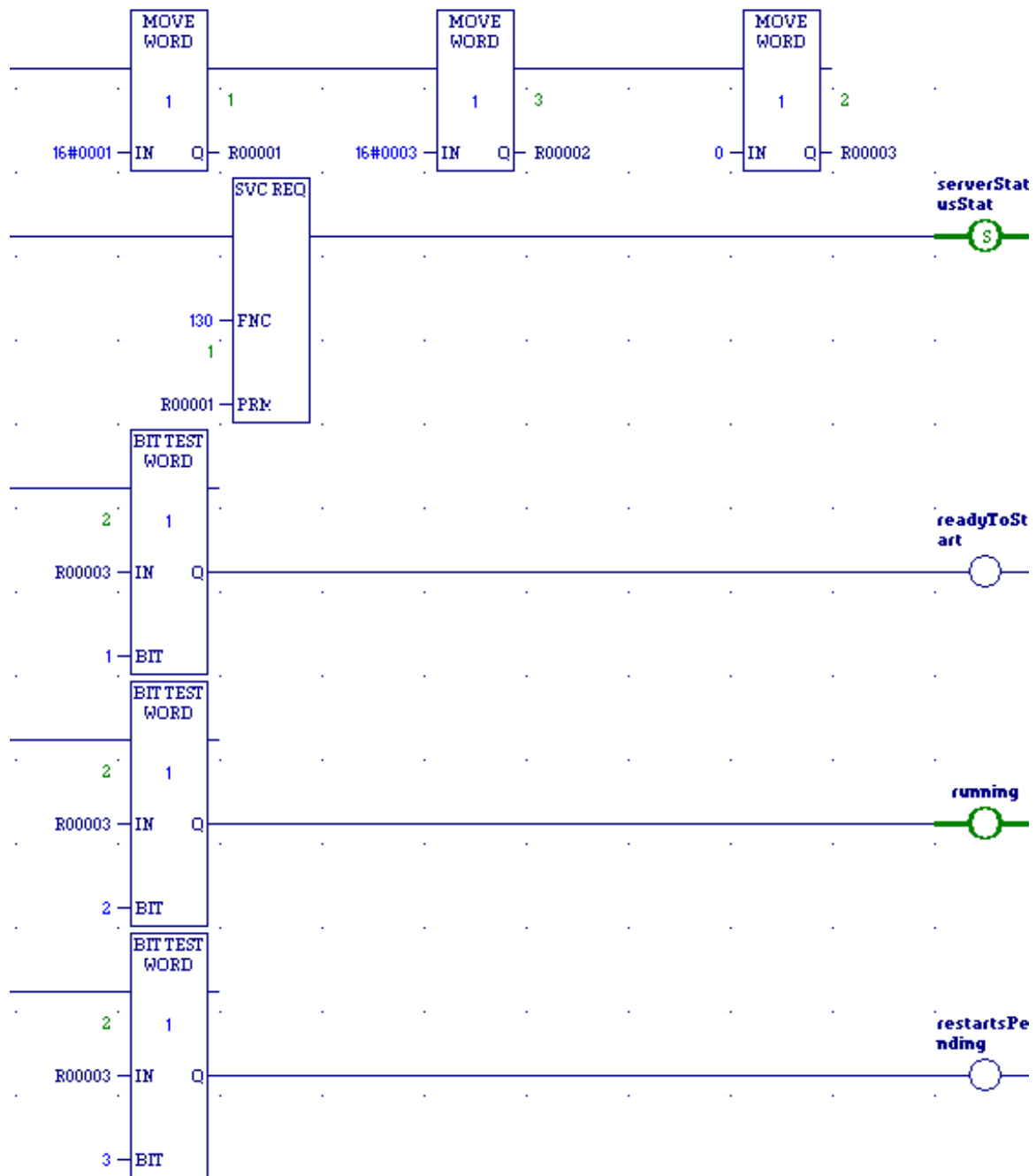
The SERVER\_STATUS word bit definitions are displayed below.



**Figure 60: SERVER\_STATUS Word bit definitions**

Example:

Get OPC UA Server Status Service Request Example





### OPC UA Server – Service Request – CONFIG\_STATUS

The CONFIG\_STATUS sub-function code can be used to obtain info about the server status. The sub-function uses the following bitmask:

OPC\_UA\_SERVER\_CONFIG\_STAT\_CONFIG\_CLEAR 0x0001

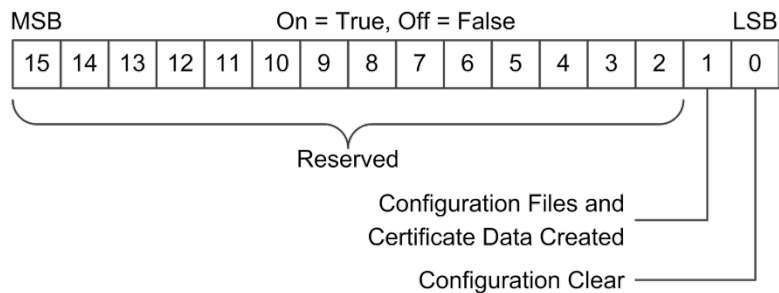
OPC\_UA\_SERVER\_CONFIG\_STAT\_CONFIG\_EXISTS 0x0002

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

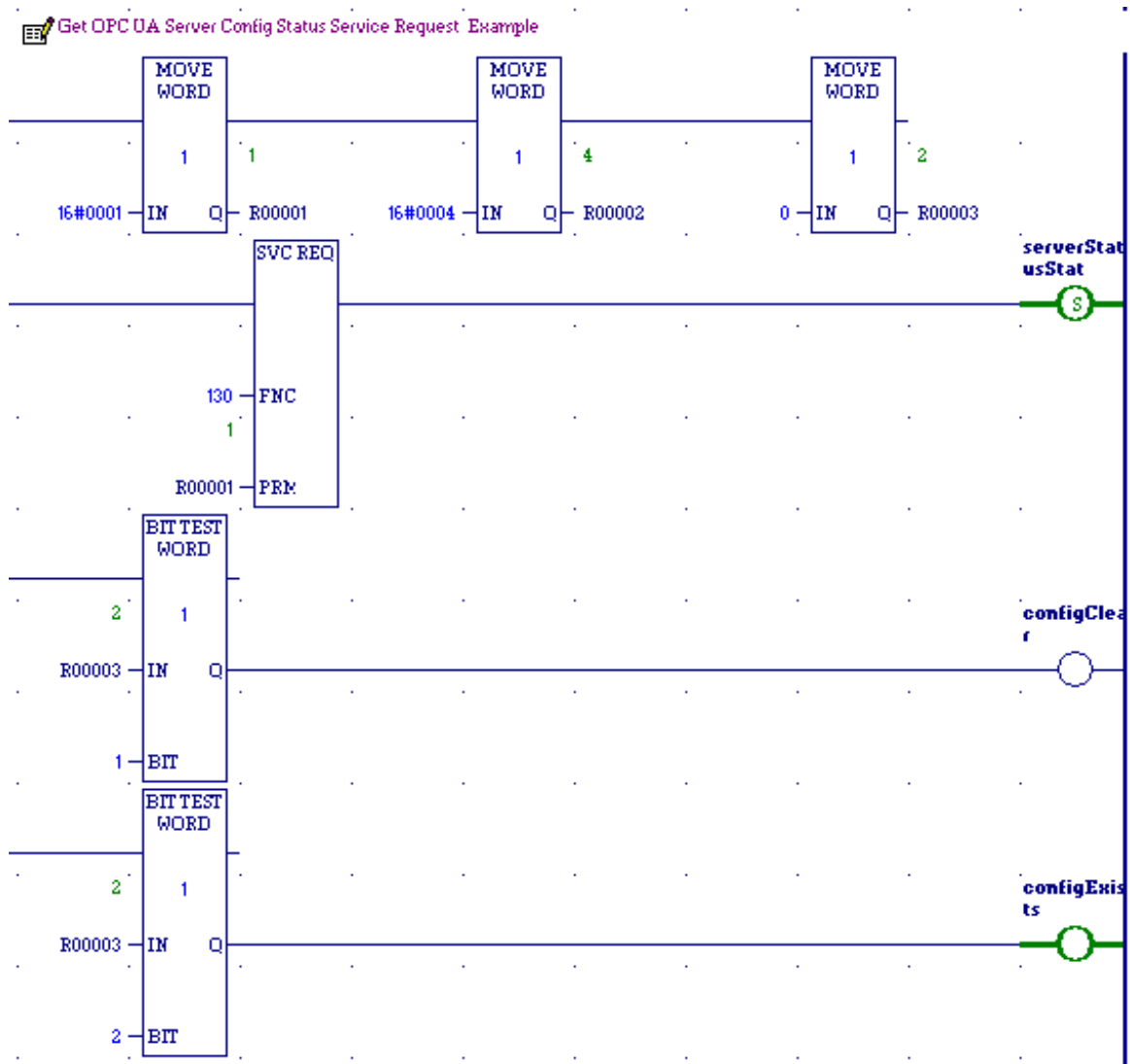
If the SVC\_REQ does not pass power, the operation did not complete.

The CONFIG\_STATUS word bit definitions are displayed below.



**Figure 61: CONFIG\_STATUS Word bit definitions**

**Example of Config Status Request:**



**10.1.2 OPC UA Server Subroutine**

It is recommended that you create a subroutine to encapsulate the service request. The subroutine is then available to the main program to use as necessary. An application note entitled *OPC-UA Server: Application Logic Quick Start Guide* that includes an example subroutine is available at the GE Intelligent Platforms support site, <http://support.ge-ip.com>. An example subroutine call, per the application note, is displayed below for reference.

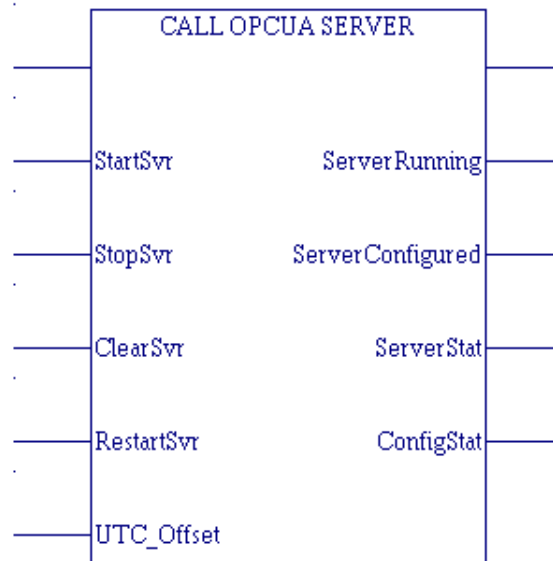


Figure 62: OPC UA Example Subroutine

**Inputs**

Parameter	Description	Data Type
StartSvr	Rising edge starts the OPC UA server. Only allowed if the server is stopped.	Bool
StopSvr	Rising edge stops the OPC UA server. Only allowed is the server is running.	Bool
ClearSvr	Rising edge clears the OPC UA configuration files and certificates. Only allowed when server is stopped (see section 10.1.14 for additional information concerning certificates).	Bool
RestartSvr	Rising edge stops and restarts the OPC UA Server. Only allowed if the server is running	Bool
UTC_Offset	Time offset in minutes, difference between the controller time and universal time (UTC). Must be set before starting or restarting the server. <b>Example 1:</b> New York, USA = UTC - 5:00 hrs. → -300 min <b>Example 2:</b> Paris, France = UTC + 1:00 hr → +60 min	Int

### Outputs

Parameter	Description	Data Type
ServerRunning	True when OPC UA server is running and ready for clients to connect and exchange data.	Bool
ServerConfiguration	True when OPC UA configuration files and certificates have been created.	Bool
ServerStatus	OPC UA Server Status: 0x01 – Server is ready to start 0x02 – Server is running 0x04 – Restart is pending	16-bit bitfield (identical to SERVER_STATUS word defined in OPC UA Server – Service Request – SERVER_STATUS, above)
ConfigStatus	OPC UA Server Configuration Status: 0x01- Configuration is clear 0x02- Configuration files and certificate data have been created.	16-bit bitfield (identical to CONFIG_STATUS word defined in OPC UA Server – Service Request – CONFIG_STATUS, above)

### 10.1.3 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 below.

`opc.tcp://<RXi IP address>:4840`

Specifies the  
OPC UA  
Binary Protocol

RXi Server  
Port Number

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 Machine Edition 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**

The screenshot displays the Project Inspector window with the Ethernet configuration for a GE IP Controller. The left pane shows the hardware tree, and the right pane shows the network parameters. The Inspector table below provides detailed configuration data.

Parameters	Value
Configuration Mode	TCP/IP
Adapter Name	0.2.0
IP Address	10.10.1.102
Subnet Mask	255.255.0.0
Gateway IP Address	0.0.0.0
Status Address	%I00001
Length	80
I/O Scan Set	1

Type	GE IP Controller
Description	
Documentation Address:	
Family	PACSystems FX3i
Controller Target Name	FX3i_Controller_1
Update Rate (ms)	250
Sweep Time (ms)	Offline
Controller Status	Offline
Scheduling Mode	Normal
Force Compact PVT	True
Enable Shared Variables	False
DLB Heartbeat (ms)	1000
Enhanced Security	False
Physical Port	ETHERNET
IP Address	10.10.0.1
Additional Configuration	

Feedback Zone: Disconnecting...  
Disconnected from the device

Figure 63: Project Inspector/Ethernet Config Window

**Note:** In the figure above, 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).

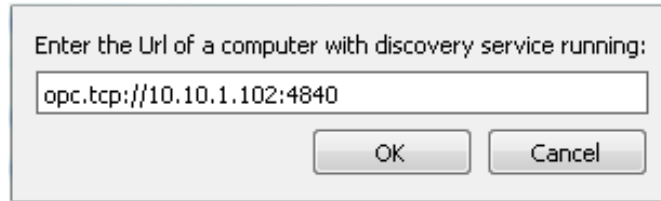


Figure 64: OPC UA Server Client Connection String

We can then connect to the OPC UA server.

**Note:** The Client can see the Controller Target Name when connecting to the server.

The Controller Target Name is set within Machine Edition and is displayed in the screenshot above. A sample client connection can be seen below.

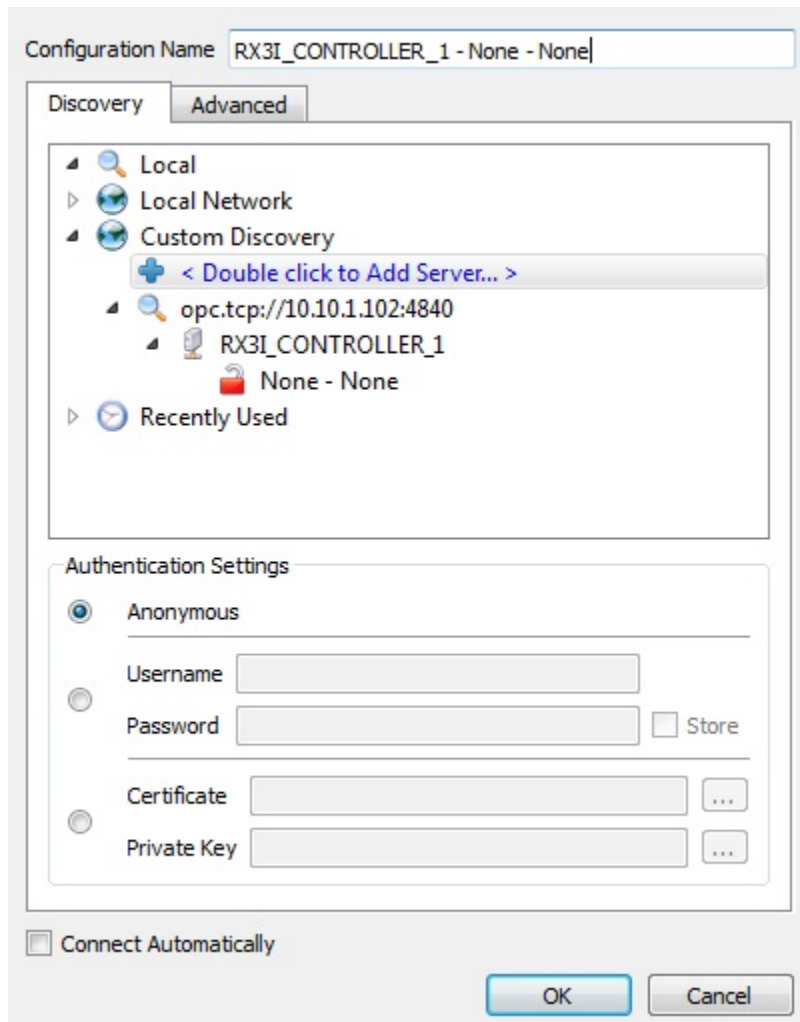


Figure 65: OPC UA Client Connection Dialog

### 10.1.4 OPC UA Client Authentication Settings

OPC UA provides three authentication methods to logon to a server:

- Anonymous,
- Username/Password, and
- Certificate-based.

The OPC UA server supports Anonymous and Username/Password Authentication methods. Machine Edition controller project settings determine the Authentication method used by the OPC UA server.

### 10.1.5 Anonymous Authentication

You enable OPC UA server Anonymous Authentication by disabling Controller passwords. Machine Edition is used to disabled controller passwords. To access this setting using Machine Edition, 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** (see screenshot below).

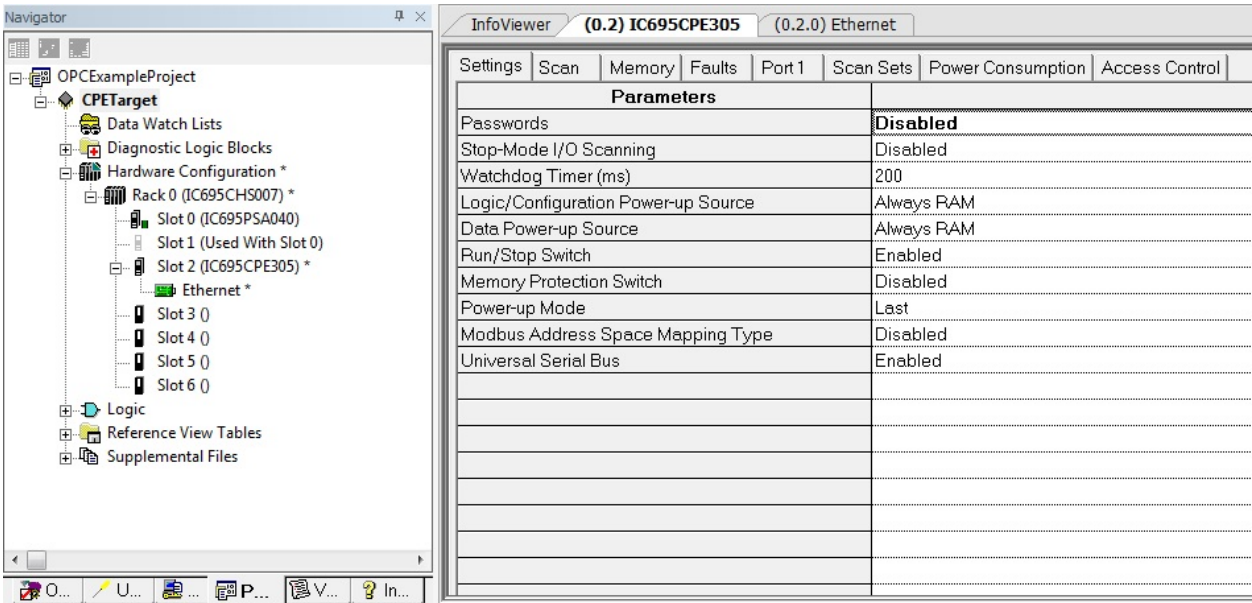


Figure 66: Machine Edition Controller Hardware Configuration – Passwords Disabled

### 10.1.6 Username/Password Authentication

You enable OPC UA server Username/Password Authentication by enabling RXi controller passwords. Machine Edition is used to enable controller passwords. To access this setting using Machine Edition, open the RXi 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 (see screenshot below).

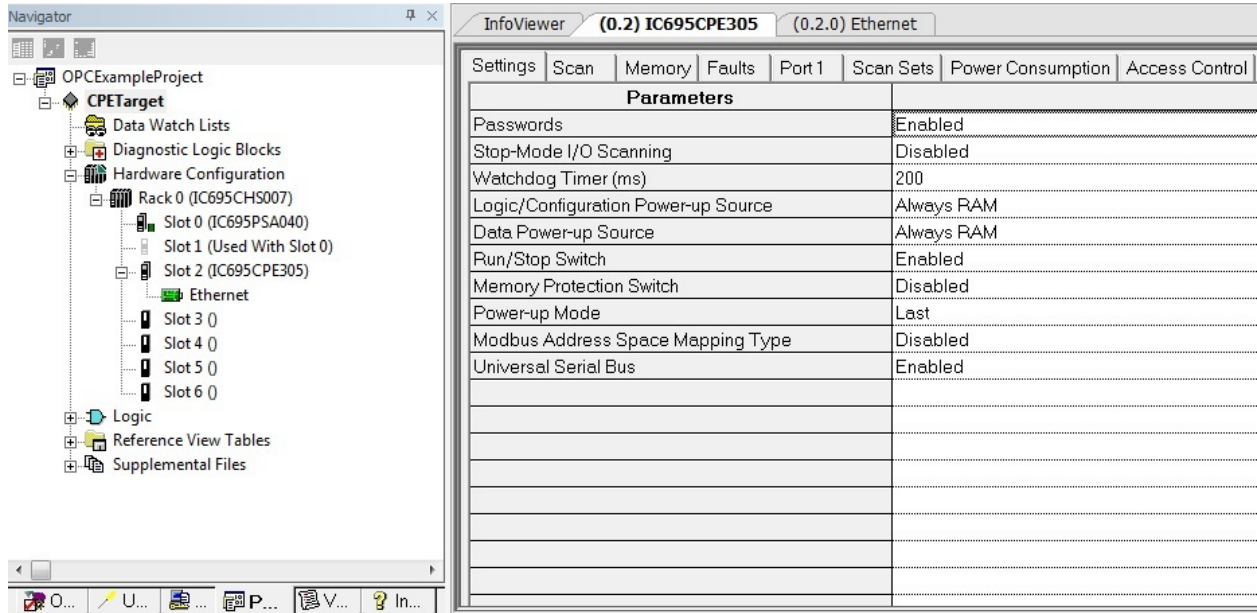


Figure 67: Machine Edition Controller Hardware Configuration – Passwords Enabled

The OPC UA server password is the same as the controller password. Controller passwords are set using the Machine Edition 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 Machine Edition documentation for additional details regarding setting passwords and the privileges assigned to different levels.



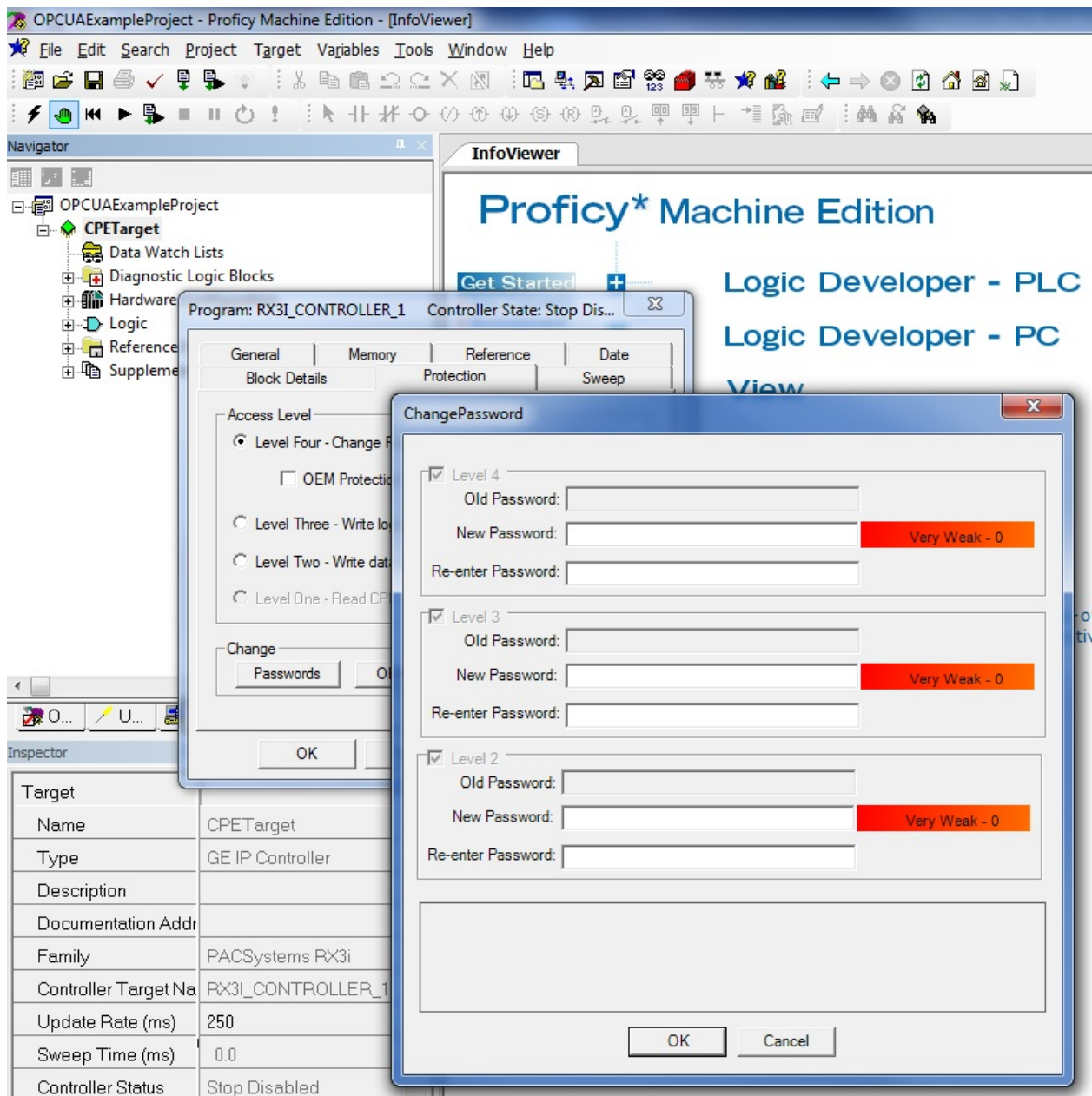


Figure 68: Machine Edition Online Command to Set Passwords

### 10.1.7 OPC UA Security Settings

The OPC UA server does not support message encryption. OPC UA clients typically have two settings for security. The first is the security policy and the second in the Message Security Mode. Both of these settings should be set to **None** for the OPC UA Server Connection (see screenshot below).

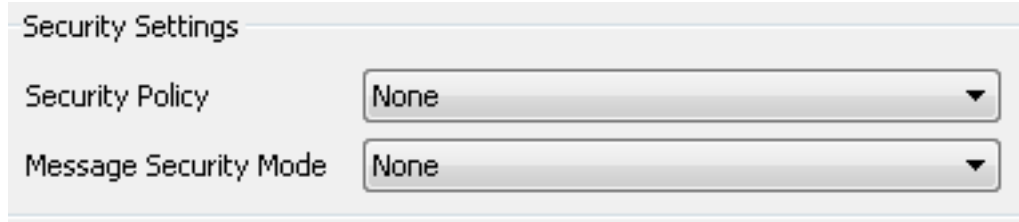


Figure 69: OPC UA Connection Security Settings

### 10.1.8 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.

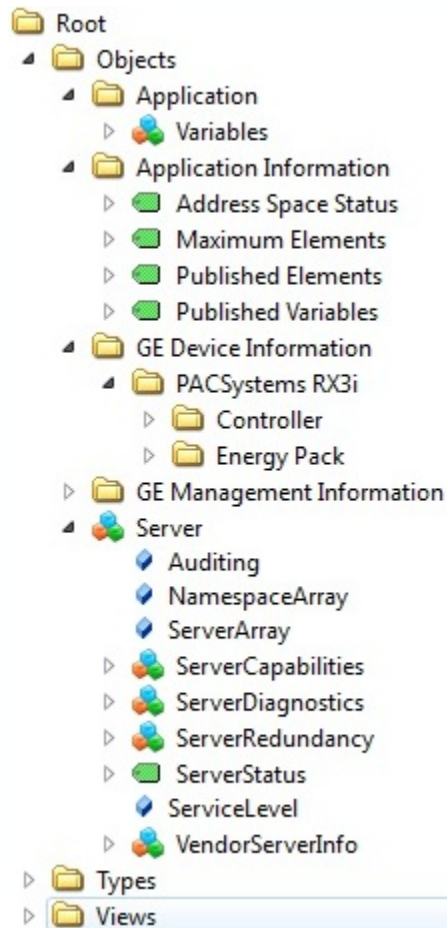


Figure 70: Example OPC UA Address Space

### 10.1.9 Publish Application Variables to OPC UA Address Space

Machine Edition allows you to select application variables to include in the OPC UA address space. This is done by means of the variable's publish attribute. The publish attribute is accessed using the variable **Inspector** within Machine Edition. The Machine Edition variable **Inspector** is displayed in the screenshot below for reference.

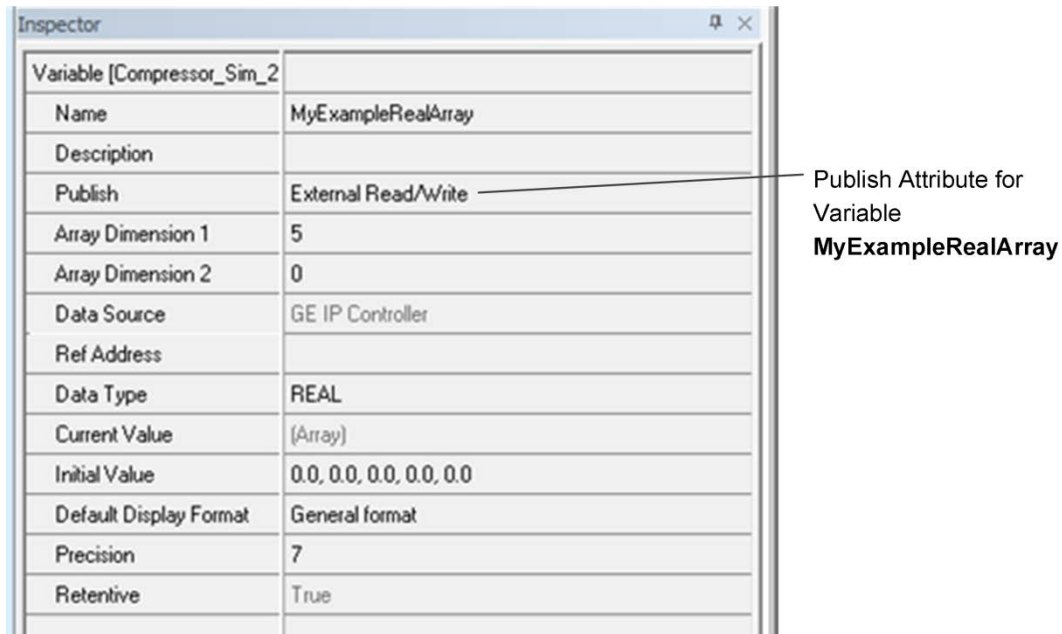


Figure 71: Machine Edition Variable Inspector

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 OPC UA address space supports 250 elements. If more than 250 variables are published, only the first 250 (listed alphabetically) will be added to the OPC UA address space.

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 you. Please reference section **10.1.1, 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.

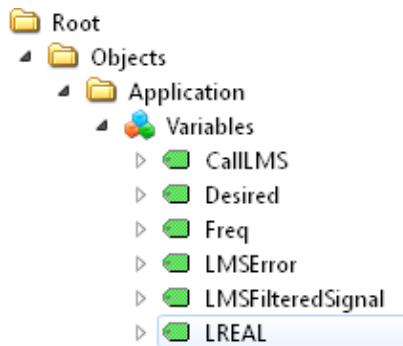


Figure 72: Application Variable Address Space

### 10.1.10 OPC UA Server Information in Address Space

OPC UA servers allow clients to self-discover the OPC UA servers and the 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 at the OPC Foundation website and in its publications.

General Server information is contained under the Server node in the address space (see the following).

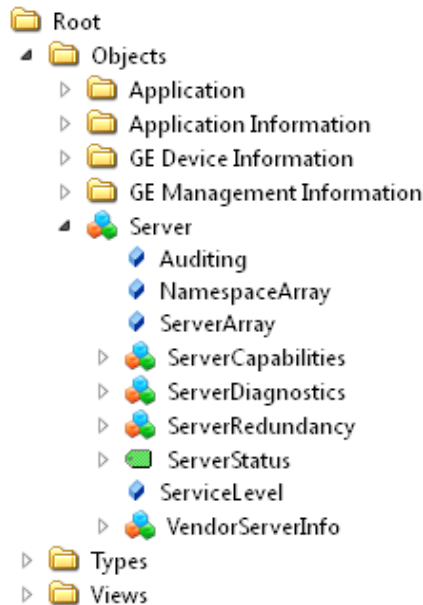


Figure 73: OPC UA Address Space - Server Node

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.

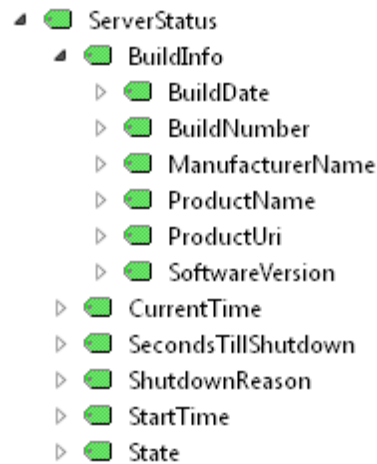


Figure 74: Server Specific Address Space

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).

#	Server	Node Id	Display Name	Value	Datatype
1	RX3i_CONTROLLER_1	NS0 Numeric2...	BuildDate	2014-09-17T08:37:14.000Z	DateTime
2	RX3i_CONTROLLER_1	NS0 Numeric2...	BuildNumber	E476	String
3	RX3i_CONTROLLER_1	NS0 Numeric2...	ManufacturerN...	GE-IP	String
4	RX3i_CONTROLLER_1	NS0 Numeric2...	ProductName	PACSystems RX3i	String
5	RX3i_CONTROLLER_1	NS0 Numeric2...	ProductUri	http://qe-ip.com/PACSystems/RX3i/OPCUAServer	String
6	RX3i_CONTROLLER_1	NS0 Numeric2...	SoftwareVersion	8.20	String

Figure 75: BuildInfo Subscription

### 10.1.11 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).

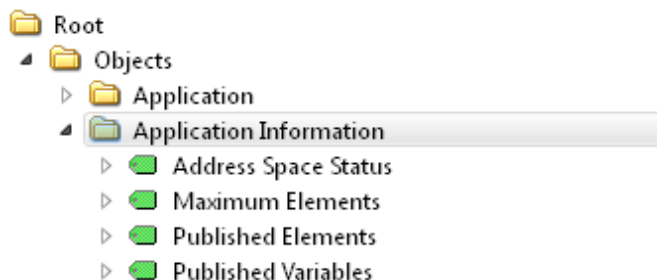


Figure 76: OPC UA Address Space - Application Information

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 element Application Variable (Array): Number Elements is the Array Dimension
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.

An example is displayed below.

Display Name	Value	Datatype
Address Space Status	All Elements Published to Address Space	String
Maximum Elements	250	Int32
Published Elements	249	Int32
Published Variables	248	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 = 250

Maximum Elements is the maximum number of application elements supported by this controller. In the example above, that limit is 250 application elements.

#### Published Elements = 249

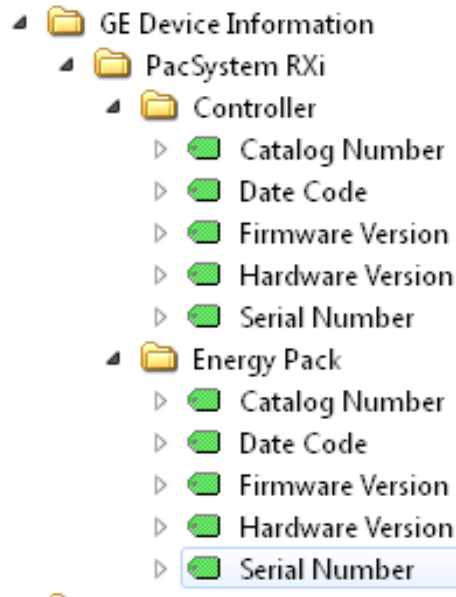
Published Elements is a count of how many application elements are currently being published. In the example above, the number is 249 application elements.

**Published Variables = 248**

**Published Variables** is a count of the controller application variables currently being published. In the example, the number is 248. Note that the **Published Variables** = 248, while **Published Elements** = 249. The difference is due to one of the application variables being a two-dimensional array.

**10.1.12 OPC UA Server – GE Device Information**

The OPC UA server publishes controller specific information under the GE Device Information node.



**Figure 77: OPC UA Address Space – GE Device Information**

The tree structure allows you 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	Device Catalog Number
Date Code	Device Date Code
Firmware Version	Firmware Version installed on device
Hardware Version	Hardware Version of the device
Serial Number	Device Serial Number

**Note:** If the controller does not have an Energy Pack installed, the values for these variables are NA.

### 10.1.13 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 variable's publish attribute requires that the server be restarted.

The OPC UA server automatically restarts when you change 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 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 a 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.

### 10.1.14 OPC UA Server Certificates

OPC UA client/server connections exchange digital certificates during the connection process. The OPC UA server generates a self-signed certificate for the connection process. The OPC UA certificate includes application-specific information within the certificate. The application-specific information includes the Target Name and the controller's TCP/IP address. Thus, if you change this information, the server certificate will not contain this new information. This may cause certain clients to either not connect and/or generate warning messages concerning the conflicts between the running OPC UA server and the information contained within the server certificate. If this information changes, the certificates should be cleared and regenerated.

The OPC UA server certificates are stored internally on the controller's non-volatile storage and are retained through power cycles, clearing of memory and configuration from the programmer, and clearing of flash storage from the programmer. The CONFIG\_STATUS service request returns a bitmask to indicate if the certificates exist on the target, or if they are currently cleared.

If the OPC UA server is started with the certificates cleared, new certificates are generated during startup of the OPC UA server. 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.

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. When a CLEAR is used to clear the certificates, the certificates are permanently deleted and cannot be restored. Once this occurs, new certificates must be generated. The CLEAR service request will not pass power if it is performed with the OPC UA server running.

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.



## Chapter 11 *RX7i PLC Monitoring Via the Web*

---

The PACSystems RX7i embedded CPU Ethernet Interface provides PLC data monitoring using a standard Web browser. Rack-based Ethernet modules and the RX3i embedded Ethernet interface do not support web server operation.

You can use the Web server to monitor the following PLC data:

- **PLC reference tables.** This data is a snapshot of the PLC Reference Tables when the data is displayed in the Browser and is not updated until you request another display. All reference tables are supported.
- **PLC and I/O Fault Tables.**

The web server cannot be used to modify PLC data (acknowledge alarms, set/force values in tables).

The maximum number of web server connections that can be configured for the Ethernet Interface is 16. If the system includes FTP server connections, fewer web server connections are available, as explained in Chapter 4.

### **11.1 System Requirements**

Web monitoring requires version 4.0 or later of Netscape Navigator or Internet Explorer. The browser must be capable of running the Java Virtual Machine (JVM) version 1.3 plug-in. The supported host operating systems are Windows NT 4.0 SP5 or SP6, Windows 95B, Windows 98 (First Edition Service Pack 1, Second Edition), and Windows 2000 Professional SP1, Windows Millennium Edition, Windows XP and Windows CE 3.0. To view the entire Reference Table page, the screen resolution must be 1024 x 768 or higher. Local web firewall blocking issues will be avoided by using HTTP protocol on port 80 to transfer standard HTML files including JavaScript and Java applets from the server to the browser and HTTP Post command to transfer form information from the browser to the server.

### **11.2 Disabling Pop-up Blocking**

Most internet browsers provided a feature that blocks pop-up windows. This prevents the viewing reference tables. Change your browser settings to permit pop-ups.

### **11.3 Web Server Operation in a Redundant System**

In a redundant 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 backup, it takes down all Web Server connections and their underlying TCP connections. The Web Server maintains its underlying TCP connection only long enough to process each web page request; a new TCP connection is opened, used, and closed for each subsequent web page display or update. Unless a web page change or update is requested during a redundancy switch, the operation of the Redundant IP address is transparent to the Web browser. Any web page request in process over the Redundant IP when a role switch occurs is ended.

Although both the active and backup units respond to Web server requests received at the direct IP address, having a remote (host) browser issue Web Server requests to the direct IP address is not recommended. Remote web browsers should use the Redundant IP address when communicating to a Redundant System.

### **11.4 Standard Web Pages**

The CPU Ethernet Interface is shipped with a set of standard PLC web pages already installed. These standard web pages include a PLC home page, a Reference Table display page, a PLC Fault Table display page, and an I/O Fault Table display page. When necessary, new or revised web page files may be transferred into the Ethernet Interface via the standard FTP protocol, as described later.

### 11.4.1 RX7i Home Page

The RX7i home page is displayed after entering the PLC CPU's URL or IP address at your web browser. From the PLC home page, you can navigate to the other PLC web pages.

### 11.4.2 Factory Default Web Page

If the PLC home page file (index.htm) is not present in the Ethernet Interface file system, the web server instead displays the factory default web page.

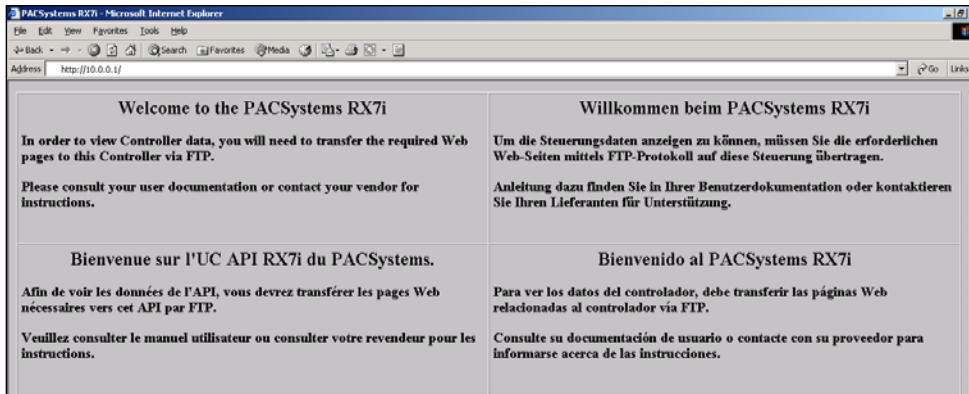


Figure 78: PACSystems Factory Default Web Page

The default web page is displayed in English, French, German and Spanish if the browser is configured to use Western European encoding.

### 11.4.3 Reference Tables Viewer Page

The Reference Tables Viewer page shows the current states of a range of data references. This data is a snapshot of the PLC Reference Tables when the data was initially requested. It is NOT updated until you refresh the display. All RX7i reference tables are available.

#### Selecting Reference Table Data

Initially, the previously-viewed reference table is displayed. To change the display, you can:

Select Reference Table Data Row-by-Row: The right column of each row contains the configuration options for that row. For each row, select the reference table, starting address, and data format. You can select the %R, %AI, %AQ, %I, %Q, %M, %T, %G, %S, %SA, %SB, %SC, %P, %L, or %W reference table. For %P and %L memory types, specify the logic program name, and for %L memory, the subroutine block name. The logic program and subroutine block names must be reentered when defining other rows.

To select the data format, click on a reference table address cell above the reference value and select the display format type. For example:

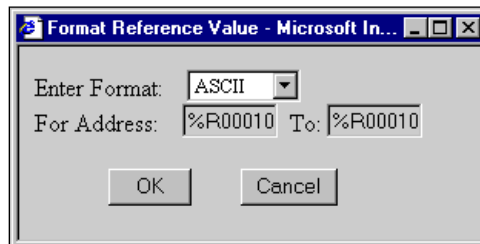


Figure 79: Selecting Display Format

To format a row, click the Format button for the entire row. Use the drop down box to select the data format for the selected reference address or row. With Internet Explorer, pressing the "OK" button changes the format immediately. With Netscape, the format changes after you refresh the screen.

## Saving Reference Table Settings

You can save up to 10 previously formatted reference table views on the computer being used to view the data. To save the current reference table settings, go to the section at the bottom of the page labeled 'Save Current Table Settings To:'. From the drop-down box, select a number to assign to these settings. Optionally, enter a description of the table settings by typing into the text box labeled 'Enter Description'. Click on the 'Save' button to save the reference table settings to the computer.

## Display Formats

**Binary:** uses 1s and 0s to represent the bits in a byte or word of data. If a discrete bit is overridden for the %I, %Q, %M or %G tables, the bit is underlined.

**+Dec:** signed decimal for one word of data. Valid range is -32768 to +32767.

**Dec:** unsigned decimal for one word of data. Valid range is 0 to 65535.

**Hex:** a four digit hexadecimal value for one word of data. The value has 16# as a prefix (for example 16#4241). Valid range is 16#0000 to 16#FFFF.

**ASCII:** ASCII representation of two 8-bit values. For example, a hex value of 16#4142 appears as "A B". ASCII display requires Internet Explorer 4.0 or Netscape 4.7 or later.

**+DbDecimal:** signed decimal for a double word (32 bits). Valid range is -2,147,483,648 to +2,147,483,647. This format is only available for word type memory (%R, %AI, %AQ, %P, %L, and %W).

**DbDecimal:** unsigned decimal for a double word (32 bits). Valid range is 0 to 4,294,967,295. This format is only available for word type memory (%R, %AI, %AQ, %P, %L, and %W).

**Real:** 7 decimal digits plus a decimal point and exponent if necessary (for example 123.4567, 1.234567e+038). This format uses 2 words or 32 bits. This format is only available for word type memory (%R, %AI, %AQ, %P, %L, and %W). The range is +-1.401298e-045 to +-3.402823e+038.

**Blank:** The associated cell or row will have no value or reference address header.

### 11.4.4 PLC Fault Table Viewer Page

The PLC Fault Table Viewer displays the contents of the PLC fault table.

The PLC name is shown at the top of the page, together with the PLC timestamp showing when the page was accessed or refreshed.

The PLC fault table provides up to 16 entries arranged from newest to oldest. If there are fewer than 16 entries, the remaining rows are blank. If there are more than 16 faults, the table displays the most recent faults in the first 8 rows and the oldest faults in the last 8 rows.

To change the format of the fault extra data, select the appropriate checkbox at the top of the page.

To refresh the fault data, click the 'Refresh PLC Fault Table' button.

When using Internet Explorer, the fault extra data can be viewed by using the mouse to highlight a particular fault and then clicking on the fault. This is shown below:

View Reference Tables
Refresh PLC Fault Table
View I/O Fault Table

Total Faults	20
Total Faults Displayed	16
Entries Overflowed	4
Table Last Cleared (MM-DD-YY HH:MM:SS)	01-01-2000 00:00:42
PLC Date/Time During Last Update (MM-DD-YY HH:MM:SS)	10-07-2003 12:18:07

Fault Extra Data	Fault Extra Data Format
<input type="checkbox"/> Show All	<input checked="" type="radio"/> Byte
	<input type="radio"/> Word
	<input type="radio"/> ASCII

PLC Fault Table																	
Location	Fault Description	Date/Time															
0.11	LAN transceiver fault; OFF network until fixed	04-10-2000 / 19:17:33.000															
	<table border="1" style="width: 100%; border-collapse: collapse; font-size: small;"> <tr> <th style="width: 20%;">Long/Short</th> <th style="width: 20%;">Task</th> <th style="width: 20%;">Group</th> <th style="width: 20%;">Action</th> <th style="width: 20%;">Error Code</th> </tr> <tr> <td>1</td> <td>6</td> <td>14</td> <td>2</td> <td>454</td> </tr> <tr> <td colspan="5">Fault Extra Data: 20 00 01 80 80 00 00 00 01 02 90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td> </tr> </table>	Long/Short	Task	Group	Action	Error Code	1	6	14	2	454	Fault Extra Data: 20 00 01 80 80 00 00 00 01 02 90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00					
Long/Short	Task	Group	Action	Error Code													
1	6	14	2	454													
Fault Extra Data: 20 00 01 80 80 00 00 00 01 02 90 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00																	
0.11	LAN transceiver fault; OFF network until fixed	04-10-2000 / 19:16:12.000															
0.11	LAN transceiver fault; OFF network until fixed	04-10-2000 / 18:59:19.000															
0.11	LAN transceiver fault; OFF network until fixed	03-29-2000 / 18:37:04.000															
0.11	LAN transceiver fault; OFF network until fixed	03-20-2000 / 21:28:20.000															
0.1	LAN transceiver fault; OFF network until fixed	03-20-2000 / 21:28:20.000															
0.1	LAN system-software fault; resuming	01-03-2000 / 20:09:37.000															
0.1	LAN system-software fault; resuming	01-03-2000 / 19:44:35.000															
0.11	LAN transceiver fault; OFF network until fixed	01-01-2000 / 02:03:32.000															
0.1	No user program present	01-01-2000 / 02:03:32.000															
0.11	LAN transceiver fault; OFF network until fixed	01-01-2000 / 02:03:09.000															
0.1	No user program present	01-01-2000 / 02:03:09.000															
0.11	LAN transceiver fault; OFF network until fixed	01-01-2000 / 02:02:44.000															
0.1	No user program present	01-01-2000 / 02:02:44.000															
0.11	LAN transceiver fault; OFF network until fixed	01-01-2000 / 00:06:46.000															
0.1	No user program present	01-01-2000 / 00:06:46.000															

Figure 80: PLC Fault Table Display

The fault extra data can be displayed in byte, word or ASCII format depending on which button is selected at the top of the screen. These selections affect the display of all fault extra data. If an error code does not have a string associated with it, the "Fault Description" field is blank.

To view the fault extra data for all faults, select the “Show All” checkbox as shown below:

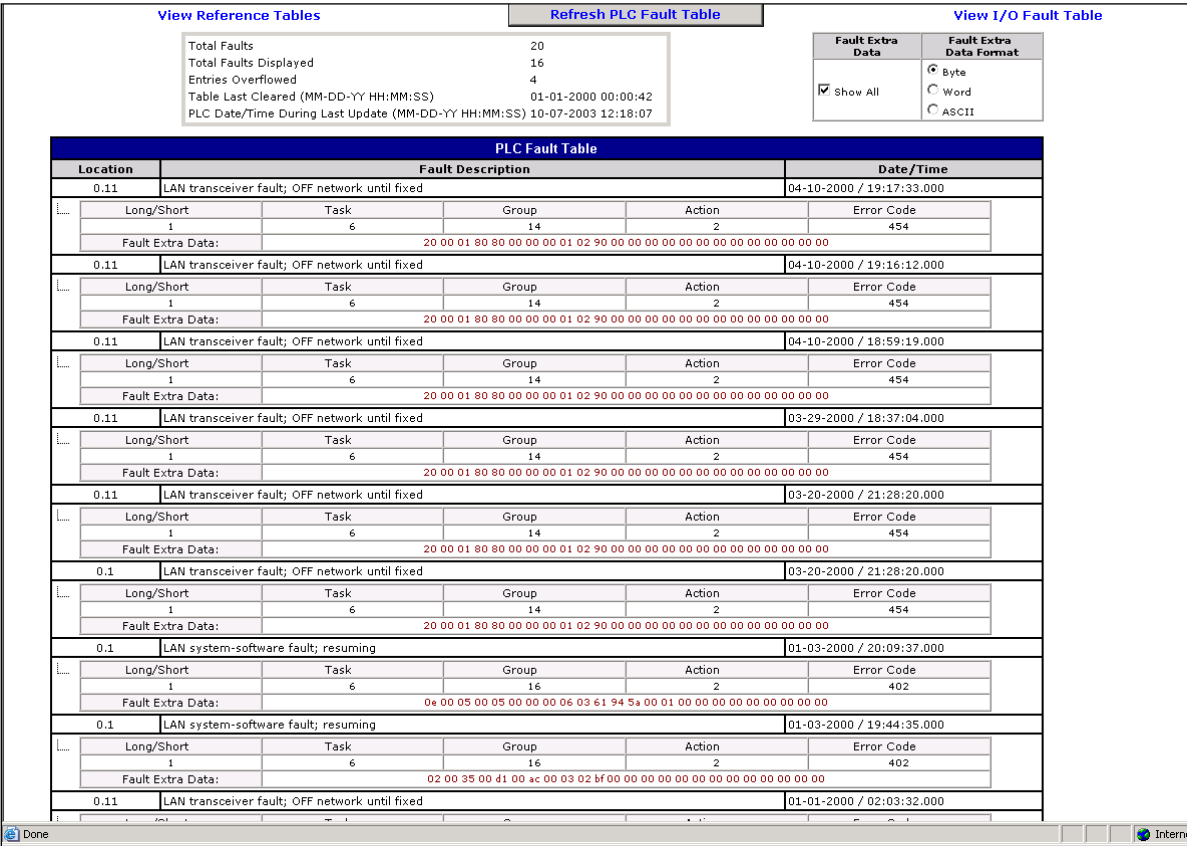


Figure 81: Fault Extra Data Display

For Netscape, first check the “Show All” checkbox and press the “Refresh PLC Fault Table” button. This will show the fault extra data for all faults. Netscape cannot show fault extra data for selected faults. To hide the fault extra data, uncheck the “Show All” checkbox and again press the “Refresh PLC Fault Table” button.

### 11.4.5 I/O Fault Table Viewer Page

The I/O Fault Table web viewer page displays the contents of the I/O Fault Table:

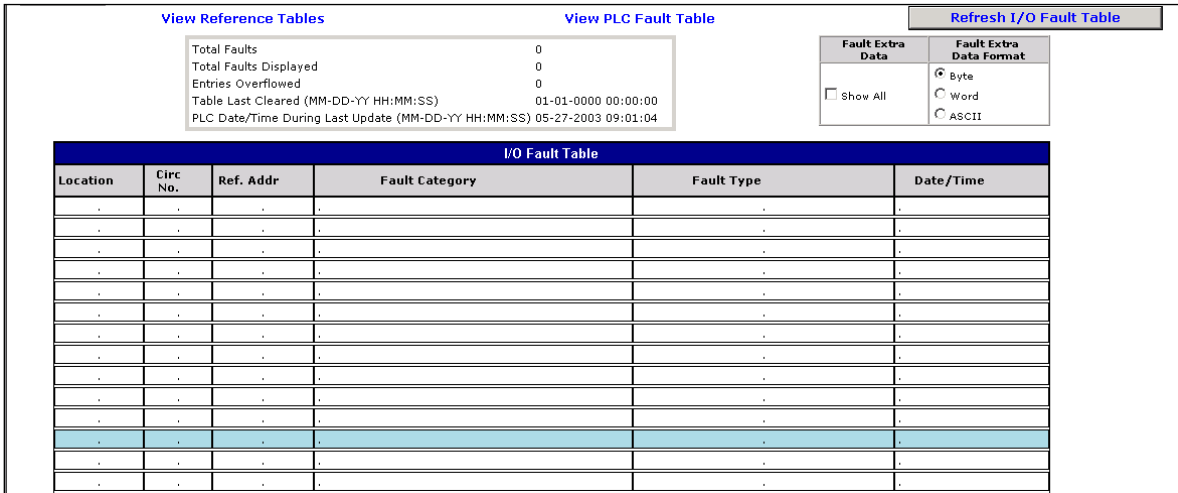


Figure 82: I/O Fault Table Display

The fault extra data can be shown or hidden by clicking on a fault. The fault extra data for all faults in the table can be displayed by selecting the checkbox at the top of the page labeled 'Fault Extra Data'. To change the format of the fault extra data, select the appropriate checkbox at the top of the page.

To refresh the fault data, click the 'Refresh I/O Fault Table' button.

## 11.5 Downloading PLC Web Pages

To add new or revised web page files or support files, you will need to transfer the appropriate files to the Ethernet Interface via FTP. Once the new web files have been obtained, they are copied into the local computer from which the FTP utility will be run. A general procedure for transferring web files via Windows FTP is described below. (You may also use a commercial FTP program.)

**Note:** You may not be able to open an FTP connection when the CPU is in Run mode and the level of Ethernet traffic is medium to heavy. If the network traffic is high, it is recommended that you reduce the network traffic before trying to create an FTP connection.

### 11.5.1 FTP Connect and Login

You can either use a commercial FTP tool or use the "ftp" command on the DOS Prompt or Command line (Note: Not all FTP tools will be guaranteed to work since the server only supports a limited set of FTP commands).

From the Windows DOS box command line interface, enter "ftp" followed by the URL or IP address of the PLC as shown below:

**ftp <URL or IP address of the Ethernet Interface>**

You will then be prompted for a login name and password as shown below. The default FTP password is "system".

**login: user**

**password: system**

The FTP server in the PLC Ethernet interface does not support multiple levels of login (there are no distinct 'anon' or 'user' logins). Once successfully logged on, you can execute any of the FTP commands described below; this login is required in order to store web page files to the Ethernet Interface.

## 11.5.2 Changing the Password

The default FTP password is "system". You can change the FTP password via a parameter in the AUP file, which is stored to the PLC via the programmer, or by using the Station Manager.

### Changing the Password from the Advanced User Parameters File

The following line should be added to the AUP file to change the FTP password (for example, to "my\_ftp\_pw"):

```
tpassword = my_ftp_pw
```

### Changing the Password from the Station Manager

In addition, you can change the FTP password (for example to "my\_ftp\_pw") using the following Station Manager command:

```
= CHPARM tpassword my_ftp_pw
```

The FTP password can be up to 10 characters long and uses the same character set listed for the reference viewer password described later in this document. These passwords are not case sensitive.

Arguments for Station Manager CHPARM command must be enclosed in double quotes to preserve the capitalization of the argument. However since these passwords are case insensitive, the double quotes are not required.

**Note:** The CHPARM command is not available if the PLC has received a valid configuration from the Programmer.

## 11.5.3 Web Page File Transfer

After logging into the PLC's FTP server, web page files can be copied from the PC to the PLC through the following steps:

1. Set the FTP file transfer type to binary by typing in "binary"
2. For each file, change to the desired directory if appropriate by typing "cd ./subdirectory". Then transfer the file using the "put" command by typing: "put filename.htm"
3. Verify all files are properly transferred by typing in: "dir" or "ls". This returns a list of the files located at the current directory on the PLC Ethernet Interface
4. Quit the FTP session by typing in "quit" or "bye".

If you copy a file that already exists in the module, the new file overwrites the existing file without warning. One of the files stored will be a fault string file that will be specific for each language supported.

The PLC FTP server also supports the following standard FTP commands:

- "get" command - allows the user to transfer files from the PLC web server to their local PC (for example "get filename1.htm").
- "delete" command - allows user to delete web pages from the server (for example "delete filename1.htm").

## 11.6 Viewing the RX7i PLC Web Pages

Each web browser (HTTP) instance (i.e., each browse window) requires at least two TCP connections and each FTP session requires two TCP connections to the PLC. The maximum number of web browser connections and FTP connections at the Ethernet interface at any one time are separately configurable from 0 to 16 (a value of 0 means that the web server or FTP capability is disabled). The total number of configured web browser connections plus FTP connections is limited to 16 connections; once the number of browser/FTP connections reaches the configurable limit, any new browser or FTP connection requests will fail.

The number of Web Server and FTP connections is configurable via the Programmer. The Programmer configuration details are described in the Programmer Help utility.

When the PLC is unconfigured, the user can change the number of web server (HTTP) connections and FTP connections with the following Station Manager commands, respectively:

```
CHSOSW web_max_conn <number from 0-16>
```

```
CHSOSW ftp_max_conn <number from 0-16>
```

As noted in the Ethernet Configuration section, the sum of web server connections plus FTP connections must not exceed 16 connections.

For example:

```
= CHSOSW web_max_conn 6
```

```
= CHSOSW ftp_max_conn 4
```

**Note:** The CHSOSW commands are not available if the PLC has received a valid configuration from the Programmer.



## Chapter 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
- EOK 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 marked on the product.
  - PLC CPU version number from CME Status screen
  - Ethernet Interface CPU Embedded or standalone
- Description of symptoms of 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 Proficy 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 GFK-2225, *TCP/IP Ethernet Communications for the PACSystems Station Manager Manual*, 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 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 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 GFK-2225, *TCP/IP Ethernet Communications for PACSystems Station Manager Manual*, for information on how to access and use the Station Manager software.

## 12.3 States of the Ethernet Interface (Rack-based and RX7i Embedded Interfaces)

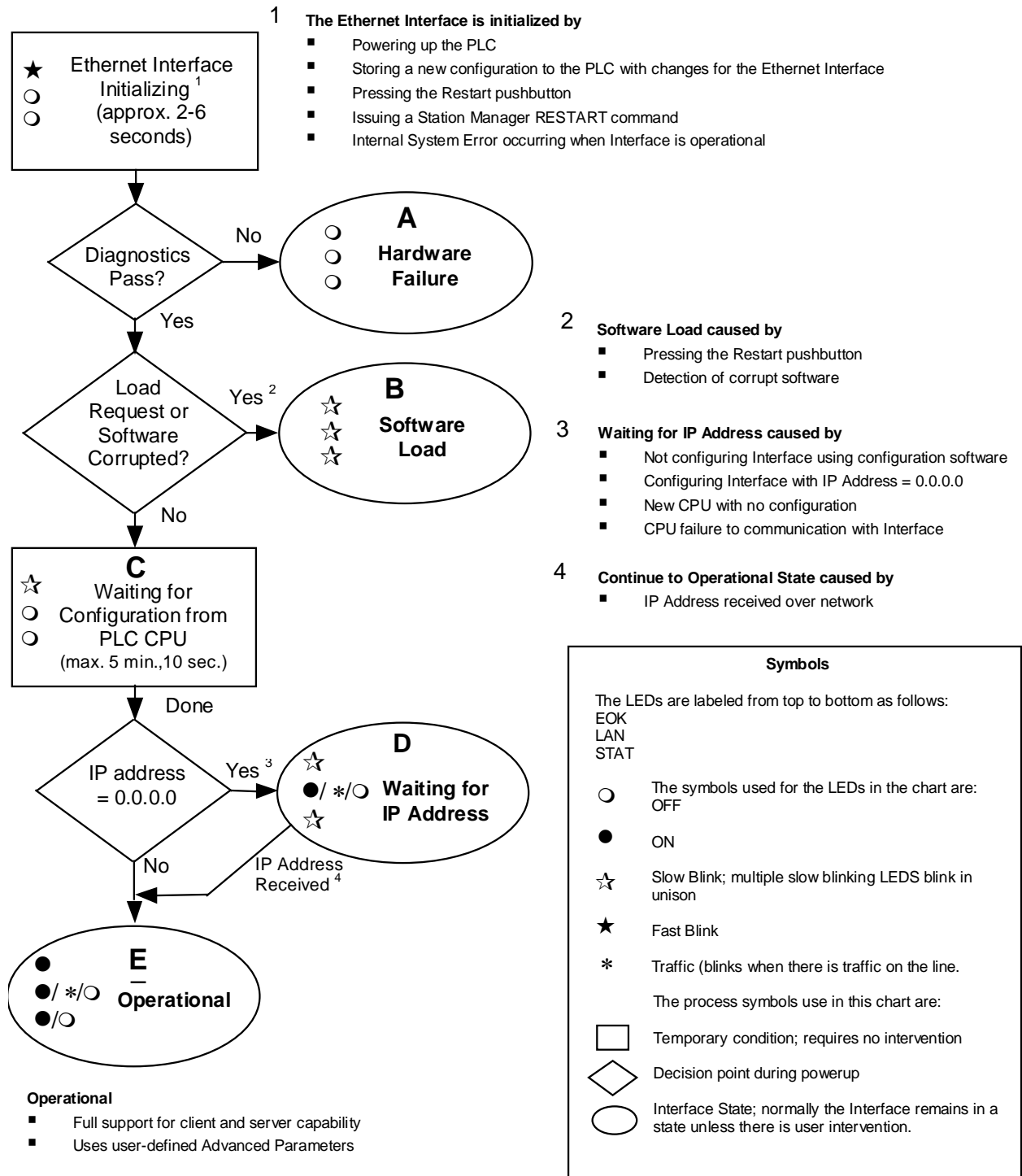


Figure 83: States of the Ethernet Interface

LED Pattern		Where Stopped	Possible Cause	Corrective Actions
<ul style="list-style-type: none"> <li>○</li> <li>○</li> <li>○</li> </ul>	<p>EOK (OFF) LAN (OFF) STAT (OFF)</p>	<p>A</p> <p>Hardware Failure</p>	<p>Fatal Hardware Error.</p>	<p>Make sure the PLC has power. Examine Controller Fault Table for clues. Recheck PLC Programmer configuration. Power off baseplate, inspect the Interface for loose components, reseal the module, and Restart. If the problem persists, replace the PLC hardware.</p>
<ul style="list-style-type: none"> <li>⚙</li> <li>⚙</li> <li>⚙</li> </ul>	<p>EOK (Slow blink) LAN (Slow blink) STAT (Slow blink)</p> <p>All LEDs blink in unison.</p>	<p>B</p> <p>Software Loader</p>	<p>Software corrupt.</p>	<p>Connect a PC Software Loader and load new software.</p>
<ul style="list-style-type: none"> <li>⚙</li> <li>○</li> <li>○</li> </ul>	<p>EOK (Slow blink) LAN (OFF) STAT (OFF)</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>	<p>Use the PLC Programmer configuration software to configure the Interface then store the configuration to the PLC CPU. Power cycle the PLC. Clear faults and Restart Interface.</p>
<ul style="list-style-type: none"> <li>⚙</li> <li>○</li> <li>○</li> </ul>	<p>EOK Blinking error code LAN Off STAT Off</p>		<p>Unrecoverable hardware or runtime failure</p>	<p>See the list of blink codes on the next page.</p>
<ul style="list-style-type: none"> <li>⚙</li> <li>● ⚙ ○</li> <li>⚙</li> </ul>	<p>EOK (Slow blink) LAN (ON/Traffic/OFF) STAT (Slow blink)</p> <p>EOK and STAT blink in unison.</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>	<p>Use the PLC Programmer to configure the Interface with a non-zero IP address. Assign IP address over network</p>
<ul style="list-style-type: none"> <li>●</li> <li>● ⚙ ○</li> <li>● ○</li> </ul>	<p>EOK (ON) LAN (ON/Traffic/OFF) STAT (ON/OFF)</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>	<p>Connect cable. Examine Controller Fault Table to find out why the STAT LED is OFF.</p>

On the RX7i interfaces, the Ethernet LEDs are labeled **EOK**, **LAN**, and **STAT**.

On the RX3i rack-based Ethernet interfaces, the Ethernet LEDs are labeled **ETHERNET OK**, **LAN OK**, and **LOG EMPTY**, respectively.

## 12.4 EOK LED Blink Codes for Hardware Failures (Rack-based and RX7i Embedded Interfaces)

The EOK LED indicates whether the module is able to perform normal operation. This LED is on for normal operation and flashing for all other operations. If a hardware or unrecoverable runtime failure occurs, the EOK LED blinks a two-digit error code. The EOK 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

<b>Blink Code</b>	<b>Description</b>	<b>Blink Code</b>	<b>Description</b>
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: Data store exception
0x24	Firmware CRC (cyclic redundancy check) error during power up or Factory Test <sup>21</sup>	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	Data store 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		

<sup>21</sup> CRC error or software error during normal operation causes Ethernet restart

## 12.5 Controller Fault Table

Most error conditions involving the Ethernet interface generate faults in the Controller Fault table. The table on the next two pages lists Ethernet interface faults and corrective actions.

To access the details of a Controller Fault Table entry, double-click the Fault Table entry and the details are displayed as “fault extra data”. Refer to Online Help in the PLC programming software for more information.

An example of the fault extra data is shown below:

```
160006000300 05 0000 00 0000 00 00 0000 00 00 0000 00 00 0000 00 00 00
```

**Figure 84: Fault Extra Data Example**

For Ethernet Interfaces the leftmost 14 digits of fault extra data (underlined in the example above) show the corresponding log Events (2 digits) and Entries 2, 3, and 4 (in that order, 4 digits each).

The example above is reporting

- an Event 16,
- Entry 2=6,
- Entry 3=3, and
- Entry 4=5.

This information can be used to refer directly to detailed fault descriptions included in the Exception Log Event tables in GFK-2225, *TCP/IP Ethernet Communications for PACSystems Station Manager Manual*. (In that document, refer to Appendix B, Exception Log Events.)

### 12.5.1 Controller Fault Table Descriptions

<b>Controller Fault</b>	<b>User Action</b>
Backplane communications with controller fault; lost request	Check to make sure that the logic application is not sending COMMREQs faster than the Ethernet Interface can process them. Reduce the rate at which the application is sending COMMREQs to the Ethernet interface. If problem persists, contact Technical Support.
Mailbox queue full – COMMREQ aborted	Check to make sure that the logic application is not sending COMMREQs faster than the Ethernet Interface can process them. Reduce the rate at which the application is sending COMMREQs to the Ethernet interface. If problem persists, contact Technical Support.
Bad local application request; discarded request	Check for valid COMMREQ command code. If problem persists, contact Technical Support.
Bad remote application request; discarded request	Try to validate the operation of the remote node. If problem persists, contact Technical Support.
Can't locate remote node; discarded request	Error reported when message received where IP/MAC address cannot be resolved. Error may indicate that remote host is not operational on the network. Check that remote host is operational on network and its addresses are correct.
Comm_req - Bad task ID programmed	Message from PLC for unknown Ethernet Interface task. Check COMMREQ function block.
Comm_req - Wait mode not allowed	Check COMMREQ to make sure sent in no-wait mode.
Configured gateway address bad; can't talk off local net	Error in configuration. 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 connection to LAN and operation of remote node.

<b>Controller Fault</b>	<b>User Action</b>
LAN controller fault; restart LAN I/F	HW fault, perform a power cycle. If problem persists, contact Technical Support.
LAN controller Tx underflow; attempt recovery	Internal system error. If problem persists, contact Technical Support.
LAN controller under run/overrun; resuming	Internal system error. If problem persists, contact Technical Support.
LAN data memory exhausted - check parameters; resuming	The Ethernet Interface does not have free memory to process communications. If 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 can't init - check parameters; running soft Sw utl	Internal system error. If 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	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 problem persists, contact Technical Support.
LAN severe network problem; attempting recovery	External condition prevented transmission of frame in specified time. Could be 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 problem persists, contact Technical Support.
LAN system-software fault; restarted LAN I/F	Internal system error. If problem persists, contact Technical Support.
LAN system-software fault; resuming	Internal system error. If 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 SQE test switch if present on transceiver.
Local request to send was rejected; discarded request	Internal error. Check that the Ethernet Interface is online. If problem persists, contact Technical Support.
Memory backup fault; may lose configuration/log on restart	Internal error accessing non-volatile device. If 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 Comm_Req; discarded	COMMREQ received when Ethernet Interface cannot process COMMREQ. Make sure Ethernet Interface is configured and online. 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 configuration	PLC firmware does not support Ethernet communications software or 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.

<b>Controller Fault</b>	<b>User Action</b>
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 - Comm_req 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 initiation of a COMM\_REQ 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 Bits	Description	
	Rack-based and RX7i Embedded	RX3i Embedded
1	Port 1A full-duplex	Port full-duplex
2	Port 1A 100Mbps	Port operating at highest supported speed
3	Port 1B full-duplex	Reserved
4	Port 1B 100 Mbps	Reserved
5	Network Time Locked	Reserved
6	Redundant IP address is active	Reserved
7-8	Reserved	Reserved
9	Any Channel Error (error on any channel)	Any Channel Error (error on any channel)
10-12	Reserved	Reserved
13	LAN OK	LAN OK
14	Resource problem	Resource problem
15	Module Overtemp (RX3i rack-based only)	Reserved
16	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
18	Channel 1: Modbus TCP Client - Reserved SRTP Client - Channel Error	Channel 1: Modbus TCP Client - Reserved SRTP Client - Channel Error
...	...	...
47	Channel 16 Status SRTP: Data Transfer Modbus TCP Client: Channel Open	Channel 16 Status SRTP: Data Transfer Modbus TCP Client: Channel Open
48	Channel 16: Modbus TCP Client - Reserved SRTP Client - Channel Error	Channel 16: Modbus TCP Client - Reserved SRTP Client - Channel Error
49-78	Channels 17-31	Reserved
79	Channel 32 Status SRTP: Data Transfer Modbus TCP Client: Channel Open	Reserved
80	Channel 32: Modbus TCP Client - Reserved SRTP Client - Channel Error	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 (Rack-based and RX7i Embedded) Port Full-duplex (RX3i Embedded)**

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 100Mbps (Rack-based and RX7i Embedded) Port Operating at Highest Supported Speed (RX3i Embedded)**

This bit is set to 1 when the port is operating at its highest supported speed.

### **Bit 3, Port 1B Full-duplex (Rack-based and RX7i Embedded)**

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 4, Port 1B 100Mbps (Rack-based and RX7i Embedded)**

This bit is set to 1 when Port 1B is operating at 100Mbps.

### **Bit 5, Network Time Locked (Rack-based and RX7i 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 timeserver, 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 "Timestamping of Ethernet Global Data Exchanges" in Chapter 5.

### **Bit 6, Redundant IP Address Active (Rack-based and RX7i Embedded)**

This bit is set to 1 when the configured Redundant IP address is active. Otherwise this status bit is set to 0.

### **Bit 9, Any Channel Error (All models)**

This bit (normally 0) indicates one or more of the channels are in error.

### **Bit 13, LAN OK (All models)**

This bit is 1 as long as the Ethernet Interface software is able to 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. In addition, the Station Manager STAT B and LOG commands can be used. See the *Station Manager 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, Connection Open Bit (Rack-based and RX7i Embedded)**

**Bits 17, 19, 21 ... 47, Connection Open Bit (RX3i 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, 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, Data Transfer Bit (Rack-based and RX7i Embedded)**

**Bits 17, 19, 21 ... 47, Data Transfer Bit (RX3i Embedded)** Typically, a channel is used to perform repetitive reads or writes. The Data Transfer bit pulses (0 to 1) to indicate a read or write. This can be an indicator to 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 completed does not guarantee that the next transfer has not begun or completed.

After an Establish Channel command, the COMM\_REQ 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, Channel Error Bit (Rack-based and RX7i Embedded)**

**Bits 18, 20, 22 ... 48, Channel Error Bit (RX3i Embedded)**

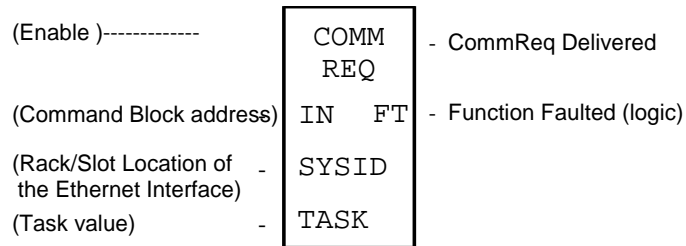
This bit (normally 0) is the primary indicator for an error on a channel. It indicates any channel error, fatal or non-fatal. It does not necessarily indicate that the channel is idle.

A Channel Error bit is not meaningful until the Ethernet Interface has updated the COMM\_REQ status word confirming the Read or Write command for that channel. For an Establish Channel command, the COMM\_REQ 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 COMM\_REQ 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 85: 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

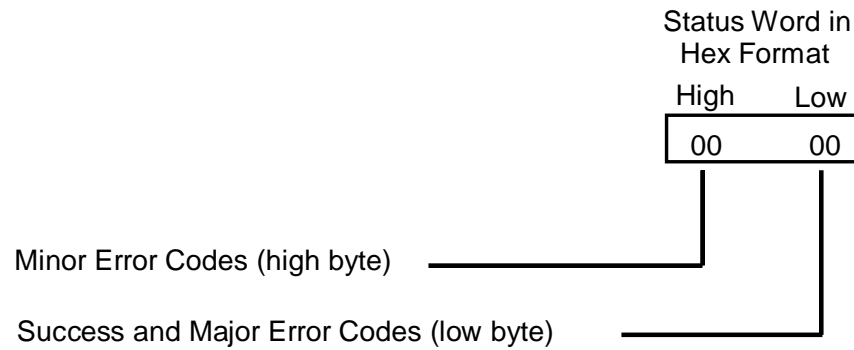
Every COMMREQ Command Block instruction specifies a 1-word memory address to receive status information about the execution of the command.

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 86: 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.

**Note:** The COMMREQ Status words for SNTP Time Transfer commands are listed in “Chapter 5, Ethernet Global Data,” following the COMMREQ command descriptions.

## 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 in lieu 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 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 in lieu 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 traffic type.
C605H	C685H	Control Program (CP) tasks exist but requestor not logged into 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 non-zero buffer size required for block commands.
CA05H	CA85H	Device is write-protected.
CB05H	CB85H	A comm or write verify error occurred during save or restore.
CC05H	CC85H	Data stored on device has been corrupted and is no longer reliable.
CD05H	CD85H	Attempt was made to read a device but no data has been stored on it.
CE05H	CE85H	Specified device has insufficient memory to handle request.
CF05H	CF85H	Specified device is not available in the system (not present).
D105H	D185H	Packet size or total program size does not match 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 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 datagram.
DC05H	DC85H	Unable to find 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 process cannot login.
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 service request.
EE05H	EE85H	Could not return block sizes.
EF05H	EF85H	Programmer is already attached.
F005H	F085H	Request only valid in stop mode.
F105H	F185H	Request only valid from programmer.
F205H	F285H	Invalid program cannot log in.
F405H	F485H	Invalid input parameter in request.
F505H	F585H	Invalid password.
F605H	F685H	Invalid sweep state to set.
F705H	F785H	Required to log in to a task for service.

Error Status (Hexadecimal)		Minor Error Code Description
Remote Server	Client	
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	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 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 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	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 availability of mail communications with the PLC.
2311H	Invalid task ID detected while attempting to send a mailbox message to the PLC.
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.



Error Status (Hex)	SRTM Error Code Description
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	Bad sequence number in the request.
3C11H	Invalid command in request.
3D11H	Response length does not match 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.
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 request to the PLC because an internal message queue is full.
E911H	Unable to send request to the PLC because the text buffer type is invalid.
EA11H	Unable to send request to the PLC because the mailbox utility function is invalid.
EB11H	Unable to send request to the PLC because the mailbox message is not specified.
EC11H	Unable to send request to the PLC because the internal message queue is not initialized.
FE11H	Request failed due to mailbox error on 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 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.
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	Server unit length is too large.
8890H	Invalid channel number.
8990H	Invalid time unit for period. (Maximum permitted 3965 hours)
8A90H	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 same as the sender's IP address.
9090H	Insufficient TCP connection resources to do request.
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 server end point.
9790H	Underlying TCP connection aborted by client end point.
9890H	The remote server has no Service Request Processor.
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 unexpected size.
9E90H	Unrecognized COMMREQ command code.
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 bit mask.
A790H	Unable to connect to remote device.
A890H	Channel Resources in Use. Try the command again; a resource will become available.

Error Status (Hex)	Error Description
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>▪ If using a switch, make sure the switch is turned on.</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 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 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 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	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 remote device.
07A0H	Password protection does not permit access at remote device.
08A0H	Attempt to write to a consumed exchange; this is not permitted.
09A0H	Internal resource error at 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	Invalid response message from remote device.
0FA0H	Address type is not supported at remote device.
10A0H	A memory access error occurred while processing this command.
11A0H	Remote device did not understand the request.
12A0H	Remote device has no variable defined at the specified address.
13A0H	An attempt was made to write a Read-Only variable at remote device.
14A0H	Data length or contents are invalid for transfer according to the data type of that variable at 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 in the midst of a run mode store. Retry your COMMREQ after updates to the target device are complete.

## 12.9 Using the EGD Management Tool (Rack-based and RX7i Embedded)

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 Chapter 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.

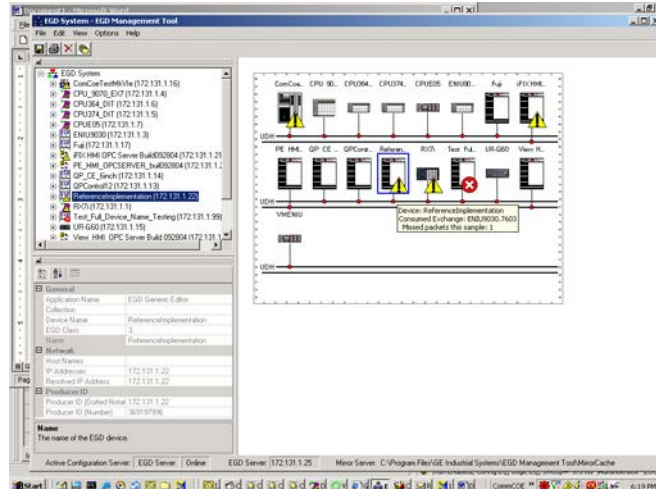


Figure 87: EGD Management Tool Screenshot

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 doing analysis of 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.

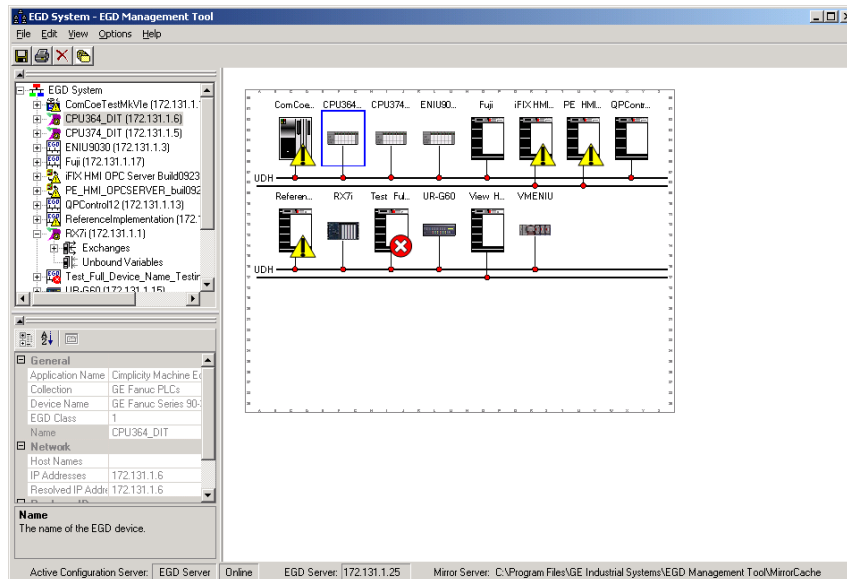


Figure 88: EGD Monitoring Tool Monitoring EGD Network

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. It does not include most of the older Series 90 PLCs.

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.

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.9322	172.131.1.22	172.131.1.1	Unicast	Producer	2004-10-15 15:23:00	2.1	3	500
RK7i.7601	172.131.1.1	172.131.255.255	Broadcast	Consumer	2004-09-09 20:13:37	2.0	2	500
RK7i.7602	172.131.1.1	172.131.255.255	Broadcast	Consumer	2004-09-09 20:13:37	1.1	2	500
RK7i.5101	172.131.1.1	224.0.7.1	Multicast	Consumer	2004-09-09 20:13:37	1.0	2	500
RK7i.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
VMEIU.1	172.131.1.8	172.131.255.255	Broadcast	Consumer	2004-09-20 19:42:59	0.0	40	500
UR-G60.3	172.131.1.15	172.131.255.255	Broadcast	Consumer	2004-07-05 15:44:39	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

Figure 89: EGD Management Tool Displaying EGD Exchange Information

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 statistics information about each exchange. The statistics are updated periodically based on a rate in the Options menu.

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
RX-7.7601	2004-09-09 20:13:37	2004-10-17 22:20:40	Healthy	2	20948	120166	2
RX-7.7602	2004-09-09 20:13:37	2004-10-17 22:20:40	Healthy	2	20824	120290	2
RX-7.5101	2004-09-09 20:13:37	2004-10-17 22:20:40	Healthy	2	20976	120138	2
RX-7.5102	2004-09-09 20:13:37	2004-10-17 22:20:40	Healthy	2	20885	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
VMENU1.1	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
ConCoeTestMKVie.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

Figure 90: EGD Management Tool Displaying EGD Statistics

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** –information about the status of the exchange. For a produced exchange, 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, 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.

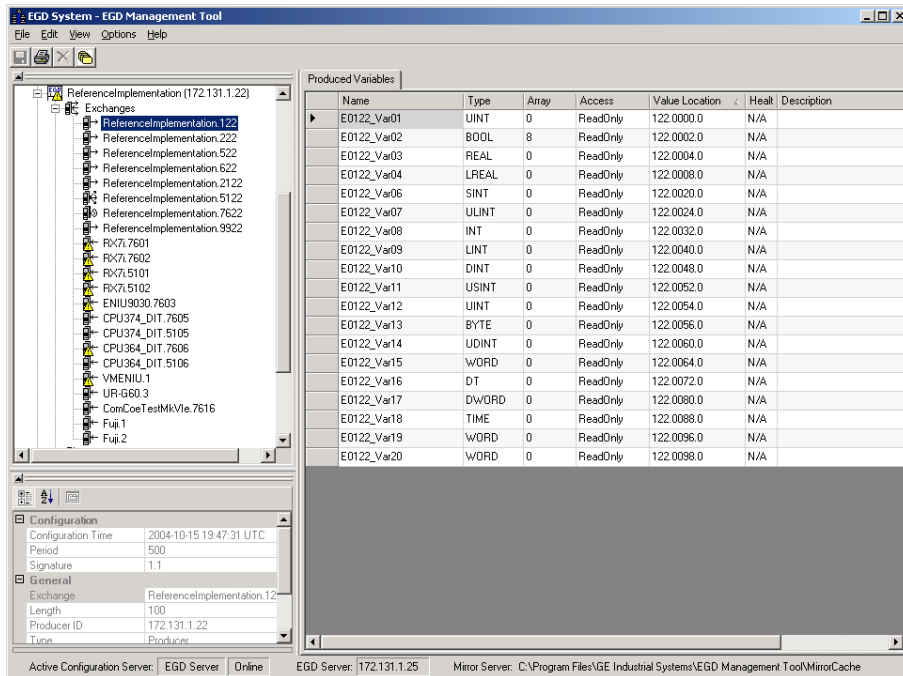


Figure 91: EGD Management Tool Displaying List of Variables for an Exchange

## 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. As previously explained, Controller Faults generated by the Ethernet interface contain Ethernet exception events within the extra fault data. See the *TCP/IP Communications for PACSystems Station Manager 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 in order 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; in this case, 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. The RX7i embedded Ethernet daughterboard uses the Controller 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. The communication service cannot be performed.  
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.

<b>Cause</b>	<b>Corrective Action</b>
Producer and Consumer exchange definitions are of different size.	Review the conflicting exchange definitions at the producer and at 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 7 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.

Release 6.00 of the Ethernet Interface introduces the 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. Revision 6.00 and higher of the PME programmer can override this initial value. See “SRTP Inactivity Timeout” in Chapter 1 for details.

If desired, the TCP connection timeout duration may be adjusted via AUP parameters. See Appendix A 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) web sever 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 in order to establish and maintain connection even under heavy load due to EGD and other SRTP connections. The Ethernet Interface, prior to Release 6.00, supports only one such privileged connection whereas the Release 6.00 Ethernet Interface introduces support for three privileged connections. 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.

As described above under “SRTP Connection Timeout”, 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 “SRTP Inactivity Timeout” in Chapter 1 for 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 Appendix A 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 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 Appendix 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 channels 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 channels 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.





## Chapter 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 92: IP Address Format for Network Classes A, B, C

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 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.

<b>Range of first integer</b>	<b>Class</b>
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 Chapter 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 actually 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.

## **13.2 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.

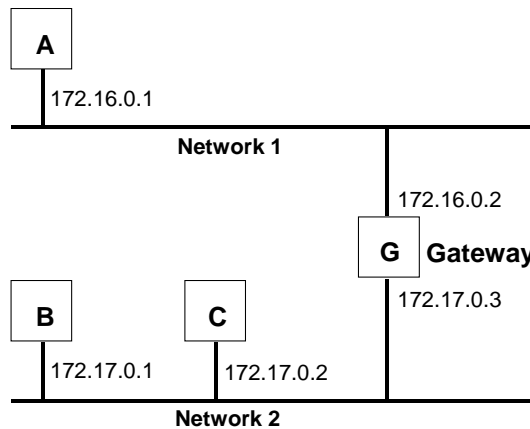


Figure 93: Gateway Connected to Two Networks

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). In order 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 Supernets

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 need 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.

### 13.3.1 Subnet Addressing and Subnet Masks

Subnet addressing is an extension of the IP address scheme that allows a site to use a single netid for multiple physical networks. Routing outside the site continues as usual by dividing the IP address into a netid and a hostid via the class.

The standard format for the netid bits and hostid bits for an IP address in a Class B network is shown below.

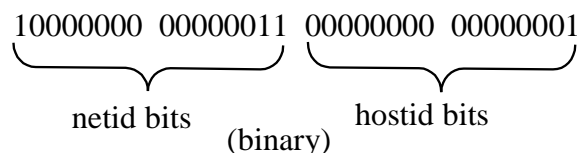
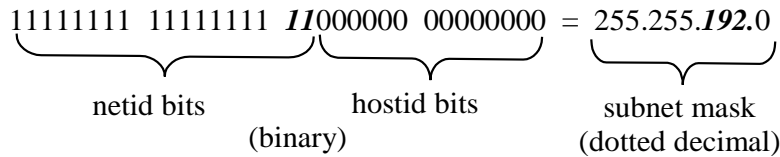


Figure 94: Class B Network netid and hostid Bit Formats

Inside a site the *subnet mask* is used to re-divide the IP address into a custom *netid* portion and *hostid* portion. Consider adding another physical network to Network 2 (a Class B network) in the previous example. The result is shown in the figure below. Selecting the subnet mask shown below would add two additional *netid* bits allowing for four physical networks addressed as 0, 64, 128, and 192. The added subnet bits are normally taken from the *hostid* bits adjacent to the *netid* and the subnet mask identifies these bits.



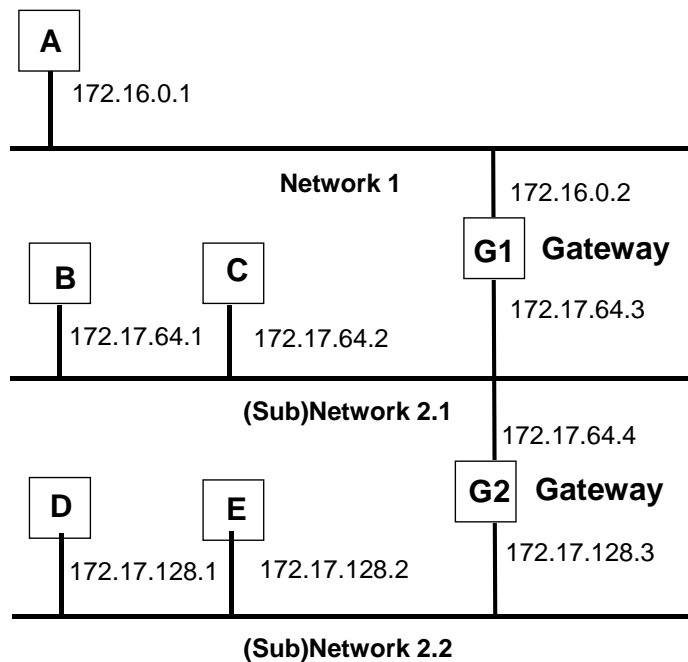
**Figure 95: Use of Subnet Mask**

The bits in the subnet mask correspond one to one with the Internet address. The bits in the mask that are 1 treat the corresponding bits in the IP address as part of the *netid* bits. The bits in the mask that are 0 treat the corresponding bits as part of the *hostid* bits.

In effect, two bits of the Class B *hostid* have been used to extend the *netid*, creating an *extended netid*, or *subnetid*. Each unique combination of bits in the part of the *hostid* where subnet mask bits are 1 specifies a different physical network.

**Example: Network Divided into Two Subnets**

The new network configuration dividing Network 2 into Subnets 2.1 and 2.2 is shown below.



**Figure 96: 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.

**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.

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 255.255.254.0 is applied to both addresses.

$$\begin{array}{ccccccc}
 11111111 & 11111111 & 11111110 & 00000000 & = & 255.255.254.0 \\
 \underbrace{\hspace{10em}} & & \underbrace{\hspace{4em}} & & & \underbrace{\hspace{4em}} \\
 \text{netid bits} & & \text{hostid bits} & & & \text{subnet mask} \\
 \text{(binary)} & & & & & \text{(dotted decimal)}
 \end{array}$$

Figure 97: Subnet Mask Used to Effect a Supernet

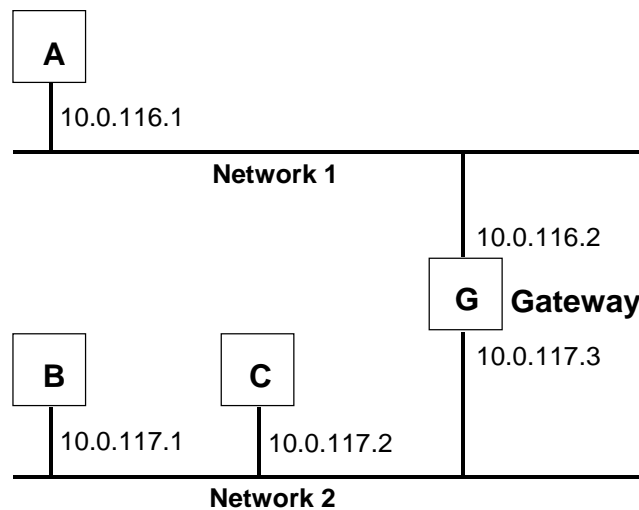


Figure 98: Resulting Supernet



## Appendix A Configuring Advanced User Parameters

---

Advanced User Parameters (AUPs) are internal operating parameters used by the Ethernet interface. For most applications, the default AUPs should not be changed. <sup>22</sup>

If it is necessary to modify any of these parameters, it must be done by creating an AUP file, using any ASCII text editor. This file must contain the names and values of only those parameters that are being changed. This user-generated AUP file is then imported into the programmer and assigned to a particular Ethernet Interface. To modify Advanced User Parameters in more than one Ethernet Interface in the same control system, import an AUP file for each Ethernet Interface. If the changes are identical, you can use the same AUP file for more than one Ethernet Interface.

When the entire hardware configuration is stored from the programmer to the CPU, the programmer also stores the parameters from all assigned AUP files. The CPU delivers any assigned AUP file data to its Ethernet Interface along with its configuration. AUP file data is transferred along with the rest of the hardware configuration during both download (programmer-to-CPU) and upload (CPU-to-programmer) operations. AUP file data is also included in the configuration Verify operation between programmer and CPU. Note that there may be a separate AUP file for each Ethernet interface (or some may have them while others do not).

If an Ethernet Interface is not configured by the programmer, its Station Manager can be used to locally modify the Advanced User Parameters for that individual module. (Setting the IP address/subnet mask via BOOTP or the "SetIP Tool" does not qualify as a programmer configuration.)

---

### Caution



**The IEEE 802.3 standard strongly discourages the manual configuration of duplex mode for a port (as would be possible using AUP.) Before manually configuring duplex mode for a port using AUP, be sure that you know the characteristics of the link partner and are aware of the consequences of your selection. In the words of the IEEE standard: "Connecting incompatible DTE/MAU combinations such as full-duplex mode DTE to a half-duplex 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 the speed and duplex mode of a port are forced using Advanced User Parameters, the switch will no longer perform automatic cable detection. This means that if you have the switch port connected to a switch or hub port you must use a crossover cable. If you have the switch port connected to the uplink port on a switch or hub or if you have the switch port connected to another Ethernet device you must use a normal cable.

---

<sup>22</sup> The RX3i CPE305/CPE310 embedded Ethernet interface does not support the full set of AUPs described in this chapter. For a list of AUPs supported by the RX3i embedded Ethernet interface, refer to page 247.

## A-1 Format of the Advanced User Parameters File

The AUP file 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>

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 sets the station manager password to "system" and the IP time-to-live for point-to-point Ethernet Global Data exchanges to 4.

AUP\_0\_1

stpasswd = "system" ; set the password to "system"

gucast\_ttl=4 ; set the EGD unicast IP TTL to 4



## A-2 Advanced User Parameter Definitions

**Note:** The RX3i CPE305/CPE310 embedded Ethernet interface does not support all AUPs listed. AUPs that can be used with CPE305/CPE310 are indicated by a footnote. Other PACSystems Ethernet interfaces support the use of all AUPs listed in the following table.

System Memory Parameters (task b)		Default	Range
staudp <sup>23</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.
stpasswd <sup>23</sup>	Station Manager password (only visible from MODIFY prompt)	“system”	0 – 8 characters, case sensitive, no spaces

Backplane Driver Parameters (task c)		Default	Range
crsp_tout <sup>23</sup>	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 <sup>23</sup>	HCT compatibility option. (Rel 2.57 and later) 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 <sup>23</sup>	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 <sup>23</sup>	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 cstorm parameter is not set to 0, this parameter should always be less than cstorm. If cstorm is set to 0, this parameter is ignored.	100 (0064H)	0 – 10,000 (2710H)
<b>RDS Parameters (task d)</b>		None	None

<sup>23</sup> Supported by RX3i CPE305 and CPE310 models.

ARP Parameters (task f)		Default	Range
fflush	Interval in seconds at which to flush the ARP cache	600 (10 minutes)	0 – 604800 (93A80H)

Ethernet Global Data Parameters <sup>24</sup> (task g)		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

<sup>24</sup> Effective with RX3i CPE310/CPE305 Firmware Release 8.30, all of the EGD commands are supported except gcmd\_pri and gc\_accel.

gnostale	When bit zero in the "Production Status" 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 values for EGD exchanges with Unicast and Broadcast destination types, the largest value will be used for all Unicast and Broadcast exchanges.

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.

<b>SRTP Client (Channels) Parameters (task h)</b>		<b>Default</b>	<b>Range</b>
hconn_tout	TCP Connect timeout (in milliseconds)	75000 (124F8H)	10 - 75000 (124F8H)

<b>IP Parameters (task i)</b>		<b>Default</b>	<b>Range</b>
lttl <sup>23</sup>	IP header default time-to-live (hop count)	64 (0040H)	0 - 255 (00ffH)
ifrag_tmr <sup>23</sup>	IP fragment timeout interval in seconds	3 (0003H)	0 - 65535 (ffffH)

<b>ICMP/IGMP Parameters (task j)</b>		<b>Default</b>	<b>Range</b>
		None	None

<b>Network Interface Parameters (task l)</b>		<b>Default</b>	<b>Range</b>
lduplex0	Ethernet duplex for Controller (0=auto-detect, 1 = half, 2= full)	0	0,1,2
lduplex1a <sup>23</sup>	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 <sup>23</sup>	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

<b>Modbus TCP/IP Server Parameters (task m)</b>		<b>None</b>	<b>None</b>

<b>SNTP Time Transfer to CPU Parameters (task n)</b>		<b>Default</b>	<b>Range</b>
ncpu_sync	Configures this Ethernet interface to support CPU TOD clock synchronization with network timeserver. (0=Not supported; 1=Supported)	0	0, 1

<b>Unicast SNTP AUP Parameters (task n)</b>		<b>Default</b>	<b>Range</b>
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
npoll_interval	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
npoll_count	Number of retransmissions that will be sent when no timely response is received from the server. This parameter is optional.	3	1 - 100
npoll_timeout	The time, in seconds, that the module will wait for a response from the server. This parameter is optional.	2	1 - 100.

<b>SNTP Local Time Corrections (LTC) and Daylight Savings Time (DST) Parameters (task n)</b>		<b>Default</b>	<b>Range</b>
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
<b>Modbus TCP/IP Client Parameters (task o)</b>		None	None

<b>Ethernet Redundancy Parameters (task q)</b>		<b>Default</b>	<b>Range</b>
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 20ms 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

<b>FTP Parameters (task t)</b>		<b>Default</b>	<b>Range</b>
tpassword	Password for login for FTP access.	“system”	0 to 8 characters

<b>UDP Parameters (task u)</b>		<b>Default</b>	<b>Range</b>
		None	None

<b>SRTP Parameters (task v)</b>		<b>Default</b>	<b>Range</b>
vconn_tout <sup>23</sup>	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 “SRTP Inactivity Timeout” in Chapter 1 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

<b>TCP Parameters (task w)</b>		<b>Default</b>	<b>Range</b>
wndelay <sup>23</sup>	TCP nodelay option (0= inactive; 1 = active)	1 (1H)	0, 1
wkal_idle <sup>23</sup>	TCP keepalive timer value (in seconds)	240 (4.0 min)	1 – 65535 (ffffH)
wkal_cnt <sup>23</sup>	TCP keepalive extra probe count (in addition to single probe always performed)	2	0 – 65535 (ffffH)
wkal_intvl <sup>23</sup>	TCP keepalive probe interval (in seconds)	60 seconds	1 – 65535 (ffffH)
wsnd_buf <sup>23</sup>	TCP send buffer size (in bytes)	65535 (ffffH)	0 – 65535 (ffffH)
wrcv_buf <sup>23</sup>	TCP receive buffer size (in bytes)	4096 (1000H)	0 – 32767 (7fffH)

### A-3 AUPs Supported by RX3i CPE305/CPE310 Embedded Ethernet Interface

The default values and ranges of valid values are the same as those in other PACSystems Ethernet interfaces.

**Note:** When explicitly configuring speed or duplex mode for an RX3i embedded Ethernet port using Advanced User Parameters (AUP), do not request a store to flash as a part of the download when communicating over the CPE305/CPE310 embedded Ethernet port. In this situation you first must store to the RX3i and then initiate a separate request to write to flash.

#### System Memory Parameters (task b)

staudp  
stpasswd

#### Backplane Driver Parameters (task c)

crsp\_tout  
chct\_comp  
cstorm  
cnostorm

#### IP Parameters (task i)

ittl  
ifrag\_tmr

#### Network Interface Parameters (task l)

lduplex1a  
lspeed1a

#### SRTP Server Parameters (task v)

vconn\_tout

#### TCP Parameters (task w)

wndelay  
wkal\_idle  
wkal\_cnt  
wkal\_intvl  
wsnd\_buf  
wrcv\_buf



# Index

---

## A

Abort Channel command (2001), 113  
Aborting a channel, 100  
Advanced User Parameters, 231  
  definitions, 233  
  RX3i embedded, 240  
Application Timeout, 220  
AUP file, 231

## B

BOOTP, 38, 40  
Broadcasting Ethernet Global Data, 62

## C

Cable  
  Ethernet, 6  
Cable, CPU Programming, 26  
Channel Commands, 99, 133  
  Abort Channel (2001), 113  
  Channel number, 105, 108, 111, 113, 114, 134, 137, 139,  
  140, 141, 143, 144, 145, 146, 147  
  Command period, 105, 109, 112, 134, 138, 139, 140, 141  
  Establish Read Channel (2003), 104  
  Establish Write Channel (2004), 108, 136, 142, 146, 147  
  Number of repetitions, 105, 108, 111, 134, 137, 139, 140,  
  141, 143, 144, 145, 146, 148  
  Retrieve Detailed Channel Status (2002), 114  
  Send Information Report (2010), 111  
  Timeout, 105, 109, 112, 138, 139, 140, 141, 143, 144,  
  145, 146, 148  
Channel Error bit, 158, 203  
Channel status bits, 100, 203  
  Modbus channels, 203  
  SRTP channels, 203  
Channel status words, 114  
Channels  
  Aborting, 100  
  Establishing, 133, 135  
  Maximum that can be established, 100  
  Monitoring, 158  
  Numbers assigned, 105, 108, 111, 113, 114, 134, 137,  
  139, 140, 141, 143, 144, 145, 146, 147  
  Re-tasking, 100  
Client PLC, 108, 134, 135, 137, 138, 139, 141, 143, 144, 145,  
  146, 147  
Client/Server Capability, 3  
COMMREQs  
  Channel Commands, 128  
  Command Block, 102, 128, 132  
  controlling execution, 129  
  fault errors, 218

format, 102  
Format for Programming EGD Commands, 83  
function block, 131  
function block status, 204  
functions, maximum pending, 118, 158  
status word, 150, 158, 204  
status word pointer, 103  
structure, 127

Communications Requests. See COMMREQs

Configuration Data

  RX3i, 29  
  RX3i rack-based, 36  
  RX7i, 36

Configuration Mismatch, 220

Configuring Ethernet Global Data, 42, 45, 50  
  redundancy, 43

Configuring the Ethernet Interface

  RX3i rack-based, 39  
  RX7i, 39

**Connection Open bit**

**Modbus TCP channels**, 203

Consumed Data Exchange Definition, 48, 50, 53

Consumer, 58

Controller Fault Table, 198

## D

Data Block

  Length, 103

Data Transfer bit

  SRTP channels, 203

Data Transfers with One Repetition, 118

Detailed channel status words

  format, 115  
  monitoring, 114  
  retrieving, 114

Determining if an IP address has been used, 15, 28

Diagnostic tools

  standard, 194

Documentation, 2

## E

EGD Command Session Conflicts, 222

EGD Management Tool, 214

Embedded switches, 23

EOK LED, 20

Establish Read Channel command (2003), 104

Establish Write Channel command (2004), 108, 136, 142,  
  146, 147

Establishing a channel, 133, 135

Ethernet Global Data, 35, 58

  Configuration for RX3i CPU, 34

  Configuring, 42, 45, 50

  redundancy, 43

Consumed Data Exchange Definition, 48, 50, 53

Consumer, 58

Effect of PLC modes and actions on, 76

EGD Command Session Conflicts, 222

- Exchange, 59
- exchange status word, 81
- Operation, 63
- Produced Data Exchange Definition, 46, 47, 51
- Producer, 58
- Selective Consumption, 55
- signatures, 44
- Variables, 59
- Ethernet interface status bits, 201
- Ethernet Plug-in Applications, 28
- Ethernet Restart Pushbutton, 20
- Exchange status word
  - Ethernet Global Data, 81

## F

- Fault table, 198
- Features
  - RX3i embedded, 4
  - RX3i rack-based and RX7i, 3
- FT Output of the COMMREQ Function Block**, 129, 149

## G

- Gateways, 226

## H

- Hardware failure, 196
- Hub, 25

## I

- Installation
  - RX3i embedded, 13
  - RX3i rack-based, 22
  - RX7i embedded, 21
  - RX7i rack-based, 21
- IP address, 40
  - Address Reserved for Private Networks, 226
  - Assignment
    - RX3i rack-based, 36
    - RX7i, 36
  - Configuration, 31
  - Determining if it has been used, 15, 28
  - Isolated network, 32, 40
- IP Address
  - Multicast, 226
- IP addressing
  - Format, 225

## L

- Ladder programming, 115, 150
- LAN interface status bits, 201
- LED Blink Codes, 197
- LEDs, 27, 195
  - RX3i embedded, 13

- RX3i rack-based and RX7i, 19
- Local PLC, 108, 134, 135, 137, 138, 139, 141, 143, 144, 145, 146, 147
- Logic program
  - controlling execution of COMMREQs, 129
- Loopback IP Addresses, 226

## M

- Mapping
  - modbus to ENIU memory, 123
- Masked Write to EGD Exchange, 95
- Modbus
  - Protocol, 123
  - reference tables, 123
- Modbus Address Space Mapping, 125
- Modbus Channel Status, 203
- Modbus Function Codes, 126
- Modbus/TCP Channel Commands, 128, 133
- Monitoring the communications channel, 158
- Multicast IP Addresses, 226
- Multicasting Ethernet Global Data, 61
- Multiple Gateways, 227

## N

- Name Server IP address, 41, 46
- Network Address, 107, 112
- Network connection
  - RX3i embedded, 14
  - RX3i rack-based and RX7i, 24
- Network time sync, 41
- New features, 2
- Number of repetitions for a Channel Command, 105, 108, 111, 134, 137, 139, 140, 141, 143, 144, 145, 146, 148

## O

- OPC UA Address Space, 178
  - Publish RXi Application Variables to, 179
  - RXi OPC UA Server Information in, 180
- OPC UA Server, 161
  - Anonymous Authentication, 175
  - Automatic Restart Function, 184
  - Certificates, 184
  - Client Authentication Settings, 175
  - Connect to OPC UA Client, 172
  - OPC UA Address Space, 178
  - RXi OPC UA Security Settings, 178
  - Service Requests, 162
  - Username/Password Authentication, 176
- Operating States, 195
- Operational state, 196
- Overtemperature, 202

## P

- Period for Channel Commands, 105, 109, 112, 134, 138, 139, 140, 141

PING Restrictions, 221  
 Pinging the TCP/IP Interfaces on the Network, 15, 27  
 Pinouts, 25  
     RX3i embedded, 14  
 PLC Timeout Errors, 219  
 Plug-in Applications, 28  
 Port Connectors  
     RX3i rack-based and RX7i, 23  
 Port LEDs, 20  
     RX3i embedded, 13  
 Port Settings, 26  
 Power-Up, 27  
 Power-up states, 195  
 Private Networks, IP addresses, 226  
 Produced Data Exchange Definition, 46, 47, 51  
 Producer, 58  
 Producer Period, 64  
 Programmer Response, 222  
 Protocol  
     Modbus, 123

**R**

Read EGD Exchange command, 90  
 Read PLC Memory command, 85  
 Redundancy, 61, 83  
     EGD class 1 operation, 11  
     EGD class 2 operation, 11  
     FTP operation, 12  
     HSB CPU, 8  
     IP address configuration, 12  
     Modbus TCP client operation, 11  
     Modbus TCP server operation, 11  
     non-HSB, 9  
     operation, 8  
     remote station manager, 12  
     role switching, 9  
     SNTP operation, 12  
     SRTP client operation, 11  
     SRTP server operation, 10  
     Web server operation, 12  
 Redundant IP address, 41, 202  
 Related documents, 2  
 Remote PLC, 108, 134, 135, 137, 138, 139, 141, 143, 144, 145, 146, 147  
 Repeater, 25  
 Repetitions, number of for Channel Commands, 105, 108, 111, 134, 137, 139, 140, 141, 143, 144, 145, 146, 148  
 Re-tasking a channel, 100  
 Retrieve Detailed Channel Status command (2002), 114  
 Role switch in a non-HSB redundancy system, 10  
 Run mode store (RMS) of EGD, 77  
     Group ID, 62  
     using signatures with, 45  
 RXi OPC UA Automatic Restart Function, 184  
 RXi OPC UA Server Certificates, 184

**S**

Sample ladder program, 115, 150  
 Scan Set, 32, 41  
 Send Information Report command (2010), 111  
 Sequencing communications requests, 118, 158  
 Serial port configuration  
     RX7i, 42  
     RX3i rack-based, 42  
 Server Capability, 3  
 Server PLC, 108, 134, 135, 137, 138, 139, 141, 143, 144, 145, 146, 147  
 Server Protocol Services, 123  
 Service request  
     set application redundancy mode, 10  
 Service Request Transfer Protocol (SRTP)  
     channel status, 203  
     inactivity timeout, 7, 221, 239  
 Signatures, 44  
 Simple isolated network configuration, 32, 40  
 SNTP  
     Broadcast Mode, 75  
     Daylight saving time correction, 76, 238  
     Local time zone correction, 76, 238  
     Multicast Mode, 75  
     Multiple Servers, 75  
     Operation, 75  
     Timing Signals, 76  
     Unicast Mode, 75  
 Software Loader, 196  
 SRTP. See Service Request Transfer Protocol  
 STAT LED, 19  
 Station Manager, 6, 194  
 Station Manager Lockout under Heavy Load, 221  
 Station Manager Port  
     RX3i rack-based and RX7i, 26  
 Station Manager supported by Modbus Server, 123  
 Status address location, 32, 41  
 Status bits, 201  
 Status data, Channel Commands, 129  
 Subnet Addressing and Subnet Masks, 227  
 Subnet mask, 228  
 Subnets, 227  
 Supernets, 227  
 Switch, 25

**T**

Table of Contents, v  
 Table of Figures, xiii  
 Technical Support. See page iii  
 Telnet, 38  
 Time units for command period, 105, 109, 112, 134, 138, 139, 140, 141  
 Time-of-day clock  
     synchronizing to SNTP server, 68  
 Timeout for Channel Commands, 105, 109, 112, 138, 139, 140, 141, 143, 144, 145, 146, 148  
 Timeout Period, 64

Timestamp  
  Obtaining from CPE305/CPE310 Embedded Ethernet  
  Interface, 67  
Timestamping of Ethernet Global Data, 65  
Troubleshooting  
  Ladder programs, 157

## **V**

Variables  
  Ethernet Global Data, 59  
  symbolic, 60

## **W**

Waiting for configuration from PLC, 196  
Waiting for IP address, 196  
Write EGD Exchange command, 93  
Write PLC Memory command, 88



**GE Intelligent Platforms  
Information Centers**

**Headquarters:**

1-800-433-2682 or 1-434-978-5100

Global regional phone numbers  
are available on our web site  
[www.ge-ip.com](http://www.ge-ip.com)

**Additional Resources**

For more information, please visit  
the GE Intelligent Platforms web site:

[www.ge-ip.com](http://www.ge-ip.com)

Copyright ©2002-2014 General Electric Company. All Rights Reserved

\*Trademark of General Electric Company.

All other brands or names are property of their respective holders.

GFK-2224M