



# ProTalk

## PTQ-GEC

Quantum / Unity Platform

Generic ASCII Ethernet Interface Module

## Application Reference Guide

May 05, 2008



## Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Schneider Electric Quantum / Unity hardware, the PTQ-GEC Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

This manual is provided to assist the user. Every attempt has been made to ensure that the information provided is accurate and a true reflection of the product's installation requirements. In order to ensure a complete understanding of the operation of the product, the user should read all applicable Schneider Electric documentation on the operation of the Schneider Electric hardware.

Under no conditions will ProSoft Technology be responsible or liable for indirect or consequential damages resulting from the use or application of the product.

Reproduction of the contents of this manual, in whole or in part, without written permission from ProSoft Technology is prohibited.

Information in this manual is subject to change without notice and does not represent a commitment on the part of ProSoft Technology Improvements and/or changes in this manual or the product may be made at any time. These changes will be made periodically to correct technical inaccuracies or typographical errors.

### PTQ Installation and Operating Instructions

The statement "power, input and output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods Article 501-10(b) of the National Electrical Code, NFPA 70 for installations in the U.S., or as specified in section 18-1J2 of the Canadian Electrical Code for installations within Canada and in accordance with the authority having jurisdiction".

The following or equivalent warnings shall be included:

- A** Warning - Explosion Hazard - Substitution of components may Impair Suitability for Class I, Division 2;
- B** Warning - Explosion Hazard - When in Hazardous Locations, Turn off Power before replacing Wiring Modules, and
- C** Warning - Explosion Hazard - Do not Disconnect Equipment unless Power has been switched Off or the Area is known to be Nonhazardous.
- D** Caution: The Cell used in this Device may Present a Fire or Chemical Burn Hazard if Mistreated. Do not Disassemble, Heat above 100°C (212°F) or Incinerate.

### Important Notice:



CAUTION: THE CELL USED IN THIS DEVICE MAY PRESENT A FIRE OR CHEMICAL BURN HAZARD IF MISTREATED. DO NOT DISASSEMBLE, HEAT ABOVE 100°C (212°F) OR INCINERATE.

Maximum battery load = 200  $\mu$ A.

Maximum battery charge voltage = 3.4 VDC.

Maximum battery charge current = 500  $\mu$ A.

Maximum battery discharge current = 30  $\mu$ A.

---

## **Your Feedback Please**

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

### **ProSoft Technology**

1675 Chester Avenue, Fourth Floor

Bakersfield, CA 93301

+1 (661) 716-5100

+1 (661) 716-5101 (Fax)

<http://www.prosoft-technology.com>

Copyright © ProSoft Technology, Inc. 2000 - 2008. All Rights Reserved.

PTQ-GEC Application Reference Guide

May 05, 2008

PSFT.GEC.PTQ.UM.08.05.05

ProSoft Technology®, ProLinX®, inRAx®, ProTalk® and RadioLinX® are Registered Trademarks of ProSoft Technology, Inc.



# Contents

<b>PLEASE READ THIS NOTICE.....</b>	<b>2</b>
PTQ Installation and Operating Instructions.....	2
Important Notice: .....	2
Your Feedback Please .....	3
<b>GUIDE TO THE PTQ-GEC APPLICATION REFERENCE GUIDE.....</b>	<b>7</b>
<b>1 START HERE.....</b>	<b>9</b>
1.1 Hardware and Software Requirements.....	9
1.1.1 ProTalk Module Carton Contents .....	9
1.1.2 Quantum / Unity Hardware .....	10
1.1.3 PC and PC Software .....	10
1.2 Install ProSoft Configuration Builder Software.....	11
<b>2 CONFIGURING THE PROCESSOR WITH CONCEPT.....</b>	<b>13</b>
2.1 Information for Concept Version 2.6 Users.....	14
2.1.1 Installing MDC Configuration Files .....	14
2.2 Create a New Project.....	15
2.3 Add the PTQ Module to the Project.....	18
2.4 Set up Data Memory in Project.....	20
2.5 Download the Project to the Processor.....	23
2.6 Verify Successful Download .....	26
<b>3 CONFIGURING THE PROCESSOR WITH PROWORX .....</b>	<b>31</b>
<b>4 CONFIGURING THE PROCESSOR WITH UNITYPRO XL .....</b>	<b>35</b>
4.1 Create a New Project.....	35
4.2 Add the PTQ Module to the Project.....	37
4.3 Build the Project.....	39
4.4 Connect Your PC to the Processor.....	40
4.4.1 Connecting to the Processor with TCPIP .....	41
4.5 Download the Project to the Processor.....	42
<b>5 SETTING UP THE PROTALK MODULE.....</b>	<b>43</b>
5.1 Install the ProTalk Module in the Quantum Rack .....	43
5.1.1 Verify Jumper Settings .....	43
5.1.2 Inserting the 1454-9F connector.....	44
5.1.3 Install the ProTalk Module in the Quantum Rack.....	44
5.2 Connect the PC to the ProTalk Configuration/Debug Port .....	45
5.3 Verify Communication Between the Processor and the Module .....	47
<b>6 MODIFYING THE CONFIGURATION FILE.....</b>	<b>49</b>
6.1 ProSoft Configuration Builder .....	49
6.1.1 Set Up the Project .....	50
6.1.2 Set Module Parameters.....	52
6.2 [Module] .....	54
6.2.1 Module Name Parameter .....	54
6.3 [BACKPLANE].....	54
6.3.1 3x Register Start.....	54
6.3.2 4x Register Start.....	54
6.4 [Server n].....	55
6.4.1 Enabled .....	55
6.4.2 Service Port.....	55
6.4.3 Connection Timeout .....	55
6.4.4 Connection Close Type .....	55
6.4.5 Swap RX Data Bytes .....	56

6.4.6	Swap TX Data Bytes .....	56
6.5	Ethernet Configuration.....	56
6.6	Download the Project to the Module .....	57
<b>7</b>	<b>DIAGNOSTICS AND TROUBLESHOOTING.....</b>	<b>59</b>
7.1	Reading Status Data from the Module .....	59
7.1.1	Required Hardware .....	59
7.1.2	The Configuration/Debug Menu .....	60
7.1.3	Main Menu.....	62
7.1.4	Network Menu .....	65
7.2	LED Status Indicators .....	66
7.3	Ethernet LED Indicators .....	67
<b>8</b>	<b>REFERENCE .....</b>	<b>69</b>
8.1	Product Specifications .....	69
8.1.1	Features and Benefits .....	69
8.1.2	General Specifications .....	69
8.1.3	Hardware Specifications.....	70
8.1.4	Functional Specifications.....	70
8.2	Functional Overview .....	71
8.2.1	General Concepts .....	71
8.2.2	Backplane Data Exchange .....	73
8.2.3	Input and Output Blocks .....	74
8.2.4	Network Data Transfer .....	79
8.3	Cable Connections .....	82
8.3.1	Ethernet Connection.....	82
8.3.2	RS-232 Configuration/Debug Port.....	83
8.4	PTQ-GEC Status Data .....	83
8.4.1	Configuration Error Word Bits.....	89
8.5	Configuration Data.....	89
<b>9</b>	<b>SUPPORT, SERVICE &amp; WARRANTY .....</b>	<b>91</b>
9.1	How to Contact Us: Sales and Support.....	92
9.2	Return Material Authorization (RMA) Policies and Conditions.....	92
9.2.1	All Product Returns .....	93
9.3	Procedures for Return of Units Under Warranty .....	93
9.4	Procedures for Return of Units Out of Warranty .....	94
9.4.1	Un-repairable Units .....	94
9.5	LIMITED WARRANTY.....	95
9.5.1	What Is Covered By This Warranty .....	95
9.5.2	What Is Not Covered By This Warranty.....	96
9.5.3	DISCLAIMER REGARDING HIGH RISK ACTIVITIES .....	97
9.5.4	DISCLAIMER OF ALL OTHER WARRANTIES .....	97
9.5.5	LIMITATION OF REMEDIES** .....	97
9.5.6	Time Limit for Bringing Suit .....	97
9.5.7	No Other Warranties .....	98
9.5.8	Intellectual Property.....	98
9.5.9	Additional Restrictions Relating To Software And Other Intellectual Property.....	98
9.5.10	Allocation of risks .....	98
9.5.11	Controlling Law and Severability .....	99
<b>INDEX.....</b>		<b>101</b>

## Guide to the PTQ-GEC Application Reference Guide

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 9)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 47)  Diagnostics and Troubleshooting (page 59)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 69)  Functional Overview (page 71)  Product Specifications (page 69)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 91)	This section contains Support, Service and Warranty information.  Index of chapters.



# 1 Start Here

## *In This Chapter*

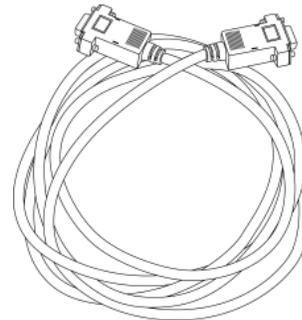
- ❖ Hardware and Software Requirements ..... 9
- ❖ Install ProSoft Configuration Builder Software..... 11

This guide is intended to guide you through the ProTalk module setup process, from removing the module from the box to exchanging data with the processor. In doing this, you will learn how to:

- Set up the processor environment for the PTQ module
- View how the PTQ module exchanges data with the processor
- Edit and download configuration files from your PC to the PTQ module
- Monitor the operation of the PTQ module

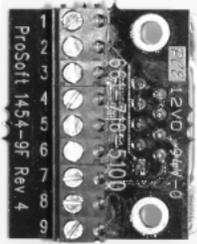
## 1.1 Hardware and Software Requirements

### 1.1.1 ProTalk Module Carton Contents



ProTalk Module

Null Modem Serial Cable



1454-9F DB-9 Female to 9 Pos Screw Terminal adapter (Serial protocol modules only)

ProSoft Solutions CD

**Note:** The DB-9 Female to 5 Pos Screw Terminal adapter is not required on Ethernet modules and is therefore not included in the carton with these types of modules.

### 1.1.2 Quantum / Unity Hardware

This guide assumes that you are familiar with the installation and setup of the Quantum / Unity hardware. The following should be installed, configured and powered up before proceeding:

- Quantum or Unity Processor
- Quantum rack
- Quantum power supply
- Quantum Modbus Plus Network Option Module (NOM Module) (optional)
- Quantum to PC programming hardware
- NOM Ethernet or Serial connection to PC

### 1.1.3 PC and PC Software

- Windows-based PC with at least one COM port
- Quantum programming software installed on machine
- or
- Concept™ PLC Programming Software version 2.6
- or
- ProWORX PLC Programming Software
- or
- UnityPro XL PLC Programming Software
- HyperTerminal (used in this guide) This is a communication program that is included with Microsoft Windows. You can normally find it in **Start / Programs / accessories / Communications**.

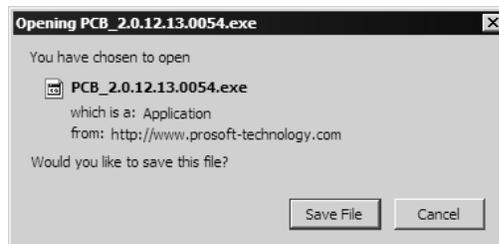
**Note:** ProTalk modules are compatible with common Quantum / Unity programming applications, including Concept and UnityPro XL. For all other programming applications, please contact technical support.

## 1.2 Install ProSoft Configuration Builder Software

You must install the ProSoft Configuration Builder (PCB) software in order to configure the PTQ-GEC module. You can always get the newest version of ProSoft Configuration Builder from the ProSoft Technology web site.

### To install ProSoft Configuration Builder from the ProSoft Web Site

- 1 Open your web browser and navigate to <http://www.prosoft-technology.com/pcb>
- 2 Click the **Download Here** link to download the latest version of ProSoft Configuration Builder.
- 3 Choose "Save" or "Save File" when prompted. The following illustrations show the file download prompt for two of the most common web browsers.



- 4 Make a note of the location where you saved the file, for example "Desktop", or "My Documents", so you can start the installation program.
- 5 When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

If you do not have access to the Internet, you can install ProSoft Configuration Builder from the ProSoft Solutions CD-ROM, included in the package with your PTQ-GEC module.

### To install ProSoft Configuration Builder from the CD-ROM

- 1 Insert the ProSoft Solutions CD-ROM into the CD drive of your PC. Wait for the startup screen to appear.
- 2 On the startup screen, click *Product Documentation*. This action opens an explorer window.

- 3 Click to open the *Utilities* folder. This folder contains all of the applications and files you will need to set up and configure your module.
- 4 Double-click the *ProSoft Configuration Builder Setup* program and follow the instructions on your screen to install the software on your PC.

**Note:** Many of the configuration and maintenance procedures use files and other utilities on the CD-ROM. You may wish to copy the files from the Utilities folder on the CD-ROM to a convenient location on your hard drive.

## 2 Configuring the Processor with Concept

### *In This Chapter*

- ❖ Information for Concept Version 2.6 Users..... 14
- ❖ Create a New Project ..... 15
- ❖ Add the PTQ Module to the Project..... 18
- ❖ Set up Data Memory in Project..... 20
- ❖ Download the Project to the Processor ..... 23
- ❖ Verify Successful Download ..... 26

The following steps are designed to ensure that the processor is able to transfer data successfully with the PTQ module. As part of this procedure, you will use Concept configuration software from Schneider Electric to create a project, add the PTQ module to the project, set up data memory for the project, and then download the project to the processor.

**Important Note:** Concept software does not report whether the PTQ module is present in the rack, and therefore is not able to report the health status of the module when the module is online with the Quantum processor. Please take this into account when monitoring the status of the PTQ module.

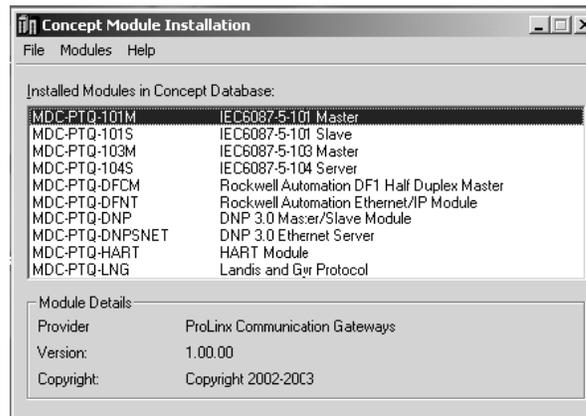
## 2.1 Information for Concept Version 2.6 Users

This guide uses Concept PLC Programming Software version 2.6 to configure the Quantum PLC. The ProTalk installation CD includes MDC module configuration files that help document the PTQ installation. Although not required, these files should be installed before proceeding to the next section.

### 2.1.1 Installing MDC Configuration Files

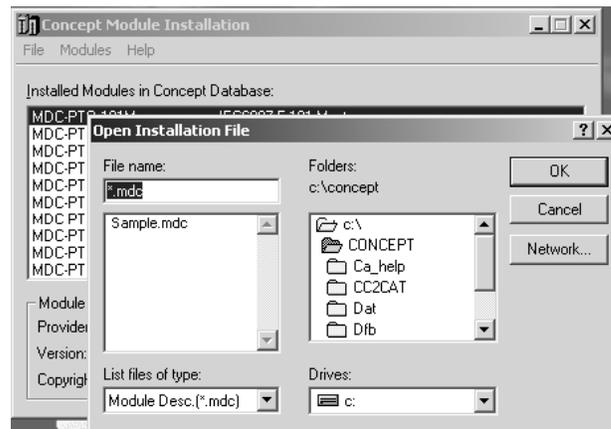
- 1 From a PC with Concept 2.6 installed, choose **Start / Programs / Concept / ModConnect Tool**.

This action opens the Concept Module Installation dialog box.



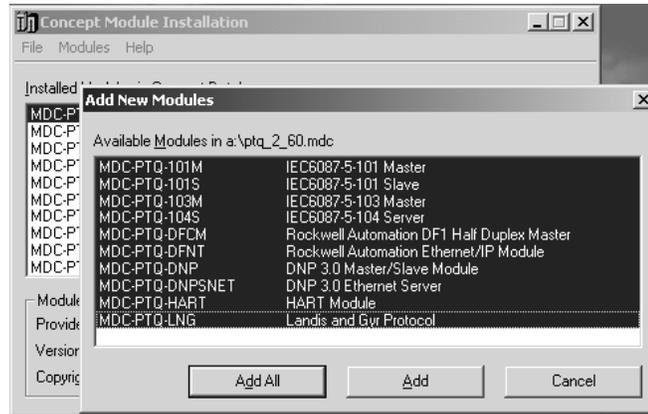
- 2 Choose **File / Open Installation File**.

This action opens the Open Installation File dialog box:



- 3 If you are using a Quantum processor, you will need the MDC files. In the Open Installation File dialog box, navigate to the **MDC Files** directory on the ProTalk CD.
- 4 Choose the MDC file and help file for your version of Concept:
  - o Concept 2.6 users: select PTQ\_2\_60.mdc and PTQMDC.hlp
  - o Concept 2.5 users: select PTQ\_2\_50.mdc and PTQMDC.hlp.

Select the files that go with the Concept version you are using, and then click **OK**. This action opens the add New Modules dialog box.

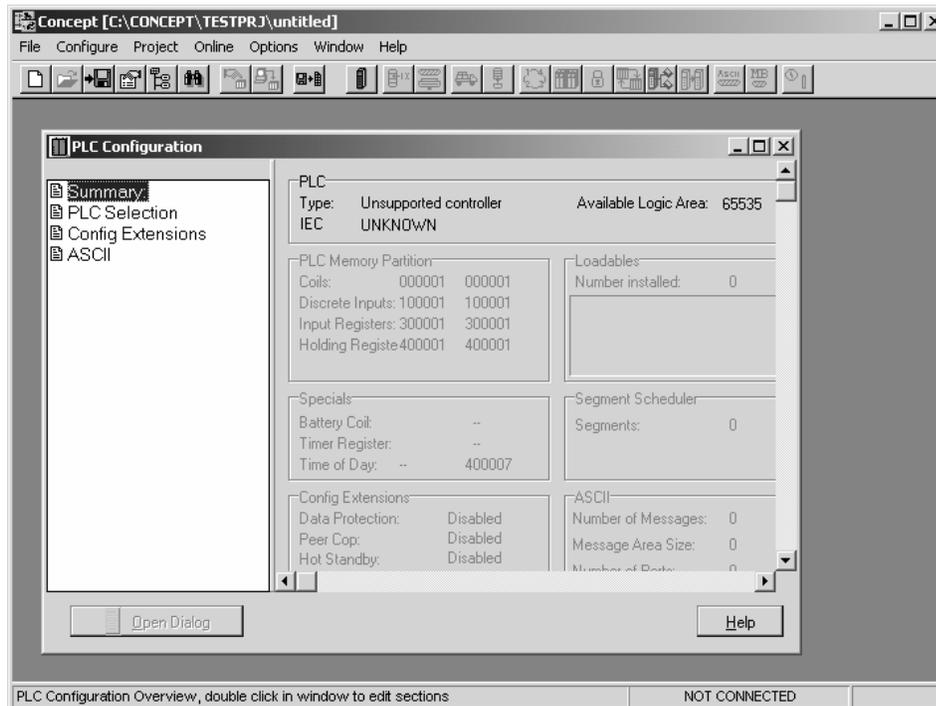


- 5 Click the **add all** button. A series of message boxes may appear during this process. Click **Yes** or **OK** for each message that appears.
- 6 When the process is complete, open the File menu and choose Exit to save your changes.

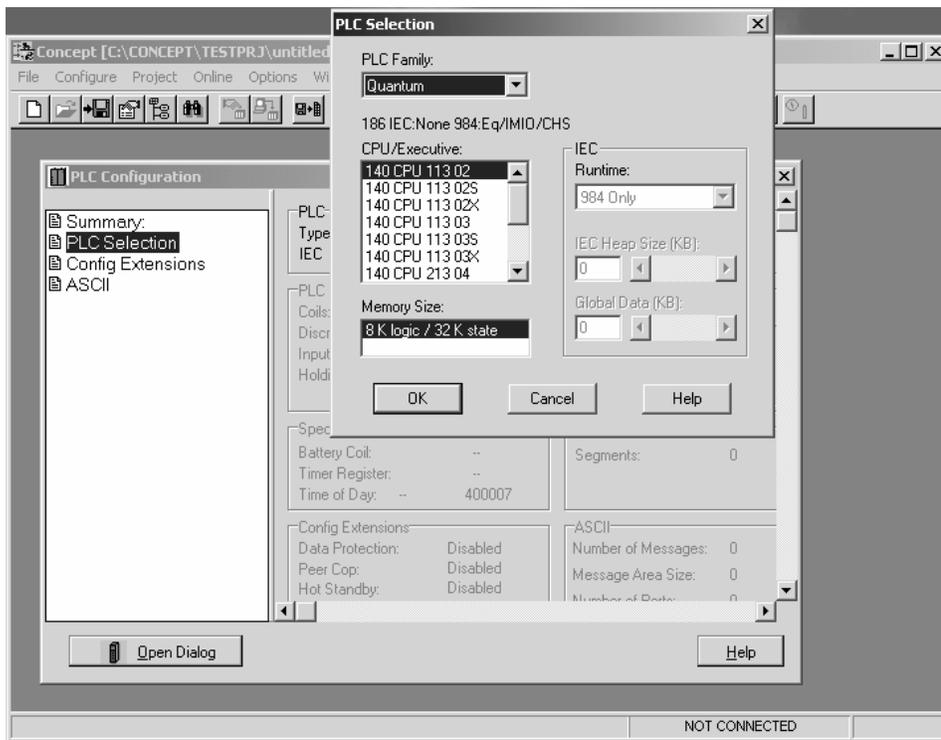
## 2.2 Create a New Project

This phase of the setup procedure must be performed on a computer that has the Concept configuration software installed.

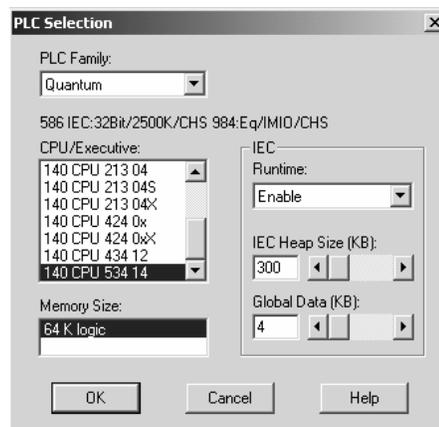
- 1 From your computer, choose **Start / Programs / Concept V2.6 XL.EN / Concept**. This action opens the Concept window.
- 2 Open the File menu, and then choose **New Project**. This action opens the PLC Configuration dialog box.



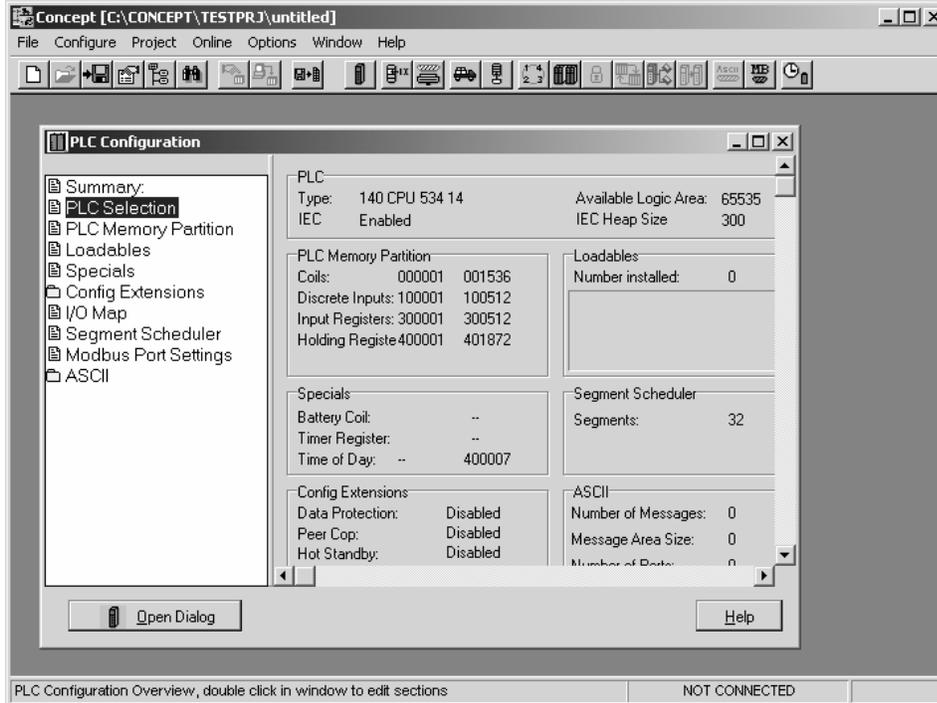
- 3 In the list of options on the left side of this dialog box, double-click the *PLC Selection* folder. This action opens the PLC Selection dialog box.



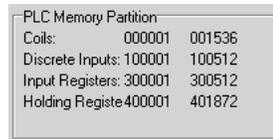
- 4 In the *CPU/Executive* pane, use the scroll bar to locate and select the PLC to configure.



- Click **OK**. This action opens the *PLC Configuration* dialog box, populated with the correct values for the PLC you selected.



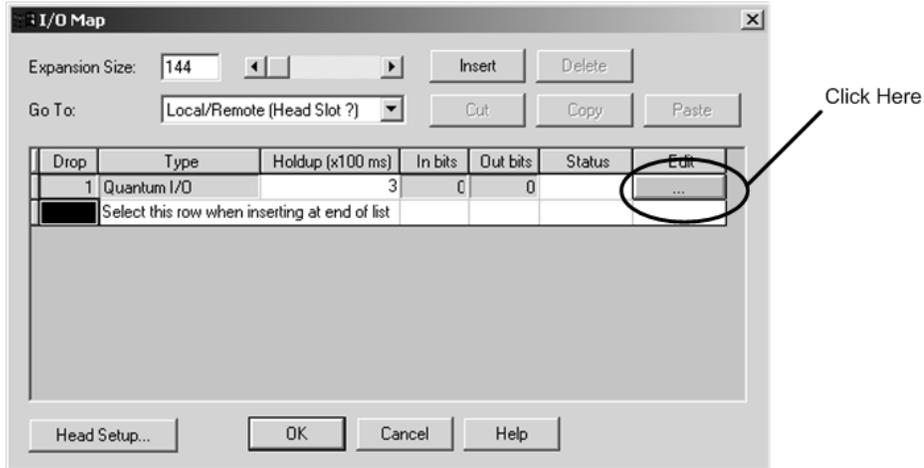
- Make a note of the holding registers for the module. You will need this information when you modify your application as outlined in the ProTalk application Reference Guides. The Holding Registers are displayed in the PLC Memory Partition pane of the PLC Configuration dialog box.



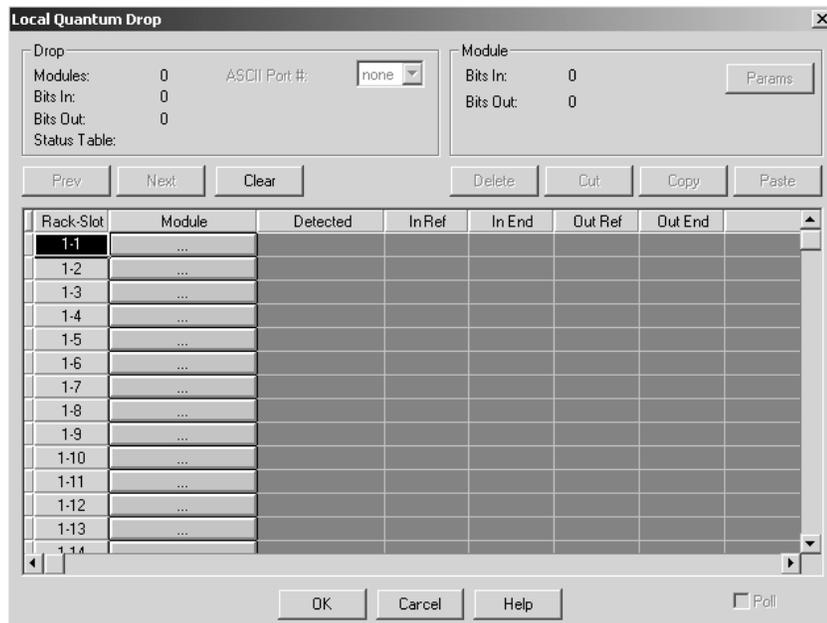
### 2.3 Add the PTQ Module to the Project

The next step is to add one or more of the PTQ modules to the Project. To add modules:

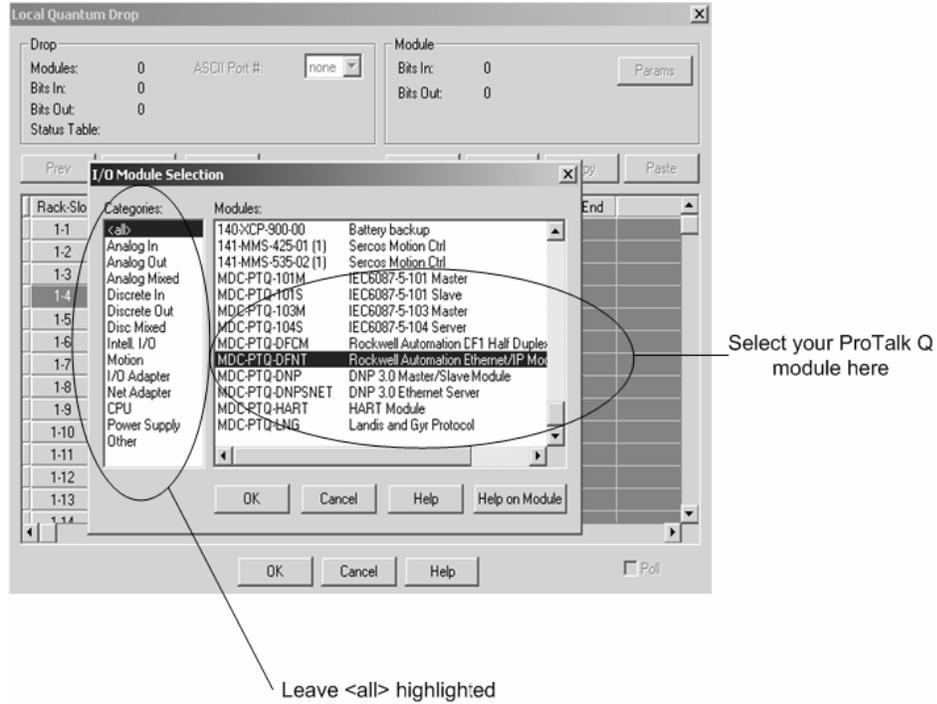
- 1 In the list of options on the left side of the *PLC Configuration* dialog box, double-click *I/O Map*. This action opens the *I/O Map* dialog box.



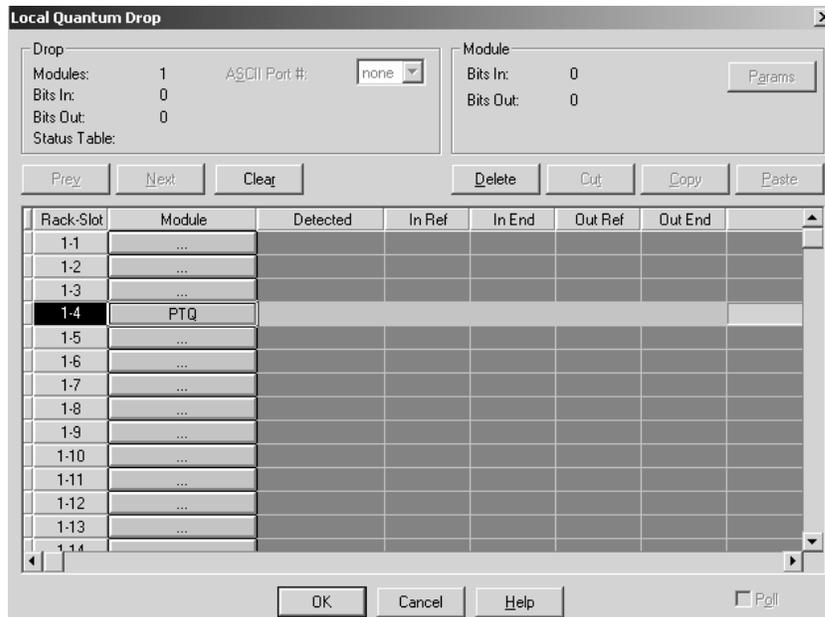
- 2 Click the **Edit** button to open the *Local Quantum Drop* dialog box. This dialog box is where you identify rack and slot locations.



- Click the Module button next to the rack/slot position where the ProTalk module will be installed. This action opens the I/O Module Selection dialog box.

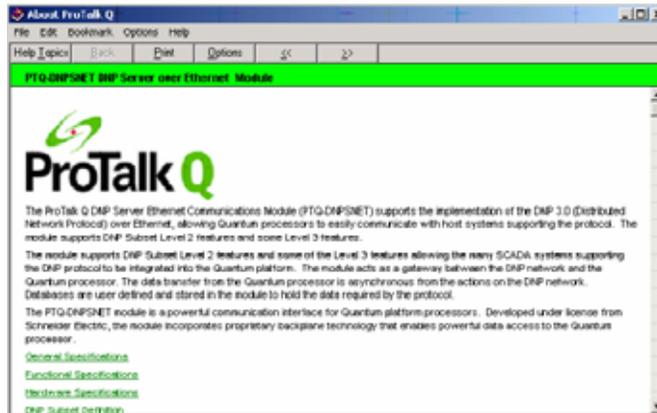


- In the Modules pane, use the scroll bar to locate and select the ProTalk module, and then click OK. This action copies the description of the ProTalk module next to the assigned rack and slot number of the Local Quantum Drop dialog box.



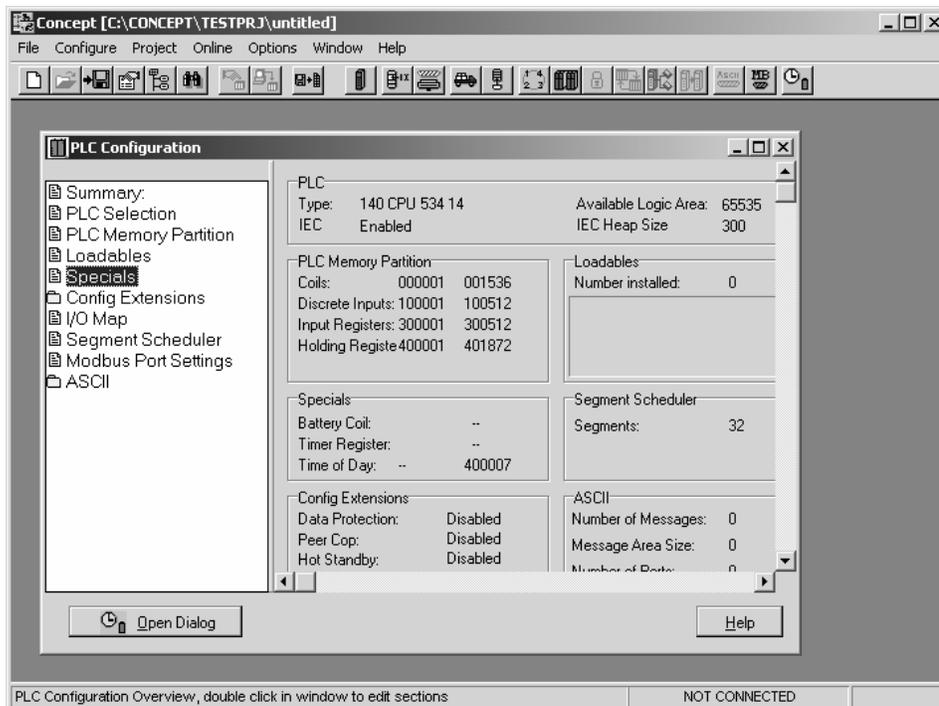
- Repeat steps 3 through 5 for each ProTalk module you plan to install. When you have finished installing your ProTalk modules, click OK to save your settings. Click Yes to confirm your settings.

Tip: Select a module, and then click the Help on Module button for help pages.



## 2.4 Set up Data Memory in Project

- In the list of options on the left side of the PLC Configuration dialog box, double-click Specials.

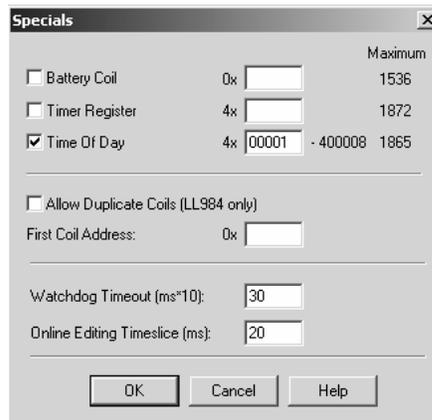


- This action opens the Specials dialog box.



### Selecting the Time of Day

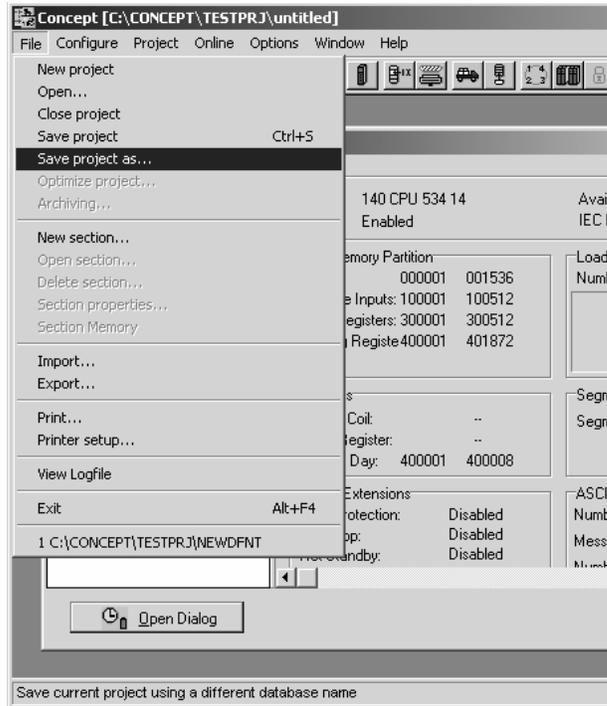
- Select (check) the Time of Day box, and then enter the value 00001 as shown in the following example. This value sets the first time of day register to 400001.



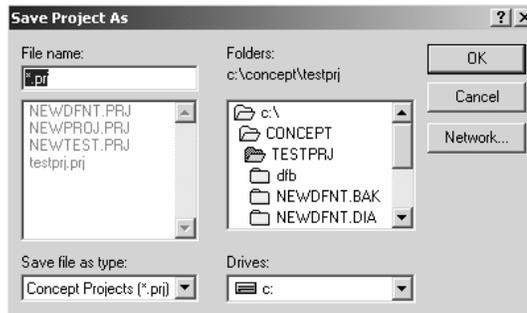
- Click OK to save your settings and close the Specials dialog box.

### Saving your project

- 1 In the PLC Configuration dialog box, choose File / Save project as.



- 2 This action opens the Save Project as dialog box.

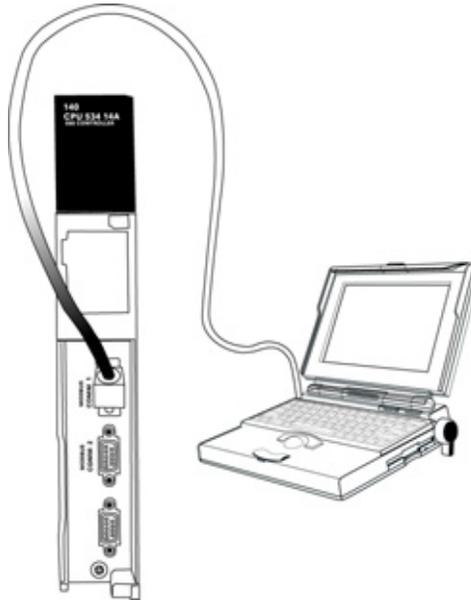


- 3 Name the project, and then click OK to save the project to a file.

## 2.5 Download the Project to the Processor

The next step is to download (copy) the project file to the Quantum Processor.

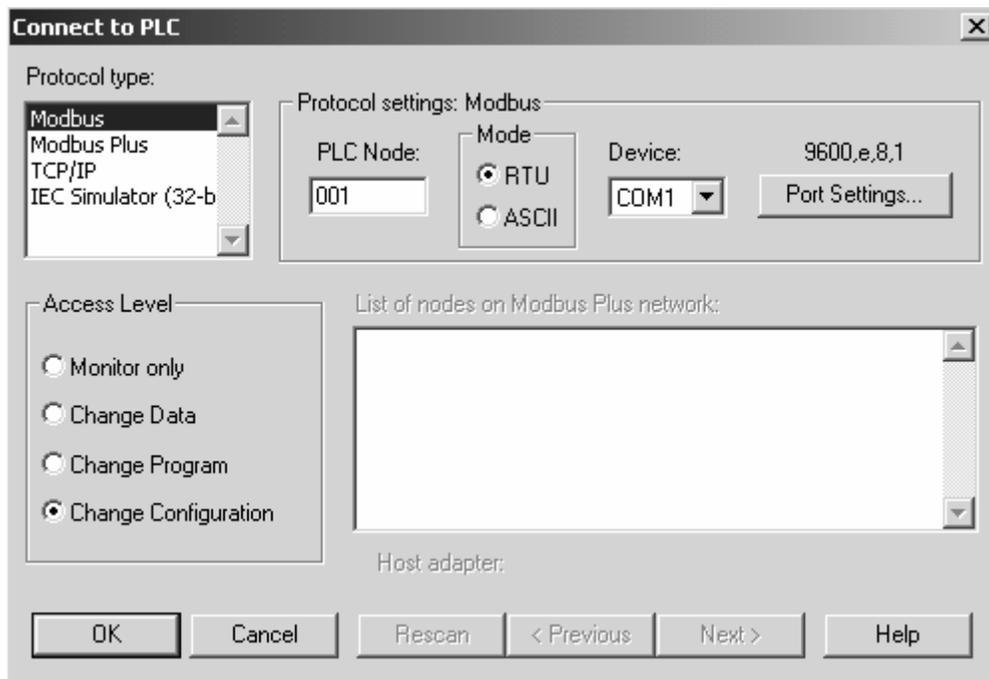
- 1 Use the null modem cable to connect your PC's serial port to the Quantum processor, as shown in the following illustration.



**Note:** You can use a Modbus Plus Network Option Module (NOM Module) module in place of the serial port if necessary.

- 2 Open the PLC menu, and then choose Connect.

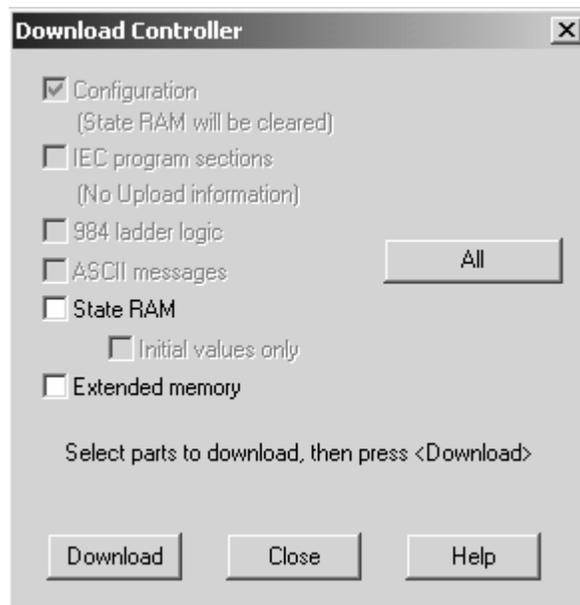
- 3 In the PLC Configuration dialog box, open the Online menu, and then choose Connect. This action opens the Connect to PLC dialog box.



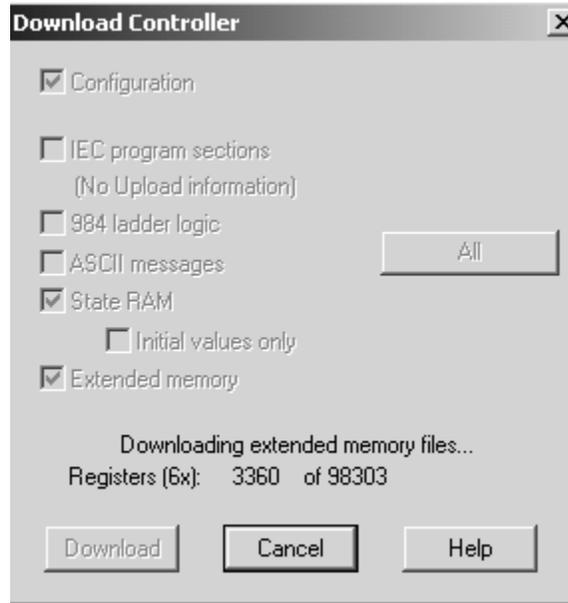
- 4 Leave the default settings as shown and click OK.

**Note:** Click OK to dismiss any message boxes that appear during the connection process.

- 5 In the PLC Configuration window, open the Online menu, and then choose Download. This action opens the Download Controller dialog box.



- Click all, and then click Download. If a message box appears indicating that the controller is running, click Yes to shut down the controller. The Download Controller dialog box displays the status of the download as shown in the following illustration.

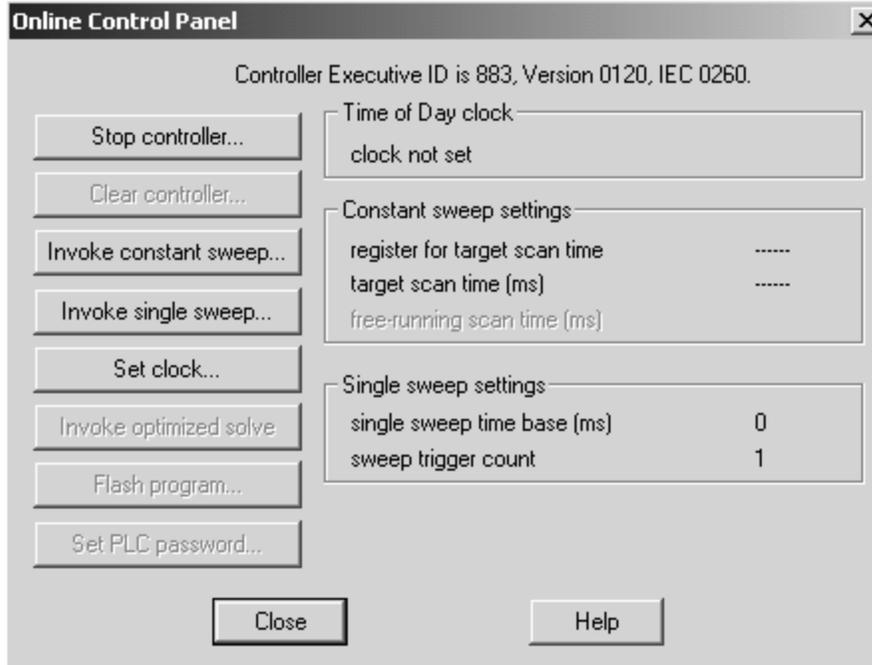


- When the download is complete, you will be prompted to restart the controller. Click Yes to restart the controller.

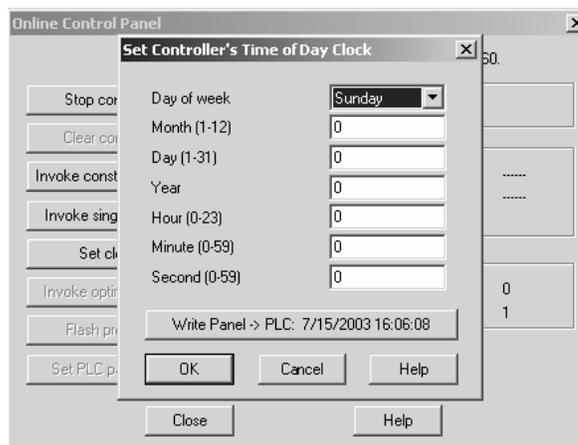
## 2.6 Verify Successful Download

The final step is to verify that the configuration changes you made were received successfully by the module, and to make some adjustments to your settings.

- 1 In the PLC Configuration window, open the Online menu, and then choose Online Control Panel. This action opens the Online Control Panel dialog box.

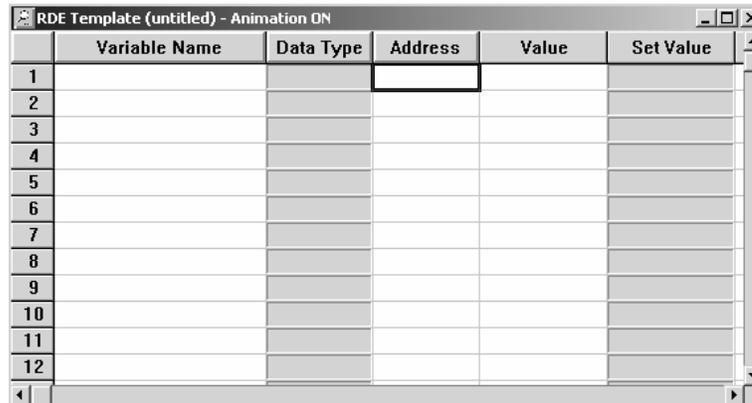


- 2 Click the Set Clock button to open the Set Controller's Time of Day Clock dialog box.



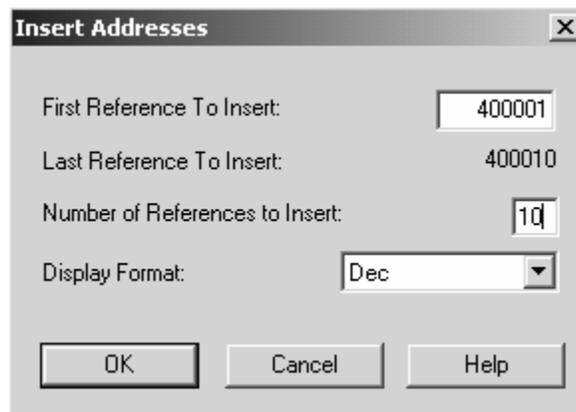
- 3 Click the Write Panel button. This action updates the date and time fields in this dialog box. Click OK to close this dialog box and return to the previous window.
- 4 Click Close to close the Online Control Panel dialog box.

- 5 In the PLC Configuration window, open the Online menu, and then choose Reference Data Editor. This action opens the Reference Data Editor dialog box. On this dialog box, you will add preset values to data registers that will later be monitored in the ProTalk module.
- 6 Place the cursor over the first address field, as shown in the following illustration.



	Variable Name	Data Type	Address	Value	Set Value
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

- 7 In the PLC Configuration window, open the Templates menu, and then choose Insert addresses. This action opens the Insert addresses dialog box.
- 8 On the Insert addresses dialog box, enter the values shown in the following illustration, and then click OK.



**Insert Addresses**

First Reference To Insert: 400001

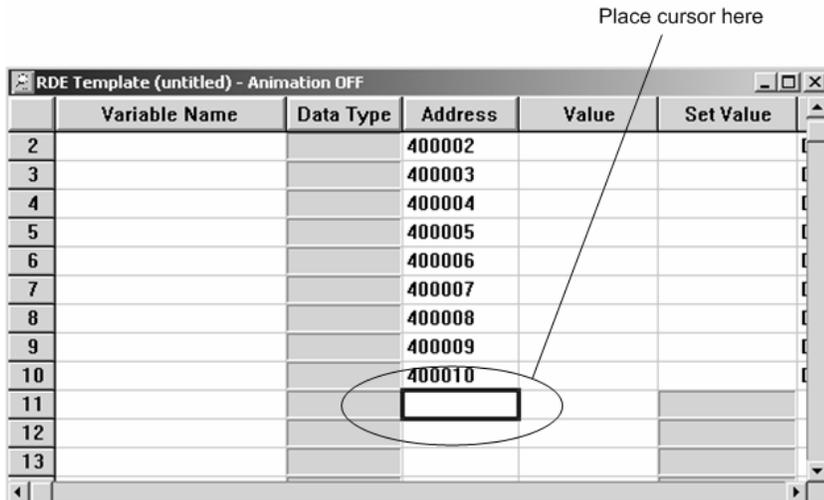
Last Reference To Insert: 400010

Number of References to Insert: 10

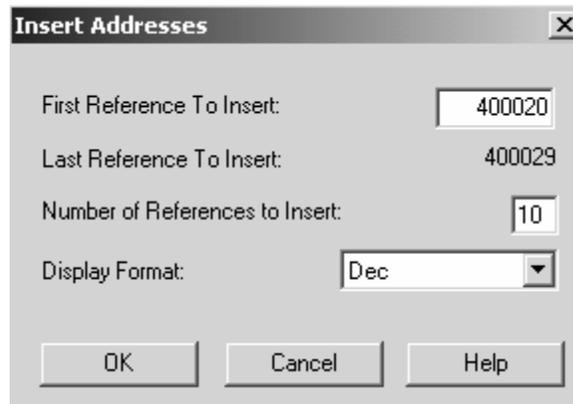
Display Format: Dec

OK Cancel Help

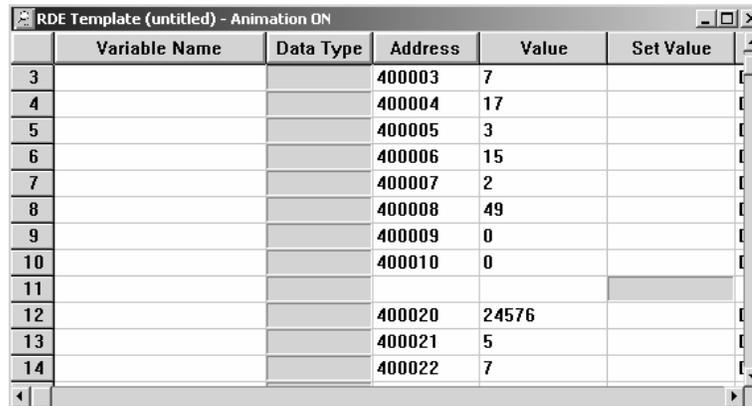
- 9 Notice that the template populates the address range, as shown in the following illustration. Place your cursor as shown in the first blank address field below the addresses you just entered.



- 10 Repeat steps 6 through 9, using the values in the following illustration:



- 11** In the PLC Configuration window, open the Online menu, and then choose animate. This action opens the RDE Template dialog box, with animated values in the Value field.



	Variable Name	Data Type	Address	Value	Set Value
3			400003	7	
4			400004	17	
5			400005	3	
6			400006	15	
7			400007	2	
8			400008	49	
9			400009	0	
10			400010	0	
11					
12			400020	24576	
13			400021	5	
14			400022	7	

- 12** Verify that values shown are cycling, starting from address 400065 on up.
- 13** In the PLC Configuration window, open the Templates menu, and then choose Save Template as. Name the template ptqclock, and then click OK to save the template.
- 14** In the PLC Configuration window, open the Online menu, and then choose Disconnect. At the disconnect message, click Yes to confirm your choice.

At this point, you have successfully

- Created and downloaded a Quantum project to the PLC
- Preset values in data registers that will later be monitored in the ProTalk module.

You are now ready to complete the installation and setup of the ProTalk module.



### 3 Configuring the Processor with ProWORX

When you use ProWORX 32 software to configure the processor, use the example SaF file provided on the ProTalk Solutions CD-ROM.

**Important Note:** Proworx software does not report whether the PTQ module is present in the rack, and therefore is not able to report the health status of the module when the module is online with the Quantum processor. Please take this into account when monitoring the status of the PTQ module.

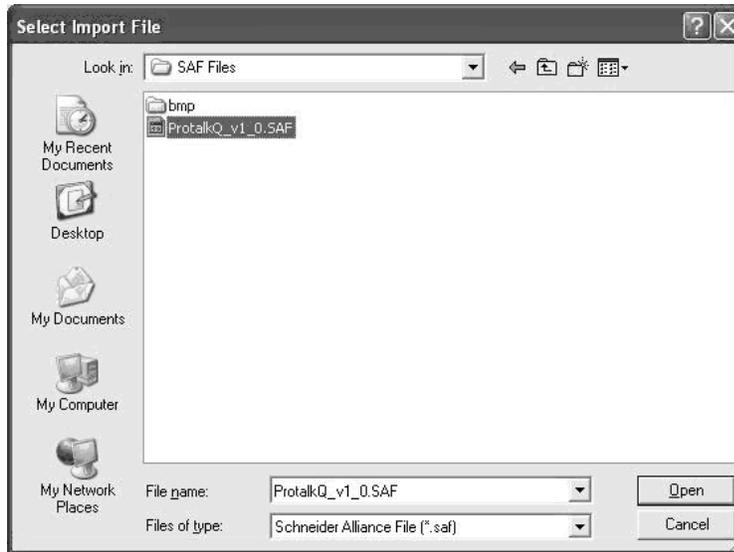
- 1 Run the Schneider\_alliances.exe application that is installed with the Proworx 32 software:



- 2 Click on Import...



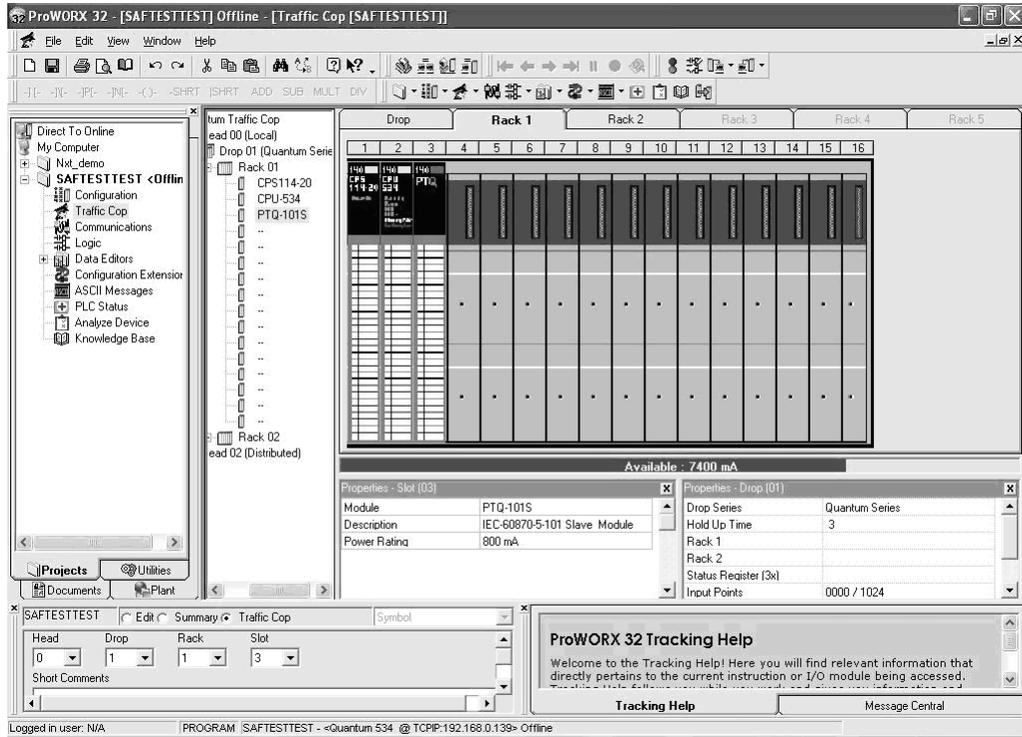
- 3 Select the .SaF File that is located at the CD-ROM shipped with the PTQ module.



- 4 After you click on Open you should see the PTQ modules imported (select I/O series as Quantum):



Now you can close the Schneider alliances application and run the Proworx 32 software. At the Traffic Cop section, select the PTQ module to be inserted at the slot:





## 4 Configuring the Processor with UnityPro XL

### *In This Chapter*

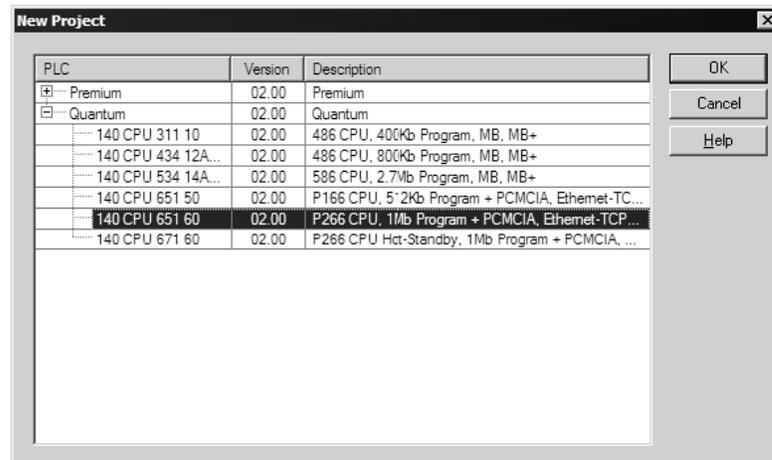
- ❖ Create a New Project ..... 35
- ❖ Add the PTQ Module to the Project..... 37
- ❖ Build the Project ..... 39
- ❖ Connect Your PC to the Processor ..... 40
- ❖ Download the Project to the Processor ..... 42

The following steps are designed to ensure that the processor (Quantum or Unity) is able to transfer data successfully with the PTQ module. As part of this procedure, you will use UnityPro XL to create a project, add the PTQ module to the project, set up data memory for the project, and then download the project to the processor.

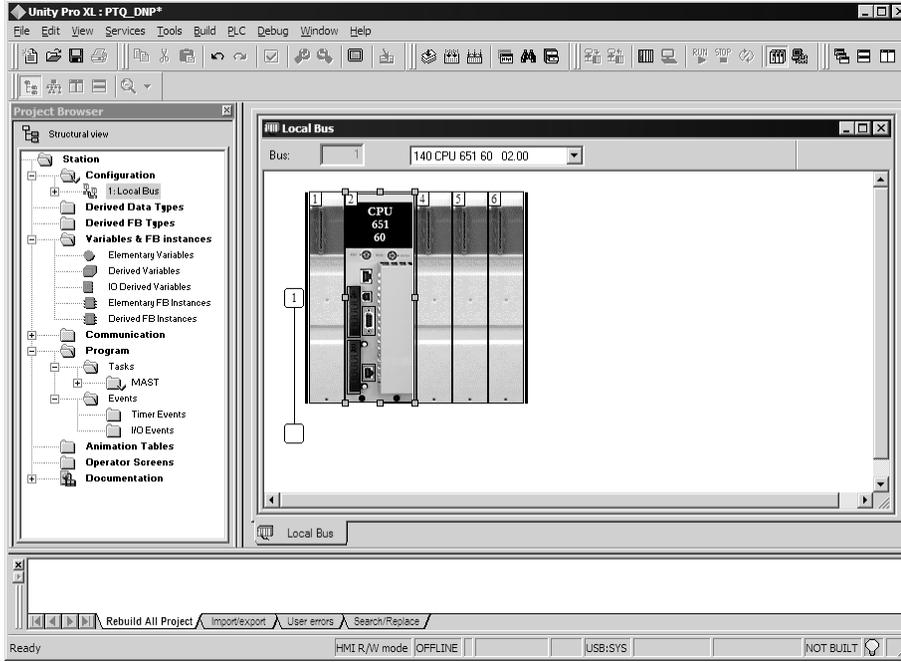
### 4.1 Create a New Project

The first step is to open UnityPro XL and create a new project.

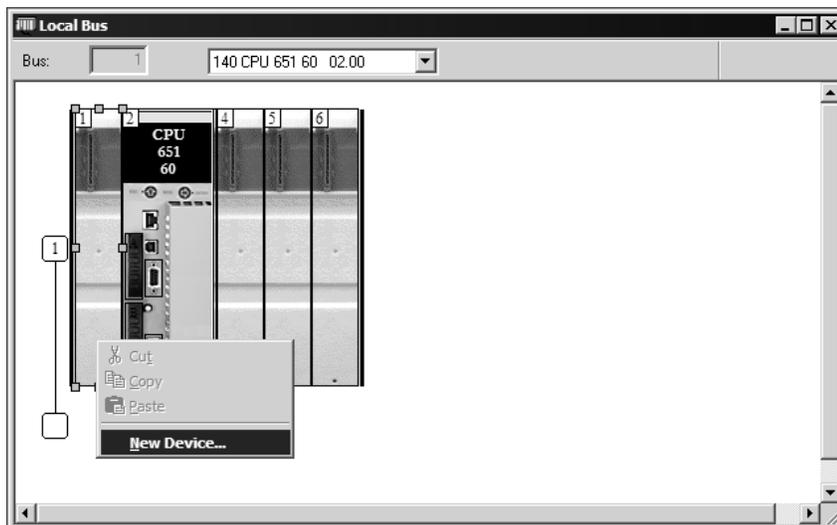
- 1 In the New Project dialog box, choose the CPU type. In the following illustration, the CPU is 140 CPU 651 60. Choose the processor type that matches your own hardware configuration, if it differs from the example. Click OK to continue.



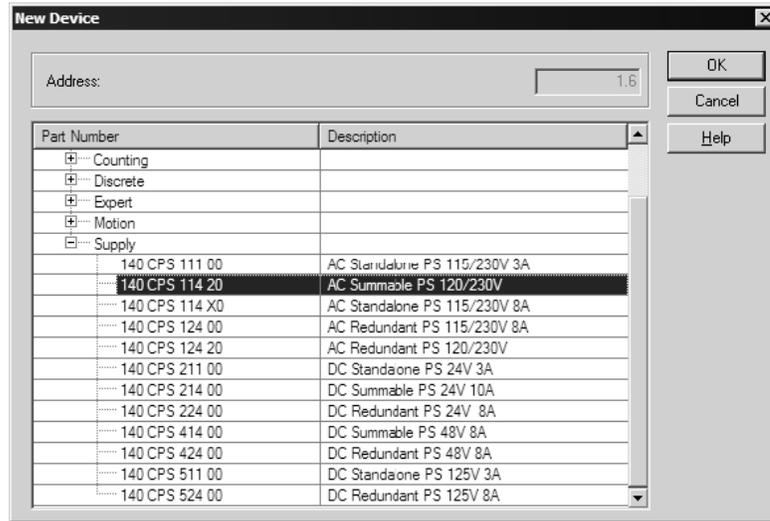
- The next step is to add a power supply to the project. In the Project Browser, expand the Configuration folder, and then double-click the 1:LocalBus icon. This action opens a graphical window showing the arrangement of devices in your Quantum rack.



- Select the rack position for the power supply, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose New Device..



- Expand the Supply folder, and then select your power supply from the list. Click OK to continue.

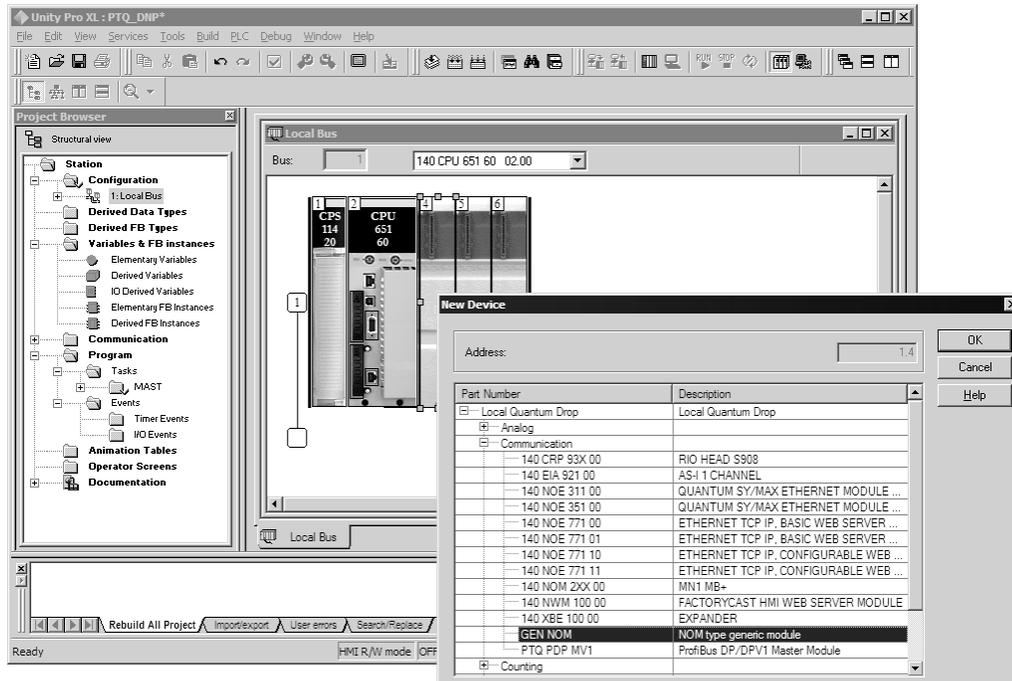


- Repeat these steps to add any additional devices to your Quantum Rack.

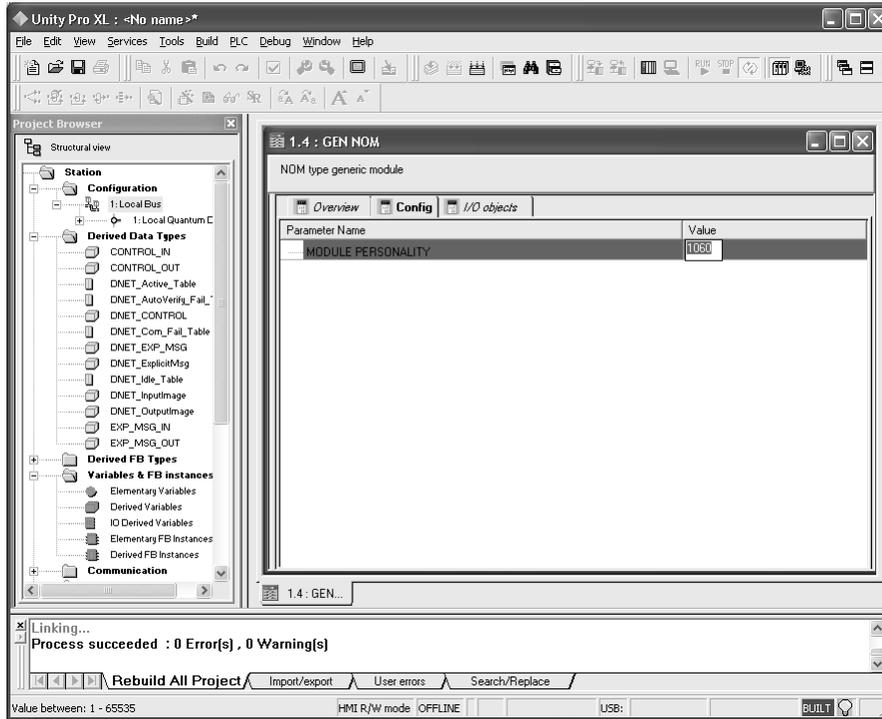
## 4.2 Add the PTQ Module to the Project

The next step is to add the PTQ module.

- Expand the Communication tree, and select GEN NOM. This module type provides extended communication capabilities for the Quantum system, and allows communication between the PLC and the PTQ module without requiring additional programming.



- Next, enter the module personality value. The correct value for ProTalk modules is 1060 decimal (0424 hex).



- Before you can save the project in UnityProXL, you must validate the modifications. Open the Edit menu, and then choose Validate. If no errors are reported, you can save the project.
- Save the project.

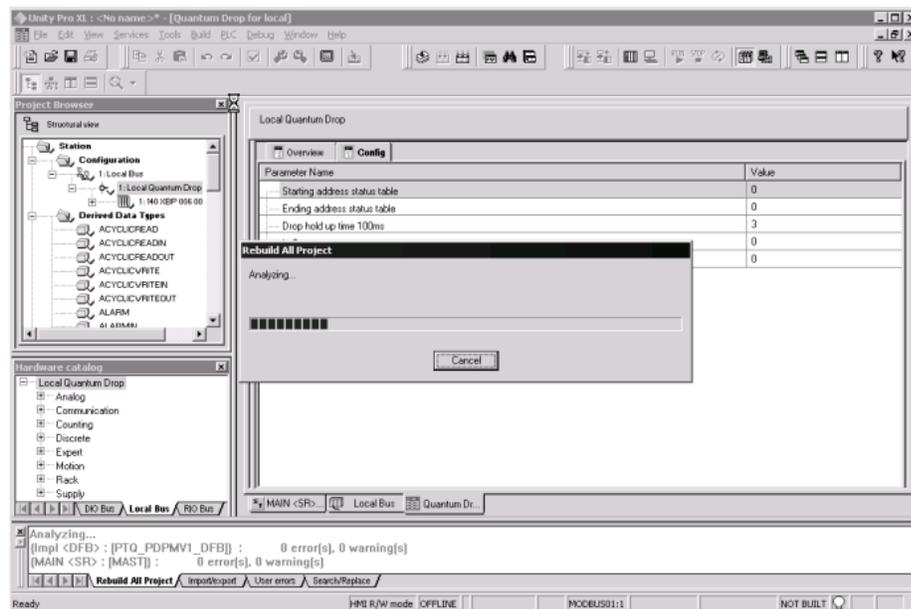
### 4.3 Build the Project

Whenever you update the configuration of your PTQ module or the processor, you must import the changed configuration from the module, and then build (compile) the project before downloading it to the processor.

**Note:** The following steps show you how to build the project in Unity Pro XL. This is not intended to provide detailed information on using Unity Pro XL, or debugging your programs. Refer to the documentation for your processor and for Unity Pro XL for specialized information.

**To build (compile) the project:**

- 1 Review the elements of the project in the Project Browser.
- 2 When you are satisfied that you are ready to download the project, open the Build menu, and then choose Rebuild all Project. This action builds (compiles) the project into a form that the processor can use to execute the instructions in the project file. This task may take several minutes, depending on the complexity of the project and the resources available on your PC.
- 3 As the project is built, Unity Pro XL reports its process in a Progress dialog box, with details appearing in a pane at the bottom of the window. The following illustration shows the build process under way.



After the build process is completed successfully, the next step is to download the compiled project to the processor.

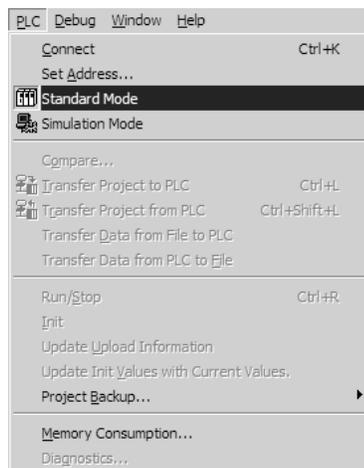
#### 4.4 Connect Your PC to the Processor

The next step is to connect to the processor so that you can download the project file. The processor uses this project file to communicate over the backplane to modules identified in the project file.

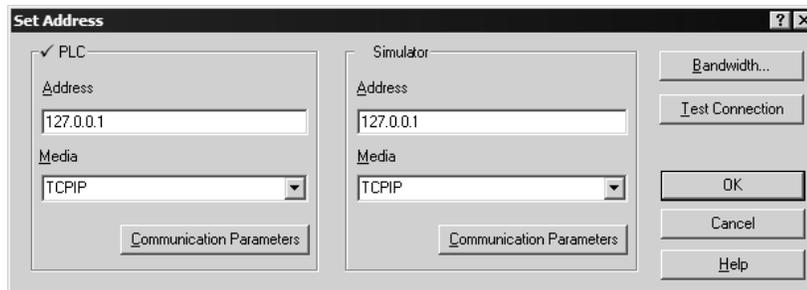
**Note:** If you have never connected from the PC to your processor before, you must verify that the necessary port drivers are installed and available to UnityPro XL.

To verify address and driver settings in UnityPro XL:

- 1 Open the PLC menu, and choose Standard Mode. This action turns off the PLC Simulator, and allows you to communicate directly with the Quantum or Unity hardware.



- 2 Open the PLC menu, and choose Set address... This action opens the Set address dialog box. Open the Media dropdown list and choose the connection type to use (TCPIP or USB).



- 3 If the Media dropdown list does not contain the connection method you wish to use, click the Communication Parameters button in the PLC area of the dialog box. This action opens the PLC Communication Parameters dialog box.



- 4 Click the Driver Settings button to open the SCHNEIDER Drivers management Properties dialog box.



- 5 Click the Install/update button to specify the location of the Setup.exe file containing the drivers to use. You will need your UnityPro XL installation disks for this step.

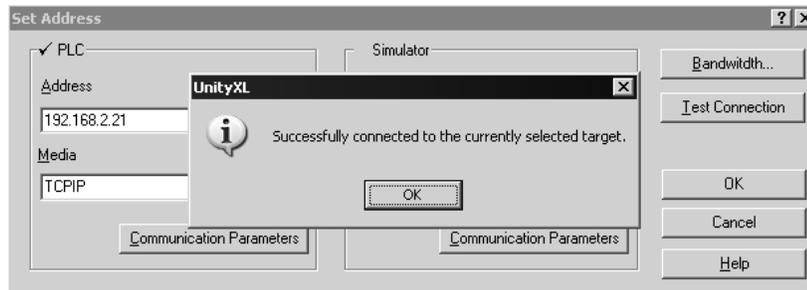


- 6 Click the Browse button to locate the Setup.exe file to execute, and then execute the setup program. After the installation, restart your PC if you are prompted to do so. Refer to your Schneider Electric documentation for more information on installing drivers for UnityPro XL.

#### 4.4.1 Connecting to the Processor with TCP/IP

The next step is to download (copy) the project file to the processor. The following steps demonstrate how to use an Ethernet cable connected from the Processor to your PC through an Ethernet hub or switch. Other connection methods may also be available, depending on the hardware configuration of your processor, and the communication drivers installed in UnityPro XL.

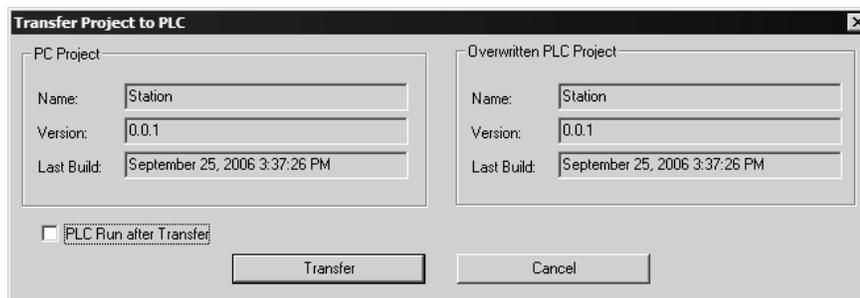
- 1 If you have not already done so, connect your PC and the processor to an Ethernet hub.
- 2 Open the PLC menu, and then choose Set address.
  - **Important:** Notice that the Set address dialog box is divided into two areas. Enter the address and media type in the PLC area of the dialog box, not the Simulator area.
- 3 Enter the IP address in the address field. In the Media dropdown list, choose TCP/IP.
- 4 Click the Test Connection button to verify that your settings are correct.



The next step is to download the Project to the Processor.

#### 4.5 Download the Project to the Processor

- 1 Open the PLC menu and then choose Connect. This action opens a connection between the Unity Pro XL software and the processor, using the address and media type settings you configured in the previous step.
- 2 On the PLC menu, choose Transfer Project to PLC. This action opens the Transfer Project to PLC dialog box. If you would like the PLC to go to "Run" mode immediately after the transfer is complete, select (check) the PLC Run after Transfer after Transfer check box.



- 3 Click the Transfer button to download the project to the processor. As the project is transferred, Unity Pro XL reports its process in a Progress dialog box, with details appearing in a pane at the bottom of the window.

When the transfer is complete, place the processor in Run mode.

## 5 Setting Up the ProTalk Module

### *In This Chapter*

- ❖ Install the ProTalk Module in the Quantum Rack ..... 43
- ❖ Connect the PC to the ProTalk Configuration/Debug Port ..... 45
- ❖ Verify Communication Between the Processor and the Module..... 47

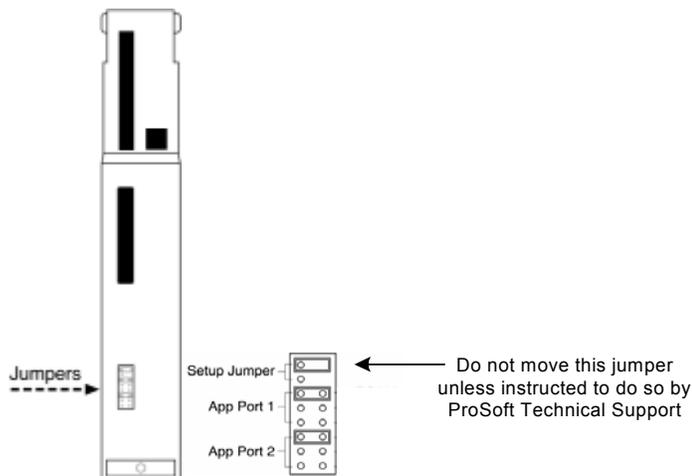
After you complete the following procedures, the ProTalk module will actively be transferring data bi-directionally with the processor.

### 5.1 Install the ProTalk Module in the Quantum Rack

#### 5.1.1 Verify Jumper Settings

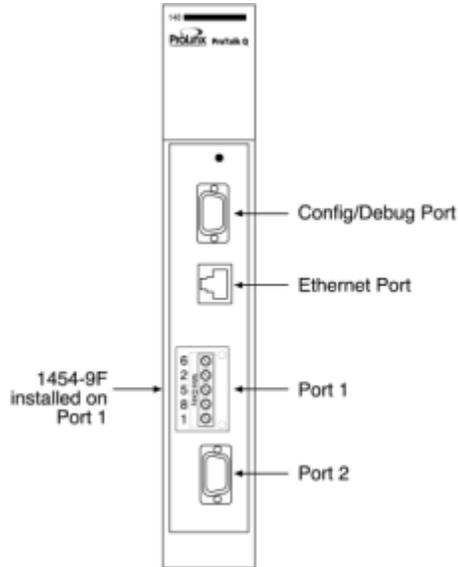
ProTalk modules are configured for RS-232 serial communications by default. To use RS-422 or RS-485, you must change the jumpers.

The jumpers are located on the back of the module as shown in the following illustration:



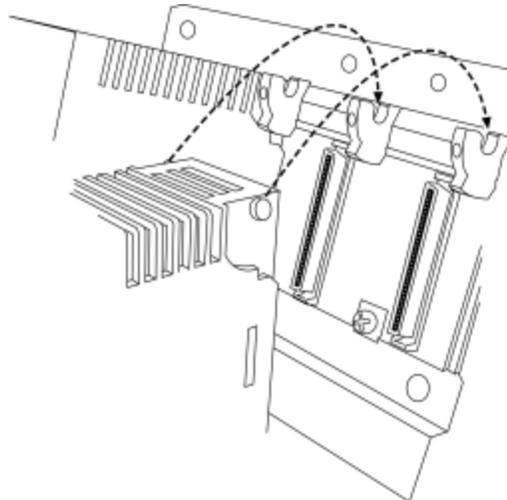
### 5.1.2 Inserting the 1454-9F connector

Insert the 1454-9F connector as shown. Wiring locations are shown in the table:

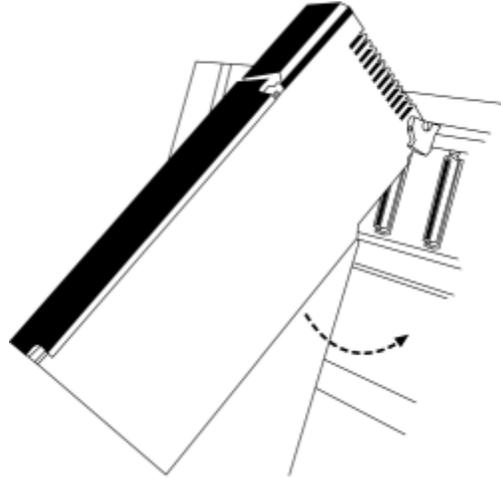


### 5.1.3 Install the ProTalk Module in the Quantum Rack

- 1 Place the Module in the Quantum Rack. The ProTalk module must be placed in the same rack as the processor.
- 2 Tilt the module at a 45° angle and align the pegs at the top of the module with slots on the backplane.



- 3 Push the module into place until it seats firmly in the backplane.

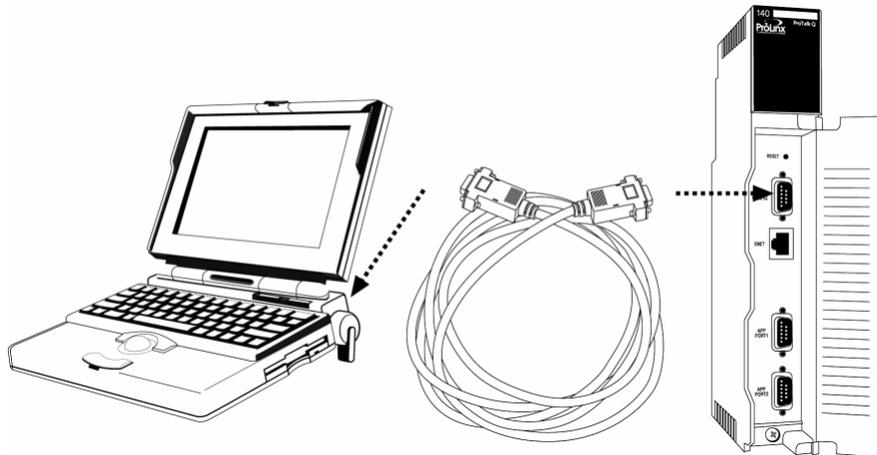


**CaUTION:** The PTQ module is hot-swappable, meaning that you can install and remove it while the rack is powered up. You should not assume that this is the case for all types of modules unless the user manual for the product explicitly states that the module is hot-swappable. Failure to observe this precaution could result in damage to the module and any equipment connected to it.

## 5.2 Connect the PC to the ProTalk Configuration/Debug Port

Make sure you have exited the Quantum programming software before performing these steps. This action will avoid serial port conflict.

- 1 Using the supplied Null Modem cable, connect your PC or Laptop to the Configuration/Debug port on the ProTalk module as shown



- 2 Click the Windows Start button, then choose Programs / accessories / Communications / HyperTerminal.

- 3 In the HyperTerminal window, enter a connection name, for example **Test**, and then click OK. This action opens the Connect To dialog box.



- 4 In the Connect Using field, ensure that the com port matches the port on your PC to which you connected the Null Modem cable, and then click OK. This action opens the COMx Properties dialog box.



- 5 Verify that the settings match those shown in the example above, and then click OK. If your port settings are configured correctly, you will return to the HyperTerminal window.
- 6 In the HyperTerminal window, press [?]. This action opens the module's Configuration/Debug menu.

### 5.3 Verify Communication Between the Processor and the Module

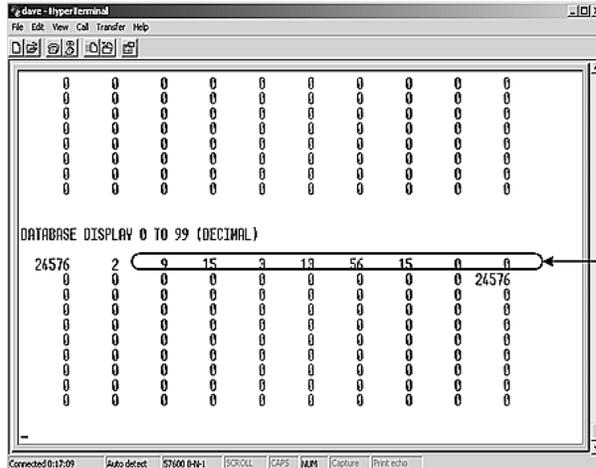
This procedure will verify that the clock values we entered in the processor's data memory (page 20) can be read into the ProTalk module.

- 1 From the *Configuration/Debug Menu*, type **[D]**, then press **[?]**. This action opens the *Database View Menu*.

```

DATABASE VIEW MENU
?=Display Menu
0-3=Pages 0 to 3000
S=Show Again
-=Back 5 Pages
P=Previous Page
+=Skip 5 Pages
N=Next Page
D=Decimal Display
H=Hexadecimal Display
F=Float Display
A=ASCII Display
M=Main Menu
    
```

- 2 Type **[0]** (zero). This displays values present in the ProTalk database for 0 to 99.



Shows date and time values set on the PLC

Value	Description
9	Month (September)
15	Day of the Month
3	Year (2003)
13	Hour (13:00 or 1:00 P.M.)
56	Minutes
15	Seconds

In this example, the register values read from the PLC indicate that the date and time returned is September, 15, 2003, 1:56:15 p.m.

- 3 Type [0] again. The values should be different than those shown in the previous view. For example, the minute and second values should be incrementing just as the values on the PLC are also incrementing.

At this point, you have successfully:

- Installed and set up the ProTalk module
- Verified Data Read access between the processor and the ProTalk module

You are now ready to proceed with implementation of your application.

## 6 Modifying the Configuration File

### *In This Chapter*

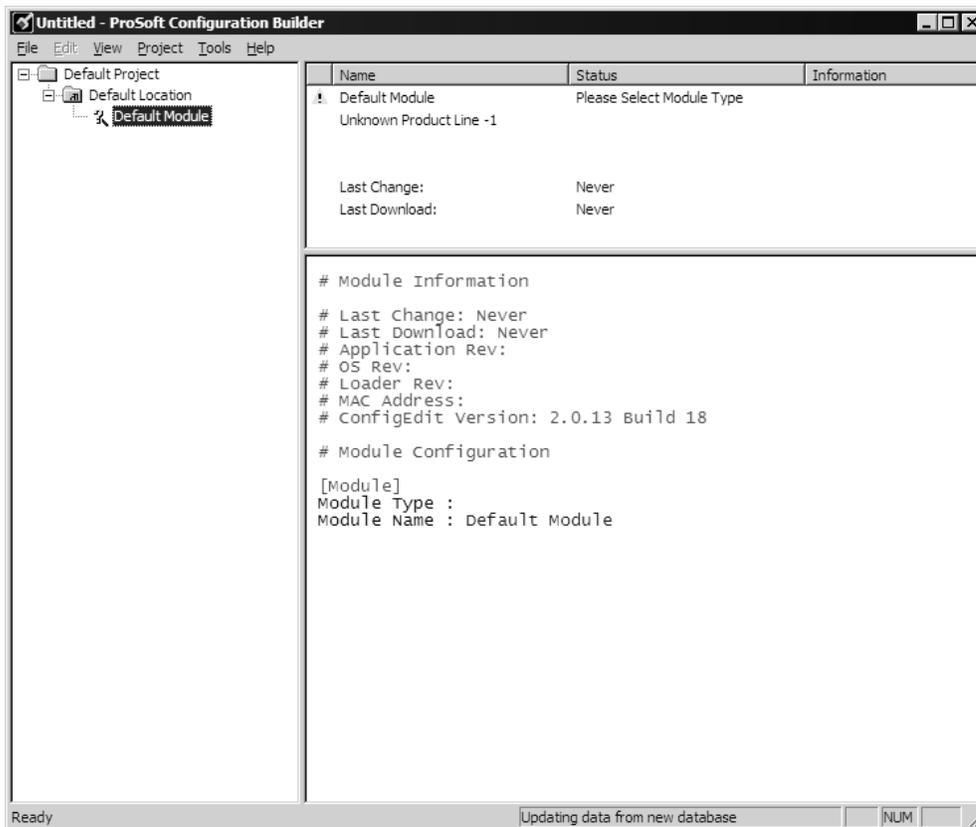
❖ ProSoft Configuration Builder .....	49
❖ [Module].....	54
❖ [BACKPLANE].....	54
❖ [Server n].....	55
❖ Ethernet Configuration .....	56
❖ Download the Project to the Module.....	57

### 6.1 ProSoft Configuration Builder

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

### 6.1.1 Set Up the Project

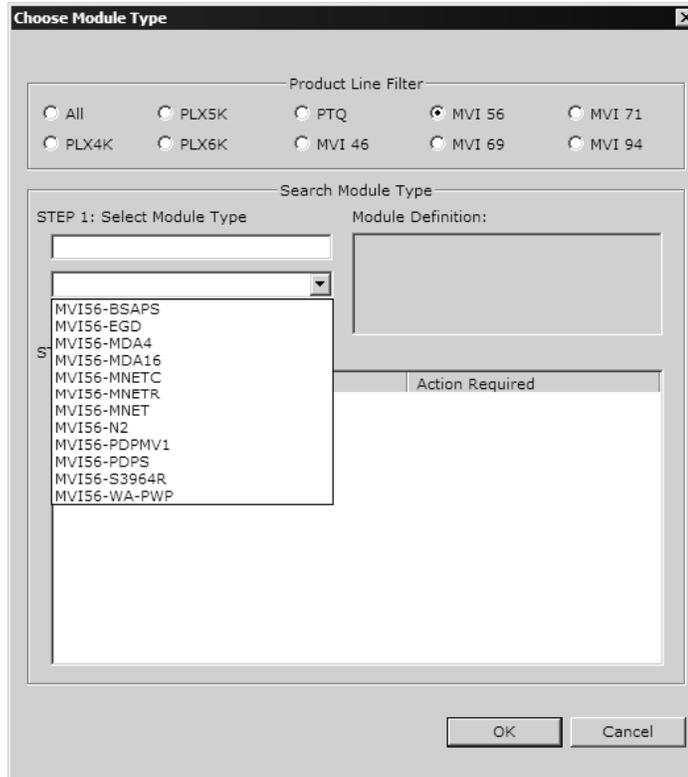
To begin, start ProSoft Configuration Builder. If you have used other Windows configuration tools before, you will find the screen layout familiar. ProSoft Configuration Builder's window consists of a tree view on the left, an information pane and a configuration pane on the right side of the window. When you first start ProSoft Configuration Builder, the tree view consists of folders for Default Project and Default Location, with a Default Module in the Default Location folder. The following illustration shows the ProSoft Configuration Builder window with a new project.



Your first task is to add the PTQ-GEC module to the project.

- 1 Use the mouse to select "Default Module" in the tree view, and then click the right mouse button to open a shortcut menu.

- 2 On the shortcut menu, choose "Choose Module Type". This action opens the Choose Module Type dialog box.



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

The next task is to set the module parameters.

### Adding a Project

#### To add a project to an existing project file:

- 1 Select the Default Project icon.
- 2 Choose Project from the Project menu, then choose Add Project. A new project folder appears.

### Adding a Module

#### To add a module to your project:

- 1 Double-click the Default Module icon to open the Choose Module Type dialog box.
- 2 On the Choose Module Type dialog box, select the module type.

Or

- 1 Open the Project menu and choose Location.

- 2 On the Location menu, choose Add Module.

To add a module to a different location:

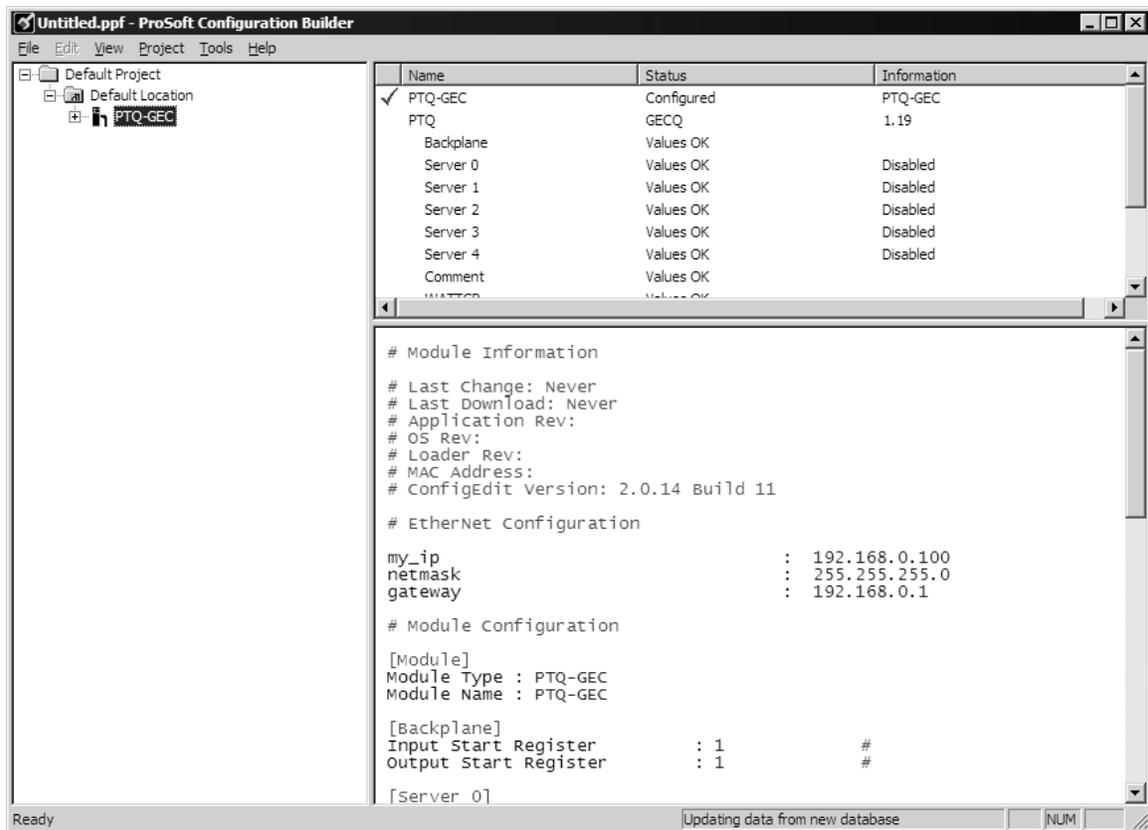
- 1 Right-click the Location folder and choose Add Module. A new module icon appears.

Or

- 1 Select the Location icon.
- 2 From the Project menu, select Location, then select Add Module.

### 6.1.2 Set Module Parameters

Notice that the contents of the information pane and the configuration pane changed when you added the PTQ-GEC module to the project.



At this time, you may wish to rename the "Default Project" and "Default Location" folders in the tree view.

To rename an object:

- 1 Select the object, and then click the right mouse button to open a shortcut menu. From the shortcut menu, choose Rename.
- 2 Type the name to assign to the object.
- 3 Click away from the object to save the new name.

### Module Entries

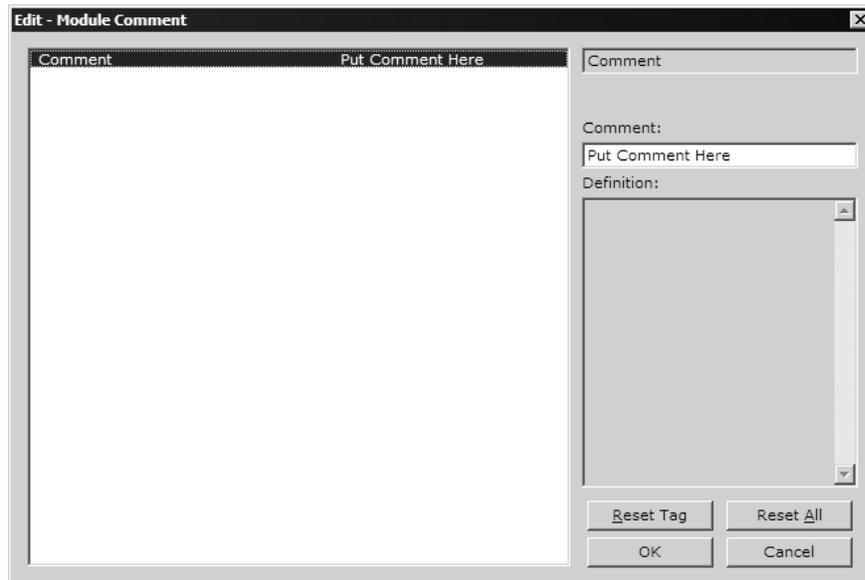
#### To configure module parameters

- 1 Click on the plus sign next to the icon  Comment to expand module information.
- 2 Double-click the  Module Comment icon to open the Edit dialog box.
- 3 To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
- 4 Click OK to save your changes.

### Comment Entries

#### To add comments to your configuration file:

- 1 Click the plus sign to the left of the  Comment icon to expand the Module Comments.
- 2 Double-click the  Module Comment icon. The Edit - Module Comment dialog appears.



- 3 Enter your comment and click OK to save your changes.

### Printing a Configuration File

#### To print a configuration file:

- 1 Select the Module icon, and then click the right mouse button to open a shortcut menu.
- 2 On the shortcut menu, choose View Configuration. This action opens the View Configuration window.
- 3 On the View Configuration window, open the File menu, and choose Print. This action opens the Print dialog box.
- 4 On the Print dialog box, choose the printer to use from the dropdown list, select printing options, and then click OK.

## 6.2 [Module]

This section provides the module with a unique name..

The following example shows a sample [Module] section:

```
# This section of the file describes the name of the module.
[Module]
Module Name: PTQ-GEC Communication Module DEFAULT
```

### 6.2.1 Module Name Parameter

The Module Name parameter assigns a name to the module that can be viewed using the configuration/debug port. Use this parameter to identify the module and the configuration file. You can enter a name from 0 to 80 characters.

## 6.3 [BACKPLANE]

This section assigns database addresses for input and output on the module and on the processor.

The following example shows a sample [Backplane] section:

```
# This section defines the state RAM areas in the processor the module
# will interface.
[PTQ BACKPLANE]
3x Register Start: 1 #3x start register where data moved from module to
processor (1-n)
4x Register Start: 1 #4x start register where data moved from processor to
module (1-n)
```

### 6.3.1 3x Register Start

```
3x Register Start : 1 #3x start register where data moved from module
#to processor (1-n)
```

The 3x Register Start parameter defines the starting address in the processor's 3x (Quantum) or %iw (Unity) memory area to use for data being moved from the module. Take care to use a starting address that will accommodate the entire block from the module, but that will not overwrite data that is used for other purposes.

### 6.3.2 4x Register Start

```
4x Register Start : 2 #4x start register where data moved from
#processor to module (1-n)
```

The 4x Register Start parameter defines the starting address in the processor's 4x (Quantum) or %iw (Unity) memory area to use for data being moved from the processor to the module. Take care to use a starting address that does not contain data in the processor's registers that is used for other purposes.

## 6.4 [Server n]

The following shows a sample Server 0 section of the configuration file.

```
[Server 0]
Enabled           : Yes  #(Y/N) Server is enabled
Service Port Number : 2201 #TCP port for this service
Connection Timeout : 0    #0=no timeout,
                    #else # of mSec before disconnect
Connection Close Type : 0    #0=Client will close socket,
                    #1=close after Tx
                    #2=Close after Rx
Swap Rx Data Bytes : No    #Swap bytes in message received (Y/N)
Swap Tx Data Bytes : No    #Swap bytes in message transmitted (Y/N)
```

### 6.4.1 Enabled

Yes or No

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

### 6.4.2 Service Port

1 to 65535

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

### 6.4.3 Connection Timeout

0 to 65535

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

### 6.4.4 Connection Close Type

0,1 or 2

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

### 6.4.5 Swap RX Data Bytes

Yes or No

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

### 6.4.6 Swap TX Data Bytes

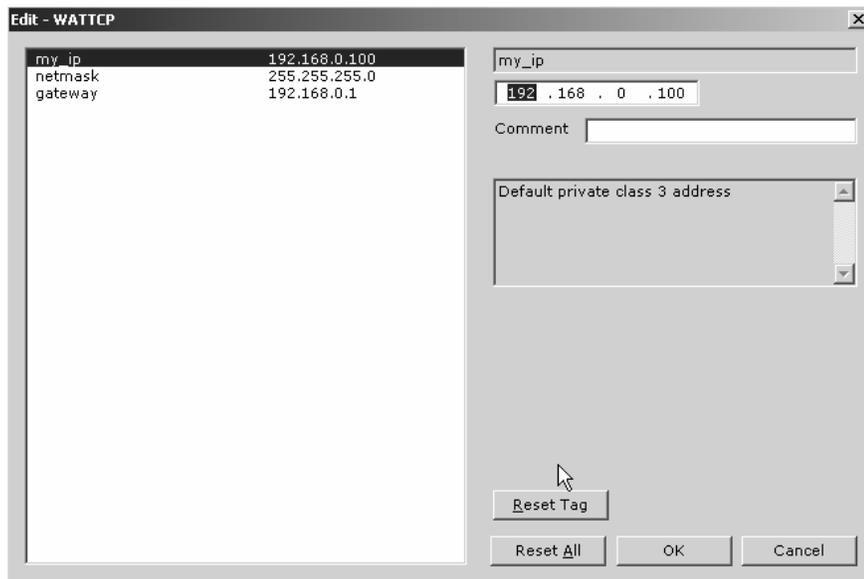
Yes or No

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

## 6.5 Ethernet Configuration

Use this procedure to configure the Ethernet settings for your module. You must assign an IP address, subnet mask and gateway address. After you complete this step, you can connect to the module with an Ethernet cable.

- 1 Determine the network settings for your module, with the help of your network administrator if necessary. You will need the following information:
  - o IP address (fixed IP required) \_\_\_\_\_ . \_\_\_\_\_ . \_\_\_\_\_ . \_\_\_\_\_
  - o Subnet mask \_\_\_\_\_ . \_\_\_\_\_ . \_\_\_\_\_ . \_\_\_\_\_
- 2 Gateway address \_\_\_\_\_ . \_\_\_\_\_ . \_\_\_\_\_ . \_\_\_\_\_ Click [+] to expand the tree for the PTQ-GEC module.
- 3 Double-click the Ethernet Configuration object. This action opens the Edit dialog box.



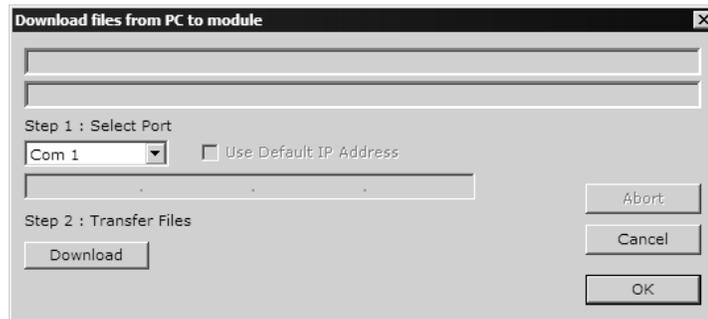
- 4 Edit the values for my\_ip, netmask (subnet mask) and gateway (default gateway).
- 5 When you are finished editing, click OK to save your changes and return to the ProSoft Configuration Builder window.

## 6.6 Download the Project to the Module

In order for the module to use the settings you configured, you must download (copy) the updated Project file from your PC to the module.

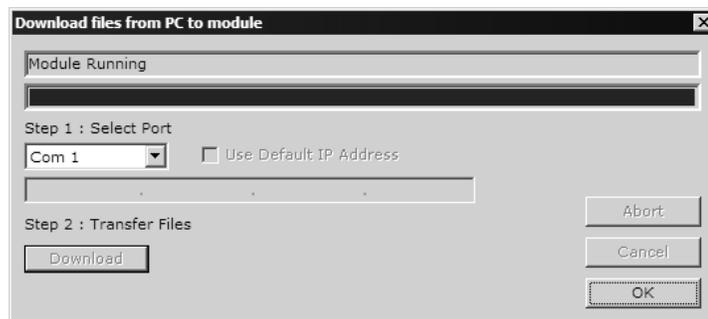
### To Download the Project File

- 1 In the tree view in ProSoft Configuration Builder, click once to select the PTQ-GEC module.
- 2 Open the **Project** menu, and then choose **Module / Download**. The program will scan your PC for a valid com port (this may take a few seconds). When PCB has found a valid com port, the following dialog box will open.



- 3 Choose the com port to use from the dropdown list, and then click the Download button.

The module will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in ProSoft Configuration Builder will be updated with the message *"Module Running"*.





## 7 Diagnostics and Troubleshooting

### *In This Chapter*

- ❖ Reading Status Data from the Module ..... 59
- ❖ LED Status Indicators..... 66
- ❖ Ethernet LED Indicators ..... 67

The module provides information on diagnostics and troubleshooting in the following forms:

- Status data values are transferred from the module to the processor.
- Data contained in the module can be viewed through the Configuration/Debug port attached to a terminal emulator.
- LED status indicators on the front of the module provide information on the module's status.

### 7.1 Reading Status Data from the Module

The PTQ-GEC module provides the status data in each read block. This data can also be located in the module's database and will be transferred to the Quantum/Unity processor continuously with each read block.

The Configuration/Debug port provides the following functionality:

- Full view of the module's configuration data
- View of the module's status data
- Version Information
- Control over the module (warm boot and cold boot)
- Facility to upload and download the module's configuration file

#### **7.1.1 Required Hardware**

You can connect directly from your computer's serial port to the serial port on the module to view configuration information, perform maintenance, and send (upload) or receive (download) configuration files.

ProSoft Technology recommends the following minimum hardware to connect your computer to the module:

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.
- A null modem serial cable.

### 7.1.2 The Configuration/Debug Menu

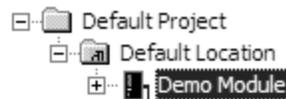
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in the diagnostic window in ProSoft Configuration Builder (PCB). The module does not respond to mouse movements or clicks. The command executes as soon as you press the command letter — you do not need to press **[Enter]**. When you type a command letter, a new screen will be displayed in your terminal application.

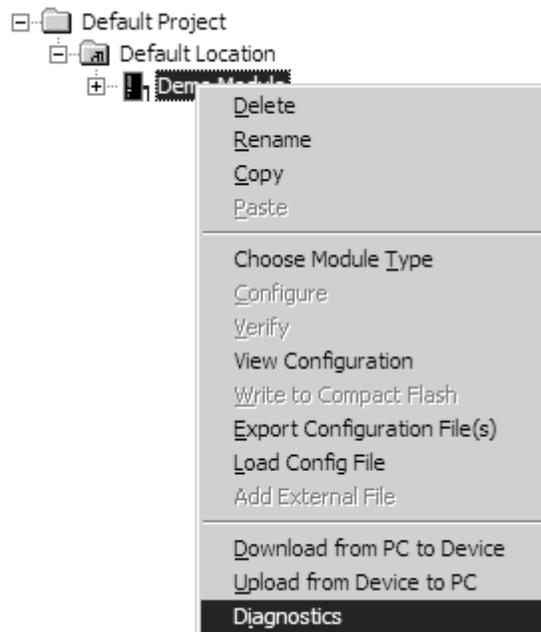
#### *Using the Diagnostic Window in ProSoft Configuration Builder*

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

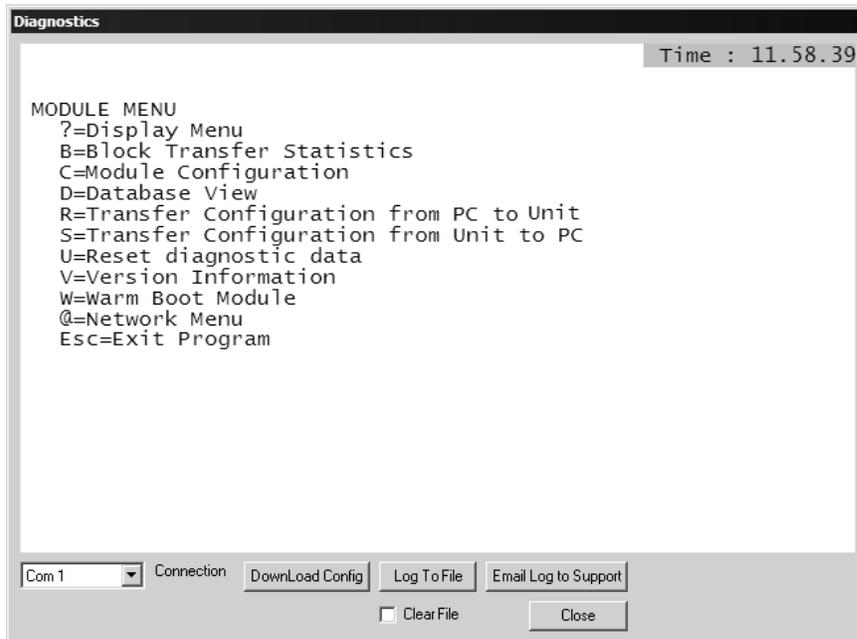
- 1 Start PCB program with the application file to be tested. Right click over the module icon.



- 2 On the shortcut menu, choose Diagnostics.



- 3 This action opens the Diagnostics dialog box. Press "?" to display the Main Menu.



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

If there is no response from the module, follow these steps:

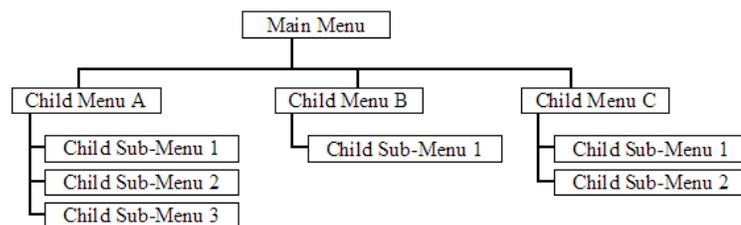
- 1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
- 2 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

### Navigation

All of the sub-menus for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a sub-menu to the next higher menu by pressing **[M]** on your keyboard.

The organization of the menu structure is represented in simplified form in the following illustration:



The remainder of this section shows you the menus available for this module, and briefly discusses the commands available to you.

### Keystrokes

The keyboard commands on these menus are almost always non-case sensitive. You can enter most commands in lower case or capital letters.

The menus use a few special characters ([**?**], [**-**], [**+**], [**@**]) that must be entered exactly as shown. Some of these characters will require you to use the [**Shift**], [**Ctrl**] or [**Alt**] keys to enter them correctly. For example, on US English keyboards, enter the [**?**] command as [**Shift**]/[**/**].

Also, take care to distinguish capital letter [**I**] from lower case letter [**i**] (L) and number [**1**]; likewise for capital letter [**O**] and number [**0**]. Although these characters look nearly the same on the screen, they perform different actions on the module.

### 7.1.3 Main Menu

When you first connect to the module from the computer, your terminal screen will be blank. To activate the main menu, press the [**?**] key on your computer's keyboard. If the module is connected properly, the following menu will appear on your terminal screen:

```
PTQ-GEC COMMUNICATION MODULE MENU
?=Display Menu
B=Block Transfer Statistics
C=Module Configuration
R=Transfer Configuration from PC to MVI Unit
S=Transfer Configuration from MVI Unit to PC
U=Reset diagnostic data
V=Version Information
W=Warm Boot Module
Status: 1=Server 0, 2=Server 1, 3=Server 2, 4=Server 3, 5=Server 4
        E=Client 0, F=Client 1, G=Client 2, H=Client 3, I=Client 4
Cfg    : 6=Server 0, 7=Server 1, 8=Server 2, 9=Server 3, 0=Server 4
@=Network Menu      Esc=Exit Program
```

**Caution:** Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

### Redisplaying the Menu

Press [**?**] to display the current menu. Use this command when you are looking at a screen of data, and want to view the menu choices available to you.

### Viewing Block Transfer Statistics

Press **[B]** to view the Block Transfer Statistics screen.

Use this command to display the configuration and statistics of the backplane data transfer operations between the module and the processor. The information on this screen can help determine if there are communication problems between the processor and the module.

**Tip:** Repeat this command at one-second intervals to determine the number of blocks transferred each second.

### Viewing Module Configuration

Press **[C]** to view the Module Configuration screen.

Use this command to display the current configuration and statistics for the module.

### Transferring the Configuration File from PC to PTQ module

Press **[R]** to send (upload) the configuration file from your PC to the module and store the file on the module's Compact Flash Disk.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully downloaded, the module will restart the program and load the new configuration information. Review the new configuration using menu commands **[6]** and **[0]** to verify that the module is configured correctly.

### Transferring the Configuration File from PTQ module to PC

Press **[S]** to receive (download) the configuration file from the module to your PC.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully downloaded, you can open and edit the file to change the module's configuration.

### Resetting diagnostic data

Press **[U]** to reset the status counters for the client and servers in the module.

### Viewing Version Information

Press **[V]** to view Version information for the module.

Use this command to view the current version of the software for the module, as well as other important values. You may be asked to provide this information when calling for technical support on the product.

Values at the bottom of the display are important in determining module operation. The Program Scan Counter value is incremented each time a module's program cycle is complete.

**Tip:** Repeat this command at one-second intervals to determine the frequency of program execution.

### Warm Booting the Module

**Caution:** Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Press **[W]** from the Main Menu to warm boot (restart) the module. This command will cause the program to exit and reload, refreshing configuration parameters that must be set on program initialization. Only use this command if you must force the module to re-boot.

### Viewing Server Communication Status (Servers 0 to 4)

Use commands **[1]** **[2]** **[3]** **[4]** or **[5]** to view the communication status and statistics of the specified server. This information can be useful when troubleshooting communication problems.

### Viewing Client Communication Status (Clients 10 to 14)

Use commands **[E]** **[F]** **[G]** **[H]** or **[I]** to view the communication status and statistics of the specified client. This information can be useful when troubleshooting communication problems.

### Viewing Server Configuration (Servers 0 to 4)

Use commands **[6]** **[7]** **[8]** **[9]** or **[0]** to view the configuration of the specified server.

### Opening the Network Menu

Press **[@]** to open the network menu. The network menu allows you to send, receive, and view the WATTCP.CFG file that contains IP, gateway, and other network specification information. You can find more information about the commands on this menu in the *Network Menu* section of this book.

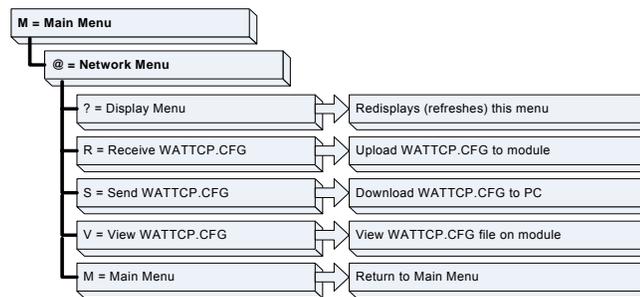
### Exiting the Program

**Caution:** Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Press **[Esc]** to restart the module and force all drivers to be loaded. The module will use the configuration stored in the module's Flash ROM to configure the module.

### **7.1.4 Network Menu**

The network menu allows you to send, receive and view the WATTCP.CFG file that contains the IP and gateway addresses, and other network specification information.



### Transferring WATTCP.CFG to the module

Press **[R]** to transfer a new WATTCP.CFG file from the PC to the module. Use this command to change the network configuration for the module (for example, the module's IP address).

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

### Transferring WATTCP.CFG to the PC

Press **[S]** to transfer the WATTCP.CFG file from the module to your PC.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully transferred, you can open and edit the file to change the module's network configuration.

Viewing the WATTCP.CFG file on the module

Press **[V]** to view the module's WATTCP.CFG file. Use this command to confirm the module's current network settings.

```

Network Menu Selected
WATTCP.CFG FILE:
# ProLinx Communication Gateways, Inc.
#
# Default private class 3 address
my_ip=192.168.0.135
#
# Default class 3 network mask
netmask=255.255.255.0
#
# The gateway I wish to use
gateway=192.168.0.1
#
#Parameters used by the ProLinx Communication Gateways, Inc. module
#Local_Domain_Name=mycompany.com
#Password=PASSWORD
    
```

Returning to the Main Menu

Press **[M]** to return to the Main Menu.

## 7.2 LED Status Indicators

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

ProTalk Module	Color	Status	Indication
PRT1	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
PRT2	Green	On	Port not used in application
		Off	Port not used in application
PRT3	Green	On	Port not used in application
		Off	Port not used in application
ERR1 P1 Fail	Red	Off	The PTQ-GEC is working normally.
		On	The PTQ-GEC module program has recognized an application error. This LED will also illuminate if any command presents an error.
ERR2 P2 Fail	N/A		Not used in application
ERR3 P2 Fail	Red	On	Configuration Error
Active	Green	On	The LED is on when the module recognizes a processor and is able to communicate if the [Backplane Data Movement] section specifies data transfer commands.
		Off	The LED is off when the module is unable to speak with the processor. The processor either absent or not running.

ProTalk Module	Color	Status	Indication
BAT Low	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or the battery is not present. The battery LED will illuminate briefly upon the first installation of the module or if the unit has been un-powered for an extended period of time. This behavior is normal, however should the LED come on in a working installation please contact ProSoft Technology.
E-Link	Green	On	The Ethernet port is connected to the TCP/IP network
		Off	No Connection
E-Data	Green	On	There is data being transferred through the Ethernet port.
		Off	No data transfer

If your module is not operating, and the status LEDs are not illustrated in the table above, please call ProSoft Technology for technical assistance.

### 7.3 Ethernet LED Indicators

LED	State	Description
Data	Off	No activity on the port.
	Green Flash	The port is either actively transmitting or receiving data.
Link	Off	No connection to hub or network is detected.
	Green Solid	Connected to hub or network correctly. This is the normal operating state.



## 8 Reference

### *In This Chapter*

❖ Product Specifications .....	69
❖ Functional Overview .....	71
❖ Cable Connections .....	82
❖ PTQ-GEC Status Data .....	83
❖ Configuration Data .....	89

### 8.1 Product Specifications

The Generic ASCII Ethernet Interface module is designed to allow Schneider Electric Quantum I/O compatible processors to interface easily with ASCII devices using the TCP/IP protocol and the Quantum processor. Compatible devices may be either ASCII instruments with Ethernet built-in or Ethernet connection via a thin server to the existing ASCII device.

#### **8.1.1 Features and Benefits**

Five servers and clients are present on the module permitting both the reception and transmission of data between the Quantum or Unity processor and attached devices.

The PTQ-GEC module is a powerful communication interface for Quantum platform processors. Developed under license from Schneider Electric, the module incorporates proprietary backplane technology that enables powerful data access between the module and the Quantum processor.

#### **8.1.2 General Specifications**

- Single Slot - Quantum backplane compatible
- The module is recognized as an Options module and has access to PLC memory for data transfer
- Configuration data is stored in non-volatile memory in the ProTalk module
- Up to six modules can be placed in a rack
- Local rack - The module must be placed in the same rack as processor
- Compatible with common Quantum / Unity programming tools
- Quantum data types supported: 3x, 4x
- High speed data transfer across backplane provides quick data update times
- Sample ladder file available

### 8.1.3 Hardware Specifications

Specification	Value
Backplane Current Load	800 mA @ 5 V
Operating Temperature	0 to 60°C (32 to 140°F)
Storage Temperature	-40 to 85°C (-40 to 185°F)
Relative Humidity	5% to 95% (non-condensing)
Vibration	Sine vibration 4-100 Hz in each of the 3 orthogonal axes
Shock	30G, 11 mSec. in each of the 3 orthogonal axes
LED Indicators	Module Status Backplane Transfer Status Serial Port Activity LED Serial Activity and Error LED Status
Configuration Serial Port (PRT1)	DB-9M PC Compatible RS-232 only No hardware handshaking
Application Serial Ports	(PRT2, PRT3) DB-9M PC Compatible RS-232/422/485 jumper selectable RS-422/485 screw termination included RS-232 handshaking configurable 500V Optical isolation from backplane
Application Ethernet port	RJ45 Connector Link and Activity LED indicators Electrical Isolation 1500 V rms at 50 Hz to 60 Hz for 60 s, applied as specified in section 5.3.2 of IEC 60950: 1991 Ethernet Broadcast Storm Resiliency = less than or equal to 5000 [ARP] frames-per-second and less than or equal to 5 minutes duration

### 8.1.4 Functional Specifications

Functional specification for this module are as follows:

- Five servers to receive and/or transmit data
- 10/100 Base-T Ethernet-compatible interface
- Configurable parameters
  - Service port number
  - Connection timeout
  - Close type
- Simple ladder logic operation
- Setup and monitoring through Concept or Unity Pro XL software and user-constructed configuration file (GEC.CFG)
- Quantum or Unity backplane interface via I/O access

- Configuration/Debug Port (CFG)
  - Module configuration
  - Operates at 57600, no parity, 1 stop (fixed), 8 data bits
  - Communication and backplane transfer stats
  - Server statistics and configuration
  - Warm boot
  - Configuration file transfer
  - Network configuration (includes the module IP address, mask, gateway address)
- Each server monitors
  - State
  - IP and port number of connected client
  - Number of messages received and transmitted
  - Error codes
- ASCII character strings up to 2048 characters in length supported
- Full hardware handshaking control, providing radio, modem, and multi-drop support
- User-definable module memory usage, supporting the storage and transfer of up to 4000 bytes to/from the control processor
- Module error and status conditions returned to processor for diagnostic purposes
  - Module status
  - Port error status word (bit mapped)
  - Port receive state
  - Port receive character count
  - Port receive block count
  - Port transmit state
  - Port transmit character count
  - Port transmit block count
- All data related to the module is contained in a single controller tag with defined objects to simplify configuration, monitoring, and interfacing with the module
- Module configuration and communication configuration data is transferred to the PTQ-GEC via a pre-defined user data type in the processor

## 8.2 Functional Overview

This section provides an overview of how the PTQ-GEC module transfers data using the GEC protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

### 8.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the PTQ-GEC module.

Module Power Up

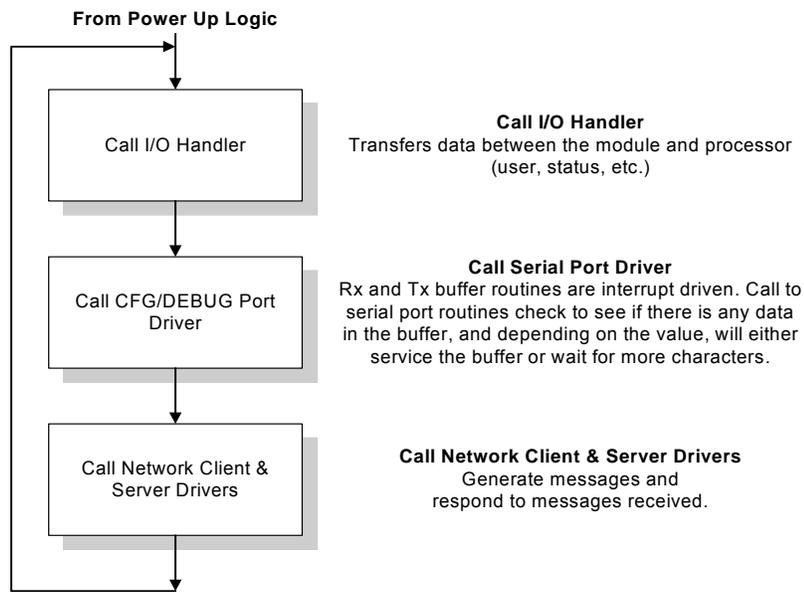
On power up the module begins performing the following logical functions:

- 1 Initialize hardware components
  - o Initialize Quantum / Unity backplane driver
  - o Test and clear all RAM
  - o Initialize the serial communication ports
  - o Initialize the TCP/IP stack and Ethernet interface
- 2 Initialize servers and clients
- 3 Set up the serial communication interface for the debug/configuration port

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

Main Logic Loop

Upon completing the power up configuration process, the module enters an infinite loop that performs the functions shown in the following diagram.



## 8.2.2 Backplane Data Exchange

Before modifying the [Backplane] section of the configuration file, you must understand some important concepts. The following topics describe these concepts.

If you have used the parameters defined in the [Module] section, you have created the following memory map. We will use this map to explain how data transfer works between the processor and the PTQ module.

PTQ Memory Address	=	Application Memory Address
0	=	0
10	=	10
20	=	20
30	=	30
40	=	40
50	=	50
60	=	60
70	=	70
80	=	80
90	=	90
100	=	100
110	=	110
120	=	120
130	=	130
140	=	140
149	=	149
...		...
500	=	500
510	=	510
520	=	520
530	=	530
540	=	540
549	=	549

A thorough understanding of the information contained in this section is required for successful implementation of the module in a user application.

### Backplane Data Transfer

The current version of the PTQ-GEC backplane driver (version 2.10 or newer), uses a Large I/O model, which differs from previous versions of the backplane driver in that it transfers all of the data in the Read and Write databases between the module and the processor on every scan.

The configuration file defines the starting registers for read and write operations. The settings for input data and output data determine how many blocks should be transferred between the module and the processor.

### 8.2.3 Input and Output Blocks

#### 3X Read Blocks

These blocks of data transfer information from the module to the Quantum / Unity processors. The structure of the input image used to transfer this data is shown in the following table:

	<b>Word Offset</b>	<b>Description</b>
Received Data	0	Server/Client Number for data received. If the word contains a value of -1, no receive data is present. If the word contains a value from 0 to 4, the block contains data for one of the servers in the module. If the word contains a value of 10 to 14, the block contains data for one of the clients.
	1	Number of characters (0 to 200) in receive block (2 to 101). If the receive data in the module is larger than 200 bytes, multiple blocks will be transferred. Any block with a value of -1 in this field represents the first or continuation block and the block contains 200 bytes of data. The last block of data will contain a positive number in this field that represents the number of characters in the last block.
	2 to 101	200 bytes of data received for specified server or client.
Module Status	102 to 234	This data area contains the status data block. Each block transferred to the processor contains this set of information.
	235 to 246	Reserved for future use.
	247	Value for designated Server or Client number during writing process. (0 to 4) for Servers (10 to 14) for clients.
	248	Number of characters processed from last write block (0 to 400).
	249	Block Sequence Number (Bumped each scan by module)

The Block Sequence Number (word 249) is an index value used to signal to the processor that a new block is ready for processing. The ladder logic must recognize a change in this value and process the data encapsulated in the input image. If data is available for a server, the module passes a block containing data received and the complete status data area to the processor. The value at word 0 in the block contains one of the following values:

<b>Value</b>	<b>Description</b>
-1	No receive data in block
0	Server 0 data in block
1	Server 1 data in block
2	Server 2 data in block
3	Server 3 data in block
4	Server 4 data in block
10	Client 0 data in block
11	Client 1 data in block
12	Client 2 data in block
13	Client 3 data in block
14	Client 4 data in block

Word 1 of the input image determines the number of bytes in the data area of the block. This word only has significance if word 0 of the block is set to a valid server index value (0 to 4), or client index value (10 to 14).

### Multiple Blocks

If a server or client receives a message longer than 200 bytes, it must send the received message in multiple blocks to the processor. In this case, the byte count field of the block will be set to -1 for each block where more than 200 characters are being sent by the server. Each block with a byte count field of -1 contains 200 bytes of data. The data set is located in the block starting at word offset 2. When the last block of data to send by the server is less than or equal to 200 bytes, the byte count field will be set to a number from 1 to 200. This signifies to the processor that this is the last block. The ladder logic must handle data received on each server or client enabled in the module.

The module status data begins at word offset 102. The ladder logic should use the state value of each server to determine which servers have open connections (state value of 1). Messages can be sent from the ladder logic to any open connection.

Word 247 of the message informs the ladder logic of the associated server or client in which data should be placed.

Word 248 of the message informs the ladder logic of the number of bytes sent in the last write block that were processed by a server. The ladder logic should check to make sure all bytes sent to a server are processed. If not, data may be lost.

The last word of the input image (word 249) is the block sequence number. This word's value is changed each time the module has processed the previous output image or is sending new information to the processor. This value should be copied to the first word of the output image (write block sequence number). Using this sequence number, the arrival of new data can easily be recognized by both the processor and the module.

### 4X Write Block

These blocks of data transfer information from the processor to the module. The structure of the output image used to transfer this data is shown in the following table.

	<b>Word Offset</b>	<b>Description</b>
	0	Block Sequence Number (Read block number as set by module)
Transmit Data	1	Server/Client Number for data to transmit. If the word contains a value of -1, no transmit data is present. If the word contains a value from 0 to maximum number of servers -1, the block contains data to send from the specified server in the module. If the word contains a value from 10 to 14, then the block contains data to send from one of the clients.

Word Offset	Description
2	Number of characters to transmit to a server or client. Each block can transmit up to 400 bytes of data. If this word contains a value of -1, the block contains 400 bytes of data and more blocks of data are to follow. When the module received the last block containing a positive value representing the number of bytes in the block, the server or client will transmit the data to the end device.
3 to 202	Data to transmit on specified server or client.
203 to 212	Client connection request data.
213 to 235	Reserved for future use
236	Server[0] Control Word (1=Abort)
237	Server[1] Control Word (1=Abort)
238	Server[2] Control Word (1=Abort)
239	Server[3] Control Word (1=Abort)
240	Server[4] Control Word (1=Abort)
241	Client[0] Control Word
242	Client[1] Control Word
243	Client[2] Control Word
244	Client[3] Control Word
245	Client[4] Control Word
246	Server Control Word
247	Module Control Word contains a value of 0, -1, -2 or -3.

The Block Sequence Number is that received on the last read block transfer through the input image on the module. The ladder logic should copy this value from word 249 of the input image to word 0 of output image in the ladder logic.

**Note:** This is the last operation performed when constructing the write block. The module's program will trigger the process write block function when a new value is recognized in word 0 of the output image.

Words 236 to 240 are used for server socket control. If the server socket is established and the user sets the value in the appropriate word to 1, the server state will be changed to 1003 (new server state) and will abort the socket immediately. It will then reopen the socket to wait for a new connection. These words should be set to zero when no operation is required.

Block ID	Description
0 to 65535	Normal data blocks for read and write

Word 1 of the block contains one of the following values:

Value	Description
-1	No transmit data in block
0	Server 0 data in block
1	Server 1 data in block
2	Server 2 data in block
3	Server 3 data in block

Value	Description
4	Server 4 data in block
10	Client 0 data in block
11	Client 1 data in block
12	Client 2 data in block
13	Client 3 data in block
14	Client 4 data in block

If the word is set to a value of -1, there is no data in the message. A value of 0 to 4 will cause the enclosed message to be sent to the server if the message length is set to a value other than zero and the server has an open connection. A value of 10 to 14 will cause the enclosed message to be sent to the client if a message length is set to a value other than zero. Word 2 of the block defines the length of data in the block.

### Multiple Blocks

Up to 400 bytes of data can be sent in each block, starting at word offset 3 in the block. If the message to be sent to a server is longer than 400 bytes, multiple blocks are required. If more than 400 bytes are required to send, the message length field should be set to a value of -1 and 400 bytes of data should be placed in the block. When 400 or fewer bytes remain to be sent, the message length field should be set to the number of bytes remaining, and the remaining data placed in the block.

The client or server will accept messages up to a length of 8192 bytes. If a message longer than 8192 bytes is sent to a server, a transmit overflow error will be recorded and the message will be sent when 8192 bytes are received. The remaining part of the message will be sent as it is received.

Words 203 and 212 request a new connection for one of the five clients in the module. The format of this data area is shown in the following table:

Word	Description
203	Client to utilized for connection (10 to 14)
204	Reserved for future use.
205 to 208	IP address of server to which connection will be made. Each word contains one of the digits of a dotted notation IP address.
209	Service port in server to which connection will be made. This service must be available in the server for the connection to succeed.
210	Swap Rx data bytes (0 = No, not 0 = Yes)
211	Swap Tx data bytes (0 = No, not 0 = Yes)
212	Client Connection time out. (0=not utilized, not 0 = timeout at entered value)

Ladder logic is required to transfer this data to the block when a connection is required. The ladder logic should also clear the data area after the connection is requested. The example ladder logic described in Customizing the Sample Ladder Logic provides code to accomplish these tasks. A connection should only be requested if the connected member of the status data for the client has a value less than one.

Client Control Word Codes

Words 241 to 245 close the connection on one of the clients. The following table lists the values recognized by the module for these words:

Code	Definition
0	No operation to perform
1	Close the client socket
2	Abort the client socket

Server Control Word Codes

The last two words of the image control the server or the module. The server control word (word 246) controls the server. The following table lists the values recognized by the module:

Code	Definition
0	No operation to perform
1	Close socket after transmit operation
2	Abort socket after transmit operation

If a value of 0 is present in the field, no action is taken by the server. If a value of 1 is present in the field, the server will close the socket in the standard way, after it transmits the message contained in the block. If a value of 2 is present, the server will send the message contained in the block, and then force the connection closed by sending a reset message to the client.

Module Control Word Codes

Word 247 is used to control the module. The following table lists the values recognized by the module.

Code	Definition
0	No operation to perform
-1	Warm boot module
-2	Cold boot module

The module will perform the action specified in the command control word.

Handling Multiple Blocks

It is very important to understand the concept of how to handle multiple blocks. The buffer size supports 8192 bytes, but the module can handle only 2048 bytes. The module can only send 200 bytes at each scan to the processor. For example, if a device sends a message that contains 1000 bytes to the module, it will break it down into 5 blocks of 200 bytes. The first 4 blocks will set the number of characters parameter as -1, indicating that each block is part of the same message. The last block will have the number of characters set to 200, indicating that there are no more blocks from that message.

The concept is the same for writing data from the processor to the module, except that you can write up to 400 bytes at a time to the module. The module buffers all data until it receives a block that contains the "number of characters" parameter set to  $\geq 0$ . In this case, it sends all data to the client that is connected to the server, or the server connects to the client.

**Important:** Messages are usually broken down into smaller frames by the IP layer in a specific LAN or WAN, according to the Maximum Transmit Unit (MTU) of the network.

For example, a message that contains 2000 bytes can be broken down into two messages by the IP layer in the network (after it is sent to the module). The same concept is applied when a client sends data to the server; although a client sends a single message to the module, it could be broken down into smaller fragments before it gets to the module. In this case, the module would interpret it as two different messages.

The application layer defines when a message is finished. This why you should consider using some sort of control so the ladder can identify different messages as actually part of one single message. You can accomplish this either by using a specific character at the end of each message, or by using a fixed length for each message.

#### **8.2.4 Network Data Transfer**

In order for data to be transferred between the module and another device, a TCP/IP connection must be made between a client on the network, and a server on the module.

The module contains five servers that listen on the user-assigned service ports waiting for a connection. When a client device wishes to send data to the module, it must open a TCP/IP connection to the module. After the connection is established, either device can send and receive data. When either device is finished with the connection, the connection must be closed. This operation can be initiated from either the client on the network or the server on the module.

The module servers and clients are configured to handle their TCP/IP session independently. The user parameter Connection Timeout determines the amount of time a connection can remain idle before the server will close the connection. If the parameter is set to 0, the server will not perform the timeout logic, and the socket will never be closed by the server on an idle condition. You can use this setting to prevent connections that may be lost and were not properly closed.

Each server on the module is assigned its own server port number. This does not mean that two or more servers cannot share the same port number. In fact, this might be desirable in some instances. It is up to the ladder logic to keep track of each message, and to ensure that a request/response transaction is associated with the correct connection.

Information to keep track of each connection is passed in each input image. The status data set provides the IP address and TCP port address for the connection on each server.

Each message transferred between the module and the processor has a server index word. This word associates the message with a server, which is associated with a connection to a specific IP address and TCP port address. Therefore, each connection is specified to the processor by the server index.

The following table describes an example of the module's status data:

Server	Status Data	Description of Server
Server 0	IP of Host (192.168.0.100) Port of Host (1243) State = 1	This server is connected (State=1) to IP address 192.168.0.100 on TCP port 1243.
Server 1	IP of Host (192.168.0.100) Port of Host (1244) State = 1	This server is connected (State=1) to IP address 192.168.0.100 on TCP port 1244.
Server 2	IP of Host (192.168.0.101) Port of Host (56443) State = 1	This server is connected (State=1) to IP address 192.168.0.101 on TCP port 56443.
Server 3	IP of Host (192.168.0.102) Port of Host (7943) State = 1	This server is connected (State=1) to IP address 192.168.0.102 on TCP port 7943.
Server 4	IP of Host (0.0.0.0) Port of Host (0) State = 0	This server is not connected (State not equal to 1) and is waiting for a connection.

Ladder logic can send messages to the clients connected to servers 0 to 3. Messages sent to server 4 will not be sent from the module because there is no connection active on that server.

Each server has a state value in the status data area. This value is utilized by the ladder logic to determine if a connection is present on server. The following table defines the state status values used by each server:

Server state list

State Value	Definition
-1	Server is initializing and is being set up to listen.
0	The server is waiting for a client to establish a connection.
1	The server has established a connection with a client and can send or receive data.
1000	The server has initiated a close operation on the connection.

State Value	Definition
1001	The server is waiting for the close on the connection to complete.
1002	The server is issuing an abort (reset) on the connection. The socket is forced closed.
1003	The server is issuing an abort (reset) on the connection due to control from user.

#### Client state list

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

#### Client Connection States

State Value	Definition
-4	Client is disconnected by the connection timeout which was entered during connection setup.
-3	Server closed connection for client or server is not available.
-2	Unable to open connection with specified server.
-1	Unable to open connection with specified server because of invalid IP address.
0	The client is idle and not connected.
1	The client set to connect to the server and waiting for the connection to establish.
2	The client is connected to the server and can transfer data.
3	The connection is being closed for the client.

Ladder logic should only direct messages to servers that have a state status value of 1. The module will ignore all messages sent to servers with any other state value.

When the ladder logic sends a message to a server, it can request that the socket be closed after the message is sent. The server control word in the output image is used for this purpose. Place a value of 1 in this register to close the connection in the normal way after the message is sent. If a value of 2 is placed in the register, the server will abort the connection to force the socket closed (send a message with the Reset Flag set to the client). Most applications will have the client close the socket.

## 8.3 Cable Connections

The PTQ-GEC module has the following communication connections on the module:

- One Ethernet port (RJ45 connector)
- One RS-232 Configuration/Debug port (DB9 connector)

### 8.3.1 Ethernet Connection

The PTQ-GEC module has an RJ45 port located on the front of the module labeled "Ethernet", for use with the TCP/IP network. The module is connected to the Ethernet network using an Ethernet cable between the module's Ethernet port and an Ethernet switch or hub.

**Note:** Depending on hardware configuration, you may see more than one RJ45 port on the module. The Ethernet port is labeled "Ethernet".

**Warning:** The PTQ-GEC module is NOT compatible with Power Over Ethernet (IEEE802.3af / IEEE802.3at) networks. Do NOT connect the module to Ethernet devices, hubs, switches or networks that supply AC or DC power over the Ethernet cable. Failure to observe this precaution may result in damage to hardware, or injury to personnel.

**Important:** The module requires a static (fixed) IP address that is not shared with any other device on the Ethernet network. Obtain a list of suitable IP addresses from your network administrator BEFORE configuring the Ethernet port on this module.

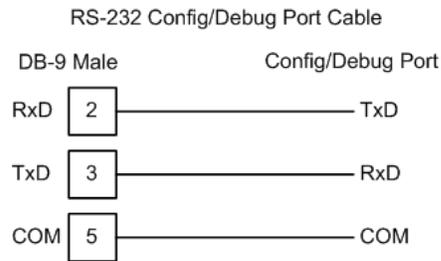
#### Ethernet Port Configuration - wattcp.cfg

The wattcp.cfg file must be set up properly in order to use a TCP/IP network connection. You can view the current network configuration using an ASCII terminal by selecting "@" (Network Menu) and "V" (View) options when connected to the Debug port.

```
# WATTCP.CFG FILE:  
# ProSoft Technology.  
my_ip=192.168.0.100  
# Default class 3 network mask  
netmask=255.255.255.0  
# The gateway I wish to use  
gateway=192.168.0.1
```

### 8.3.2 RS-232 Configuration/Debug Port

This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following diagram:



### 8.4 PTQ-GEC Status Data

This section contains a listing of the data contained in the status data object.

Object In GSCInStat	Block Offset Start	Description
PassCnt	102	Program cycle counter
Product	103	Product name as ASCII string
Rev	105	Revision level as ASCII string
OP	107	Operating system level as ASCII string
Run	109	Run number as ASCII string
BlkErrs.Read	111	Number of blocks transferred from module to processor
BlkErrs.Write	112	Number of blocks transferred from processor to module
BlkErrs.Parse	113	Number of blocks parsed by module
BlkErrs.Err	114	Number of block errors in module
Server[0].Enabled	115	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[0].State	116	This flag defines the current state of the server.
Server[0].IP	117	This double-word value contains the IP address of the client connected to the server.
Server[0].Port	119	This word value contains the port address for the client connected to the server.
Server[0].Open	120	Indicates that the module has opened the server port and is waiting for a connection to a client.
Server[0].Established	121	This status value contains the total number of times a connection was established on the socket.
Server[0].Closed	122	This status value contains the total number of times a close operation was performed on the socket.
Server[0].RxCount	123	This status value contains the total number of messages received by the server.

<b>Object In GSCInStat</b>	<b>Block Offset Start</b>	<b>Description</b>
Server[0].RxOverflow	124	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[0].TxCount	125	This status value contains the total number of messages transmitted by the server.
Server[0].TxOverflow	126	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[0].Timeout	127	This status value contains the total number of times a connection timeout occurred on the socket. This status value is incremented each time a connection timeout or a socket close timeout (2 seconds) occurs.
Server[0].CfgErrWord	128	This bit mapped word defines the configuration errors for the server.
Server[1].Enabled	129	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[1].State	130	This flag defines the current state of the server.
Server[1].IP	131	This double-word value contains the IP address of the client connected to the server.
Server[1].Port	133	This word value contains the port address for the client connected to the server.
Server[1].Open	134	Indicates that the module has opened the server port and is waiting for a connection to a client.
Server[1].Established	135	This status value contains the total number of times a connection was established on the socket.
Server[1].Closed	136	This status value contains the total number of times a close operation was performed on the socket.
Server[1].RxCount	137	This status value contains the total number of messages received by the server.
Server[1].RxOverflow	138	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[1].TxCount	139	This status value contains the total number of messages transmitted by the server.
Server[1].TxOverflow	140	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[1].Timeout	141	This status value contains the total number of times a connection timeout occurred on the socket. This status value is incremented each time a connection timeout or a socket close timeout (2 seconds) occurs.
Server[1].CfgErrWord	142	This bit mapped word defines the configuration errors for the server.
Server[2].Enabled	143	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[2].State	144	This flag defines the current state of the server.

<b>Object In GSCInStat</b>	<b>Block Offset Start</b>	<b>Description</b>
Server[2].IP	145	This double-word value contains the IP address of the client connected to the server.
Server[2].Port	147	This word value contains the port address for the client connected to the server.
Server[2].Open	148	Indicates that the module has opened the server port and is waiting for a connection to a client.
Server[2].Established	149	This status value contains the total number of times a connection was established on the socket.
Server[2].Closed	150	This status value contains the total number of times a close operation was performed on the socket.
Server[2].RxCount	151	This status value contains the total number of messages received by the server.
Server[2].RxOverflow	152	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[2].TxCount	153	This status value contains the total number of messages transmitted by the server.
Server[2].TxOverflow	154	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[2].Timeout	155	This status value contains the total number of times a connection timeout occurred on the socket. This status value is incremented each time a connection timeout or a socket close timeout (2 seconds) occurs.
Server[2].CfgErrWord	156	This bit mapped word defines the configuration errors for the server.
Server[3].Enabled	157	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[3].State	158	This flag defines the current state of the server.
Server[3].IP	159	This double-word value contains the IP address of the client connected to the server.
Server[3].Port	161	This word value contains the port address for the client connected to the server.
Server[3].Open	162	Indicates that the module has opened the server port and is waiting for a connection to a client.
Server[3].Established	163	This status value contains the total number of times a connection was established on the socket.
Server[3].Closed	164	This status value contains the total number of times a close operation was performed on the socket.
Server[3].RxCount	165	This status value contains the total number of messages received by the server.
Server[3].RxOverflow	166	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[3].TxCount	167	This status value contains the total number of messages transmitted by the server.

<b>Object In GSCInStat</b>	<b>Block Offset Start</b>	<b>Description</b>
Server[3].TxOverflow	168	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[3].Timeout	169	This status value contains the total number of times a connection timeout occurred on the socket. This status value is incremented each time a connection timeout or a socket close timeout (2 seconds) occurs.
Server[3].CfgErrWord	170	This bit mapped word defines the configuration errors for the server.
Server[4].Enabled	171	This flag defines if the server is utilized. A value of 0 indicates the server is not used. Any other value indicates the server is used.
Server[4].State	172	This flag defines the current state of the server.
Server[4].IP	173	This double-word value contains the IP address of the client connected to the server.
Server[4].Port	175	This word value contains the port address for the client connected to the server.
Server[4].Open	176	Indicates that the module has opened the server port and is waiting for a connection to a client.
Server[4].Established	177	This status value contains the total number of times a connection was established on the socket.
Server[4].Closed	178	This status value contains the total number of times a close operation was performed on the socket.
Server[4].RxCount	179	This status value contains the total number of messages received by the server.
Server[4].RxOverflow	180	This status value contains the total number of messages received that exceed the specified buffer size for the server.
Server[4].TxCount	181	This status value contains the total number of messages transmitted by the server.
Server[4].TxOverflow	182	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the server.
Server[4].Timeout	183	This status value contains the total number of times a connection timeout occurred on the socket. This status value is incremented each time a connection timeout or a socket close timeout (2 seconds) occurs.
Server[4].CfgErrWord	184	This bit mapped word defines the configuration errors for the server.
Client[0].Connected	185	This flag defines if the client is utilized and connected to a server. A value of 0 indicates the client is not connected and can be utilized for a connection. Any other value indicates the client is connected and being used.
Client[0].State	186	This flag defines the current state of the client.
Client[0].IP	187	This double-word value contains the IP address of the server connected to the client.
Client[0].Port	189	This word value contains the port address for the server connected to the client.

<b>Object In GSCInStat</b>	<b>Block Offset Start</b>	<b>Description</b>
Client[0].RxCount	190	This status value contains the total number of messages received by the client.
Client[0].RxOverflow	191	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[0].TxCount	192	This status value contains the total number of messages transmitted by the client.
Client[0].TxOverflow	193	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[0].spare	194	Reserved for future use
Client[1].Connected	195	This flag defines if the client is utilized and connected to a server. A value of 0 indicates the client is not connected and can be utilized for a connection. Any other value indicates the client is connected and being used.
Client[1].State	196	This flag defines the current state of the client.
Client[1].IP	197	This double-word value contains the IP address of the server connected to the client.
Client[1].Port	199	This word value contains the port address for the server connected to the client.
Client[1].RxCount	200	This status value contains the total number of messages received by the client.
Client[1].RxOverflow	201	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[1].TxCount	202	This status value contains the total number of messages transmitted by the client.
Client[1].TxOverflow	203	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[1].spare	204	Reserved for future use
Client[2].Connected	205	This flag defines if the client is utilized and connected to a server. A value of 0 indicates the client is not connected and can be utilized for a connection. Any other value indicates the client is connected and being used.
Client[2].State	206	This flag defines the current state of the client.
Client[2].IP	207	This double-word value contains the IP address of the server connected to the client.
Client[2].Port	209	This word value contains the port address for the server connected to the client.
Client[2].RxCount	210	This status value contains the total number of messages received by the client.
Client[2].RxOverflow	211	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[2].TxCount	212	This status value contains the total number of messages transmitted by the client.

<b>Object In GSCInStat</b>	<b>Block Offset Start</b>	<b>Description</b>
Client[2].TxOverflow	213	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[2].spare	214	Reserved for future use
Client[3].Connected	215	This flag defines if the client is utilized and connected to a server. A value of 0 indicates the client is not connected and can be utilized for a connection. Any other value indicates the client is connected and being used.
Client[3].State	216	This flag defines the current state of the client.
Client[3].IP	217	This double-word value contains the IP address of the server connected to the client.
Client[3].Port	219	This word value contains the port address for the server connected to the client.
Client[3].RxCount	220	This status value contains the total number of messages received by the client.
Client[3].RxOverflow	221	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[3].TxCount	222	This status value contains the total number of messages transmitted by the client.
Client[3].TxOverflow	223	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[3].spare	224	Reserved for future use
Client[4].Connected	225	This flag defines if the client is utilized and connected to a server. A value of 0 indicates the client is not connected and can be utilized for a connection. Any other value indicates the client is connected and being used.
Client[4].State	226	This flag defines the current state of the client.
Client[4].IP	227	This double-word value contains the IP address of the server connected to the client.
Client[4].Port	229	This word value contains the port address for the server connected to the client.
Client[4].RxCount	230	This status value contains the total number of messages received by the client.
Client[4].RxOverflow	231	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[4].TxCount	232	This status value contains the total number of messages transmitted by the client.
Client[4].TxOverflow	233	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[4].spare	234	Reserved for future use

The format of the server configuration error word is shown in the following table:

#### 8.4.1 Configuration Error Word Bits

Bit Position	Description
Bit 0	0x01 A value other than Y or N was entered for the server enabled parameter.
Bit 1	0x02 Not utilized
Bit 2	0x04 A value less than 5000 is entered for the Connection Timeout parameter other than 0.
Bit 3	0x08 An invalid value is entered for the Connection Close Type parameter. The program only accepts value of 0 to 2.

## 8.5 Configuration Data

This section contains a listing of the parameters and their definitions for the module configuration definition.

[Section]/Item	Value	Range	Description
[MODULE]			Module parameter definitions
Module Name:		0 to 80 characters	This parameter assigns a name to the module that can be viewed using the configuration/debug port. Use this parameter to identify the module and the configuration file.

[Section]/Item	Value	Range	Description
[Server 0]			Server definitions
Enabled:		Y or N	This parameter determines if the server will be utilized by the module. If a value of 'Y' is entered, the server will be used. Any other value will disable the server.
Service Port Number:		1 to 65535	This parameter sets the TCP/IP service port for this server. Each server can have its own unique service port or can share the same number with other servers.
Connection Timeout:		0 to 65535	This parameter specifies the number of milliseconds the server will permit the server to be inactive after a connection is made before closing the socket. This timeout period is reset on each read or write packet. If the parameter is set to 0, the connection will not timeout.
Connection Close Type:		0, 1 or 2	This coded parameter defines the personality of the server after a connection is made. If the parameter is set to 0, the socket will only be closed when a request from the client is received or the connection timeout is exceeded. If a value of 1 is selected, the server will close the socket after it transmits a single message. If a value of 2 is selected, the server will close the socket after it receives a message.
Swap Rx Data Bytes:		Y or N	This parameter is determines if the data received by the server will have the byte order of the data swapped. If the parameter is set to N, no byte swapping will occur. If the parameter is set to Y, the odd byte will be swapped with the even byte in each word of data received.

[Section]/Item	Value	Range	Description
Swap Tx Data Bytes:		Y or N	This parameter is determines if the data to be transmitted by the server will have the byte order of the data swapped. If the parameter is set to N, no byte swapping will occur. If the parameter is set to Y, the odd byte will be swapped with the even byte in each word of data received.

**Note:** Add other [Server n] sections for each server to be utilized.

[Section]/Item	Value	Range	Description
[Server n]			Server definitions
Enabled:		Y or N	This parameter determines if the server will be utilized by the module. If a value of 'Y' is entered, the server will be used. Any other value will disable the server.
Service Port Number:		1 to 65535	This parameter sets the TCP/IP service port for this server. Each server can have its own unique service port or can share the same number with other servers.
Connection Timeout:		0 or 5000 to 65535	This parameter specifies the number of milliseconds the server will permit the server to be inactive after a connection is made before closing the socket. This timeout period is reset on each read or write packet. If the parameter is set to 0, the connection will not timeout.
Connection Close Type:		0, 1 or 2	This coded parameter defines the personality of the server after a connection is made. If the parameter is set to 0, the socket will only be closed when a request from the client is received or the connection timeout is exceeded. If a value of 1 is selected, the server will close the socket after it transmits a single message. If a value of 2 is selected, the server will close the socket after it receives a message.
Swap Rx Data Bytes:		Y or N	This parameter is determines if the data received by the server will have the byte order of the data swapped. If the parameter is set to N, no byte swapping will occur. If the parameter is set to Y, the odd byte will be swapped with the even byte in each word of data received.
Swap Tx Data Bytes:		Y or N	This parameter is determines if the data to be transmitted by the server will have the byte order of the data swapped. If the parameter is set to N, no byte swapping will occur. If the parameter is set to Y, the odd byte will be swapped with the even byte in each word of data received.

## 9 Support, Service & Warranty

### *In This Chapter*

- ❖ How to Contact Us: Sales and Support ..... 92
- ❖ Return Material Authorization (RMA) Policies and Conditions..... 92
- ❖ Procedures for Return of Units Under Warranty ..... 93
- ❖ Procedures for Return of Units Out of Warranty ..... 94
- ❖ LIMITED WARRANTY ..... 95

*Be sure and read the full Warranty that can be found on our web site at [www.prosoft-technology.com](http://www.prosoft-technology.com) for details and other terms and conditions. The content in this summary is subject to change without notice. The content is current at date of publication.*

ProSoft Technology, Inc. strives to provide meaningful support to its customers. Should any questions or problems arise, please feel free to contact us at:

---

<b>Internet</b>	Web Site: <a href="http://www.prosoft-technology.com/support">http://www.prosoft-technology.com/support</a>
	E-mail address: <a href="mailto:support@prosoft-technology.com">support@prosoft-technology.com</a>

---

Those of us at ProSoft Technology, Inc. want to provide the best and quickest support possible, so before calling please have the following information available. You may wish to fax this information to us prior to calling.

- 1 Product Version Number
- 2 System architecture
- 3 Network details

In the case of hardware, we will also need the following information:

- 1 Module configuration and contents of file
- 2 Module Operation
- 3 Configuration/Debug status information
- 4 LED patterns
- 5 Information about the processor and user data files as viewed through the development software and LED patterns on the processor
- 6 Details about the networked devices interfaced, if any

For technical support calls within the United States, an after-hours answering system allows pager access to one of our qualified technical and/or application support engineers at any time to answer your questions.

## 9.1 How to Contact Us: Sales and Support

All ProSoft Technology Products are backed with full technical support. Contact our worldwide Technical Support team and Customer Service representatives directly by phone or email:

### **USA / Latin America (excluding Brasil) (Office in California)**

+1(661) 716-5100  
+1(661) 716-5101 (Fax)  
1675 Chester Avenue, 4th Floor  
Bakersfield, California 93301  
U.S.A.  
+1.661.716.5100, support@prosoft-technology.com  
Languages spoken include: English, Spanish

### **Asia Pacific Sales (office in Malaysia)**

+603.7724.2080  
+603.7724.2090 (Fax)  
C210, Damansara Intan,  
1 Jalan SS20/27, 47400 Petaling Jaya  
Selangor, Malaysia  
+603.7724.2080, asiapc@prosoft-technology.com  
Languages spoken include: Chinese, Japanese, English

### **Asia Pacific Support (office in China)**

+86.21.64518356 x 8011  
+86.21.64756957 (Fax)  
4/F, No. 16 Hongcao Road  
Shanghai, China 200233  
China  
+86.21.64518356 x 8011, zhang@prosoft-technology.com  
Languages spoken include: Chinese, English

### **Europe / Middle East / Africa (office in Toulouse, France)**

+33 (0) 5.34.36.87.20  
+33 (0) 5.61.78.40.52 (Fax)  
Zone d'activité de Font Grasse  
17, rue des Briquetiers  
F-31700 Blagnac  
France  
+33 (0) 5.34.36.87.20. support.emea@prosoft-technology.com  
Languages spoken include: French, English

### **Brasil (office in Sao Paulo)**

+55-11-5084-5178  
+55-11-5083-3776 (Fax)  
Rua Vergueiro, 2949 - sala 182 - Edifício Vergueiro Work Center  
Vila Mariana - São Paulo  
Cep: 04101-300 - Brasil  
+55-11-5084-5178, eduardo@prosoft-technology.com  
Languages spoken include: Portuguese, English

## 9.2 Return Material Authorization (RMA) Policies and Conditions

The following RMA Policies and Conditions apply to any returned product. These RMA Policies are subject to change by ProSoft without notice. For warranty information, see Section C below entitled "Limited Warranty". In the event of any inconsistency between the RMA Policies and the Warranty, the Warranty shall govern.

### 9.2.1 All Product Returns

- 1 In order to return a Product for repair, exchange or otherwise, the Customer must obtain a Returned Material Authorization (RMA) number from ProSoft and comply with ProSoft shipping instructions.
- 2 In the event that the Customer experiences a problem with the Product for any reason, Customer should contact ProSoft Technical Support at one of the telephone numbers listed above in Section A. A Technical Support Engineer will request several tests in an attempt to isolate the problem. If after these tests are completed, the Product is found to be the source of the problem, ProSoft will issue an RMA.
- 3 All returned Products must be shipped freight prepaid, in the original shipping container or equivalent, to the location specified by ProSoft, and be accompanied by proof of purchase. The RMA number is to be prominently marked on the outside of the shipping box. Customer agrees to insure the Product or assume the risk of loss or damage in transit. Products shipped to ProSoft without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
- 4 Out of warranty returns are not allowed on RadioLinX accessories such as antennas, cables, and brackets.

The following policy applies for Non-Warranty Credit Returns:

- A 10% Restocking Fee if Factory Seal is *not* broken
- B 20% Restocking Fee if Factory Seal is broken

ProSoft retains the right, in its absolute and sole discretion, to reject any non-warranty returns for credit if the return is not requested within three (3) months after shipment of the Product to Customer, if the Customer fails to comply with ProSoft's shipping instructions, or if the Customer fails to return the Product to ProSoft within six (6) months after Product was originally shipped.

### 9.3 Procedures for Return of Units Under Warranty

- 1 A Technical Support Engineer must pre-approve all product returns.
- 2 Module is repaired or replaced after a Return Material Authorization Number is entered and a replacement order is generated.
- 3 Credit for the warranted item is issued within 10 business days after receipt of product and evaluation of the defect has been performed by ProSoft. The credit will only be issued provided the product is returned with a valid Return Material Authorization Number and in accordance with ProSoft's shipping instructions.

- a) If no defect is found, a credit is issued.
- b) If a defect is found and is determined to be customer generated or if the defect is otherwise not covered by ProSoft's Warranty, or if the module is not repairable, a credit is not issued and payment of the replacement module is due.

## 9.4 Procedures for Return of Units Out of Warranty

- 1 Customer sends unit in for evaluation.
- 2 If no defect is found, Customer will be charged the equivalent of US \$100 plus shipping, duties and taxes that may apply. A new Purchase Order will be required for this evaluation fee.

If the unit is repaired the charge to the Customer will be 30%\* of the list price plus any shipping, duties and taxes that may apply. A new Purchase Order will be required for a product repair.

- 3 For an immediate exchange, a new module may be purchased and sent to Customer while repair work is being performed. Credit for purchase of the new module will be issued when the new module is returned in accordance with ProSoft's shipping instructions and subject to ProSoft's policy on non-warranty returns. This is in addition to charges for repair of the old module and any associated charges to Customer.
- 4 If, upon contacting ProSoft Customer Service, the Customer is informed that unit is believed to be unrepairable, the Customer may choose to send unit in for evaluation to determine if the repair can be made. Customer will pay shipping, duties and taxes that may apply. If unit cannot be repaired, the Customer may purchase a new unit.

### 9.4.1 Un-repairable Units

- 3150-All
- 3750
- 3600-All
- 3700
- 3170-All
- 3250
- 1560 can be repaired, if defect is the power supply
- 1550 can be repaired, if defect is the power supply
- 3350
- 3300
- 1500-All

**\* 30% of list price is an estimated repair cost only. The actual cost of repairs will be determined when the module is received by ProSoft and evaluated for needed repairs.**

### Purchasing Warranty Extension

As detailed below in ProSoft's Warranty, the standard Warranty Period is one year (or in the case of RadioLinx modules, three years) from the date of delivery. The Warranty Period may be extended for an additional charge, as follows:

- Additional 1 year = 10% of list price
- Additional 2 years = 20% of list price
- Additional 3 years = 30% of list price

## **9.5 LIMITED WARRANTY**

This Limited Warranty ("Warranty") governs all sales of hardware, software and other products (collectively, "Product") manufactured and/or offered for sale by ProSoft, and all related services provided by ProSoft, including maintenance, repair, warranty exchange, and service programs (collectively, "Services"). By purchasing or using the Product or Services, the individual or entity purchasing or using the Product or Services ("Customer") agrees to all of the terms and provisions (collectively, the "Terms") of this Limited Warranty. All sales of software or other intellectual property are, in addition, subject to any license agreement accompanying such software or other intellectual property.

### **9.5.1 What Is Covered By This Warranty**

**A** *Warranty On New Products:* ProSoft warrants, to the original purchaser only, that the Product that is the subject of the sale will (1) conform to and perform in accordance with published specifications prepared, approved, and issued by ProSoft, and (2) will be free from defects in material or workmanship; provided these warranties only cover Product that is sold as new. This Warranty expires one year (or in the case of RadioLinx modules, three years) from the date of shipment (the "Warranty Period"). If the Customer discovers within the Warranty Period a failure of the Product to conform to specifications, or a defect in material or workmanship of the Product, the Customer must promptly notify ProSoft by fax, email or telephone. In no event may that notification be received by ProSoft later than 15 months (or in the case of RadioLinx modules, 39 months) from the date of delivery. Within a reasonable time after notification, ProSoft will correct any failure of the Product to conform to specifications or any defect in material or workmanship of the Product, with either new or used replacement parts. Such repair, including both parts and labor, will be performed at ProSoft's expense. All warranty service will be performed at service centers designated by ProSoft. If ProSoft is unable to repair the Product to conform to this Warranty after a reasonable number of attempts, ProSoft will provide, at its option, one of the following: a replacement product, a full refund of the purchase price or a credit in the amount of the purchase price. All replaced product and parts become the property of ProSoft. These remedies are the Customer's only remedies for breach of warranty.

- B** *Warranty On Services:* Material and labor used by ProSoft to repair a verified malfunction or defect are warranted on the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.
- C** The Warranty Period for RadioLinx accessories (such as antennas, cables, brackets, etc.) are the same as for RadioLinx modules, that is, three years from the date of shipment.

### **9.5.2 What Is Not Covered By This Warranty**

- A** ProSoft makes no representation or warranty, expressed or implied, that the operation of software purchased from ProSoft will be uninterrupted or error free or that the functions contained in the software will meet or satisfy the purchaser's intended use or requirements; the Customer assumes complete responsibility for decisions made or actions taken based on information obtained using ProSoft software.
- B** With the exception of RadioLinx accessories referenced in paragraph 1(c) this Warranty does not cover any product, components, or parts not manufactured by ProSoft.
- C** This Warranty also does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3 programming languages, or "C") not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges outside of the limits indicated on the product specifications; or (viii) disasters such as fire, flood, earthquake, wind or lightning.
- D** The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guides included with your original product purchased by you from ProSoft, contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

**9.5.3 DISCLAIMER REGARDING HIGH RISK ACTIVITIES**

PRODUCT MANUFACTURED OR SUPPLIED BY PROSOFT IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN HAZARDOUS ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE (INCLUDING, WITHOUT LIMITATION, THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION OF COMMUNICATION SYSTEMS, AIR TRAFFIC CONTROL, DIRECT LIFE SUPPORT MACHINES OR WEAPONS SYSTEMS), IN WHICH THE FAILURE OF THE PRODUCT COULD LEAD DIRECTLY OR INDIRECTLY TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE (COLLECTIVELY, "HIGH RISK ACTIVITIES"). PROSOFT SPECIFICALLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR HIGH RISK ACTIVITIES.

**9.5.4 DISCLAIMER OF ALL OTHER WARRANTIES**

THE WARRANTIES SET FORTH IN PARAGRAPH 1 ABOVE ARE IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

**9.5.5 LIMITATION OF REMEDIES\*\***

IN NO EVENT WILL PROSOFT (OR ITS DEALER) BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES BASED ON BREACH OF WARRANTY, BREACH OF CONTRACT, NEGLIGENCE, STRICT TORT, OR ANY OTHER LEGAL THEORY. DAMAGES THAT PROSOFT AND ITS DEALER WILL NOT BE RESPONSIBLE FOR INCLUDE, BUT ARE NOT LIMITED TO: LOSS OF PROFITS; LOSS OF SAVINGS OR REVENUE; LOSS OF USE OF THE PRODUCT OR ANY ASSOCIATED EQUIPMENT; LOSS OF DATA; COST OF CAPITAL; COST OF ANY SUBSTITUTE EQUIPMENT, FACILITIES, OR SERVICES; DOWNTIME; THE CLAIMS OF THIRD PARTIES, INCLUDING CUSTOMERS OF THE PURCHASER; AND INJURY TO PROPERTY.

\*\* Some areas do not allow time limitations on an implied warranty, or allow the exclusion or limitation of incidental or consequential damages. In such areas the above limitations may not apply. This Warranty gives you specific legal rights, and you may also have other rights which vary from place to place.

**9.5.6 Time Limit for Bringing Suit**

Any action for breach of warranty must be commenced within 15 months (or in the case of RadioLinx modules, 39 months) following shipment of the Product.

### **9.5.7 No Other Warranties**

Unless modified in writing and signed by both parties, this Warranty is understood to be the complete and exclusive agreement between the parties, suspending all oral or written prior agreements and all other communications between the parties relating to the subject matter of this Warranty, including statements made by salesperson. No employee of ProSoft or any other party is authorized to make any warranty in addition to those made in this Warranty. The Customer is warned, therefore, to check this Warranty carefully to see that it correctly reflects those terms that are important to the Customer.

### **9.5.8 Intellectual Property**

- A** Any documentation included with Product purchased from ProSoft is protected by copyright and may not be photocopied or reproduced in any form without prior written consent from ProSoft.
- B** ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.
- C** Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.
- D** Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.
- E** Customer agrees that it shall not, and shall not authorize others to, copy software provided by ProSoft (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the Product; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of Product.

### **9.5.9 Additional Restrictions Relating To Software And Other Intellectual Property**

In addition to complying with the Terms of this Warranty, Customers purchasing software or other intellectual property shall comply with any license agreement accompanying such software or other intellectual property. Failure to do so may void this Warranty with respect to such software and/or other intellectual property.

### **9.5.10 Allocation of risks**

This Warranty allocates the risk of product failure between ProSoft and the Customer. This allocation is recognized by both parties and is reflected in the price of the goods. The Customer acknowledges that it has read this Warranty, understands it, and is bound by its Terms.

**9.5.11 Controlling Law and Severability**

This Warranty shall be governed by and construed in accordance with the laws of the United States and the domestic laws of the State of California, without reference to its conflicts of law provisions. If for any reason a court of competent jurisdiction finds any provisions of this Warranty, or a portion thereof, to be unenforceable, that provision shall be enforced to the maximum extent permissible and the remainder of this Warranty shall remain in full force and effect. Any cause of action with respect to the Product or Services must be instituted in a court of competent jurisdiction in the State of California.



# Index

- [  
[BACKPLANE] • 54  
[Module] • 54  
[Server n] • 55
- 3**  
3X Read Blocks • 74  
3x Register Start • 54
- 4**  
4x Register Start • 54  
4X Write Block • 75
- A**  
Add the PTQ Module to the Project • 18, 37  
Adding a Module • 51  
Adding a Project • 51  
Additional Restrictions Relating To Software And Other  
Intellectual Property • 98  
All Product Returns • 93  
Allocation of risks • 98
- B**  
Backplane Data Exchange • 73  
Backplane Data Transfer • 73  
Build the Project • 39
- C**  
Cable Connections • 82  
Client Connection States • 81  
Client Control Word Codes • 78  
Client state list • 81  
Comment Entries • 53  
Configuration Data • 89  
Configuration Error Word Bits • 89  
Configuring the Processor with Concept • 13  
Configuring the Processor with ProWORX • 31  
Configuring the Processor with UnityPro XL • 35  
Connect the PC to the ProTalk Configuration/Debug  
Port • 45  
Connect Your PC to the Processor • 40  
Connecting to the Processor with TCP/IP • 41  
Connection Close Type • 55  
Connection Timeout • 55  
Controlling Law and Severability • 99  
Create a New Project • 15, 35
- D**  
Diagnostics and Troubleshooting • 7, 59  
DISCLAIMER OF ALL OTHER WARRANTIES • 97
- DISCLAIMER REGARDING HIGH RISK ACTIVITIES •  
97  
Download the Project to the Module • 57  
Download the Project to the Processor • 23, 42
- E**  
Enabled • 55  
Ethernet Configuration • 56  
Ethernet Connection • 82  
Ethernet LED Indicators • 67  
Ethernet Port Configuration - wattcp.cfg • 82  
Exiting the Program • 65
- F**  
Features and Benefits • 69  
Functional Overview • 7, 71  
Functional Specifications • 70
- G**  
General Concepts • 71  
General Specifications • 69  
Guide to the PTQ-GEC Application Reference Guide •  
7
- H**  
Handling Multiple Blocks • 78  
Hardware and Software Requirements • 9  
Hardware Specifications • 70  
How to Contact Us  
Sales and Support • 92
- I**  
Information for Concept Version 2.6 Users • 14  
Input and Output Blocks • 74  
Inserting the 1454-9F connector • 44  
Install ProSoft Configuration Builder Software • 11  
Install the ProTalk Module in the Quantum Rack • 43,  
44  
Installing MDC Configuration Files • 14  
Intellectual Property • 98
- K**  
Keystrokes • 62
- L**  
LED Status Indicators • 66  
LIMITATION OF REMEDIES\*\* • 97  
LIMITED WARRANTY • 95
- M**  
Main Logic Loop • 72  
Main Menu • 62  
Modifying the Configuration File • 49  
Module Control Word Codes • 78  
Module Entries • 53  
Module Name Parameter • 54  
Module Power Up • 72  
Multiple Blocks • 75, 77

## N

Navigation • 61  
Network Data Transfer • 79  
Network Menu • 65  
No Other Warranties • 98

## O

Opening the Network Menu • 64

## P

PC and PC Software • 10  
Please Read This Notice • 2  
Printing a Configuration File • 53  
Procedures for Return of Units Out of Warranty • 94  
Procedures for Return of Units Under Warranty • 93  
Product Specifications • 7, 69  
ProSoft Configuration Builder • 49  
ProTalk Module Carton Contents • 9  
PTQ Installation and Operating Instructions • 2  
PTQ-GEC Status Data • 83  
Purchasing Warranty Extension • 95

## Q

Quantum / Unity Hardware • 10

## R

Reading Status Data from the Module • 59  
Redisplaying the Menu • 62  
Reference • 7, 69  
Required Hardware • 59  
Resetting diagnostic data • 63  
Return Material Authorization (RMA) Policies and Conditions • 92  
Returning to the Main Menu • 66  
RS-232 Configuration/Debug Port • 83

## S

Server Control Word Codes • 78  
Server state list • 80  
Service Port • 55  
Set Module Parameters • 52  
Set up Data Memory in Project • 20, 47  
Set Up the Project • 50  
Setting Up the ProTalk Module • 43  
Start Here • 7, 9  
Support, Service & Warranty • 7, 91  
Swap RX Data Bytes • 56  
Swap TX Data Bytes • 56

## T

The Configuration/Debug Menu • 60  
Time Limit for Bringing Suit • 97  
Transferring the Configuration File from PC to PTQ module • 63  
Transferring the Configuration File from PTQ module to PC • 63  
Transferring WATTCP.CFG to the module • 65  
Transferring WATTCP.CFG to the PC • 65

## U

Un-repairable Units • 94  
Using the Diagnostic Window in ProSoft Configuration Builder • 60

## V

Verify Communication Between the Processor and the Module • 7, 47  
Verify Jumper Settings • 43  
Verify Successful Download • 26  
Viewing Block Transfer Statistics • 63  
Viewing Client Communication Status (Clients 10 to 14) • 64  
Viewing Module Configuration • 63  
Viewing Server Communication Status (Servers 0 to 4) • 64  
Viewing Server Configuration (Servers 0 to 4) • 64  
Viewing the WATTCP.CFG file on the module • 66  
Viewing Version Information • 63

## W

Warm Booting the Module • 64  
What Is Covered By This Warranty • 95  
What Is Not Covered By This Warranty • 96

## Y

Your Feedback Please • 3