ProSoft TECHNOLOGY

Where Automation Connects.

ProTalk®

PTQ-103M

Quantum Platform

IEC 60870-5-103 Master

June 30, 2009

USER MANUAL

Information for ProTalk® Product Users

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.

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

AVERTISSEMENT - RISQUE D'EXPLOSION - AVANT DE DÉCONNECTER L'EQUIPMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

CL I Div 2 GPs A, B, C, D

Temp Code T5

II 3 G

Ex nA IIC T5 X

0° C <= Ta <= 60° C

II - Equipment intended for above ground use (not for use in mines).

3 - Category 3 equipment, investigated for normal operation only.

G - Equipment protected against explosive gasses.

Warnings

North America Warnings

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 or rewiring modules.
Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be nonhazardous.
C Suitable for use in Class I, division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

ATEX Warnings and Conditions of Safe Usage:

Power, Input, and Output (I/O) wiring must be in accordance with the authority having jurisdiction

A Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.
B Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
C These products are intended to be mounted in an IP54 enclosure. The devices shall provide external means to prevent the rated voltage being exceeded by transient disturbances of more than 40%. This device must be used only with ATEX certified backplanes.
D DO NOT OPEN WHEN ENERGIZED.

Electrical Ratings

- Backplane Current Load: 800 mA @ 5 V DC; 3mA @ 24V DC
- Operating Temperature: 0 to 60°C (32 to 140°F)
- Storage Temperature: -40 to 85°C (-40 to 185°F)
- Shock: 30g Operational; 50g non-operational; Vibration: 5 g from 10 to 150 Hz
- Relative Humidity 5% to 95% (non-condensing)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Markings:

ANSI / ISA	ISA 12.12.01 Class I Division 2, GPs A, B, C, D
CSA/cUL	C22.2 No. 213-1987
CSA CB Certified	IEC61010
ATEX	EN60079-0 Category 3, Zone 2 EN60079-15

243333

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 µA.

Maximum battery charge voltage = 3.4 VDC.

Maximum battery charge current = 500 µA.

Maximum battery discharge current = 30 µA.

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
5201 Truxtun Ave., 3rd Floor
Bakersfield, CA 93309
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com

PTQ-103M User Manual
June 30, 2009

ProSoft Technology® Product Documentation

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed CD, and are available at no charge from our web site: www.prosoft-technology.com

Printed documentation is available for purchase. Contact ProSoft Technology for pricing and availability.

Asia Pacific: +603.7724.2080

Europe, Middle East, Africa: +33 (0) 5.3436.87.20

Latin America: +1.281.298.9109

North America: +1.661.716.5100

Contents

Guide to the PTQ-103M User Manual

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 75)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview	→	Reference (page 99) Functional Overview (page 101) Product Specifications (page 99)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 171)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

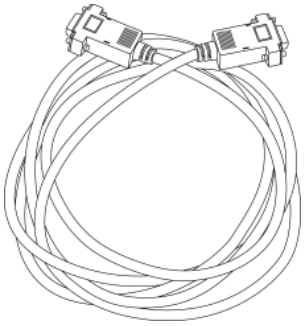
In This Chapter

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 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.
4. Save the file to your Desktop, so that you can find it easily when you have finished downloading.
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 module.

To install ProSoft Configuration Builder from the Product CD

1. Insert the ProSoft Solutions Product CD 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 **SETUPCONFIGURATIONTOOL** folder, double-click the **"PCB_*.EXE"** file and follow the instructions on your screen to install the software on your PC. The information represented by the **"*"** character in the file name is the PCB version number and, therefore, subject to change as new versions of PCB are released.

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

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 consider this when monitoring the status of the PTQ module.

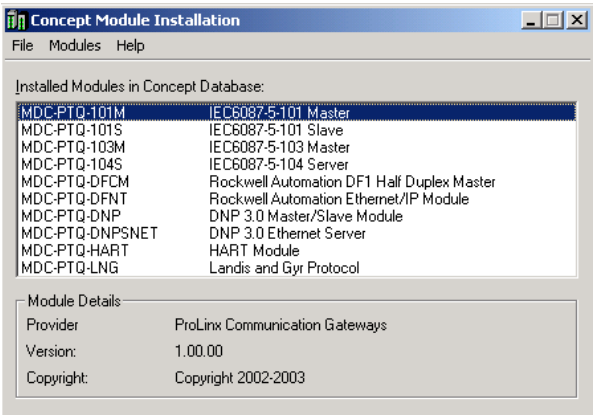
2.1 Information for Concept Version 2.6 Users

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

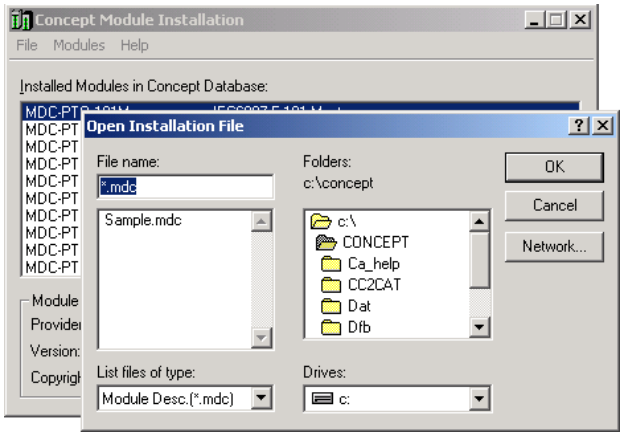
1 From a PC with Concept 2.6 installed, choose **START / PROGRAMS / CONCEPT / MODCONNECT TOOL**.

This action opens the Concept Module Installation dialog box.



2 Choose **FILE / OPEN INSTALLATION FILE.**

This action opens the Open Installation File dialog box:

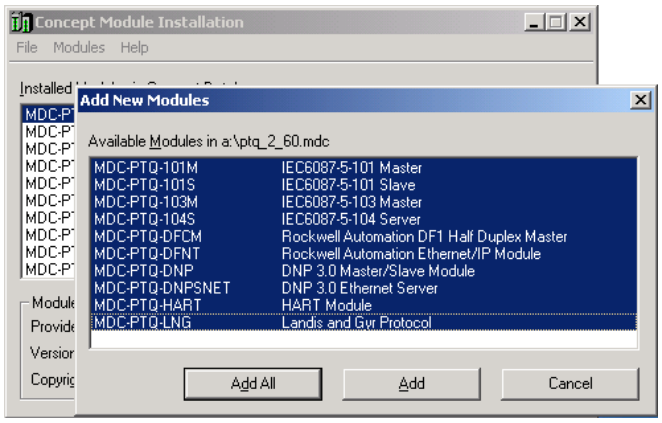


3 If you are using a Quantum processor, you will need the MDC files. In the Open Installation File dialog box, navigate to the **MDC FILES** directory on the ProTalk CD.

4 Choose the MDC file and help file for your version of Concept:

- Concept 2.6 users: select PTQ_2_60.mdc and PTQMDC.hlp
- 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.

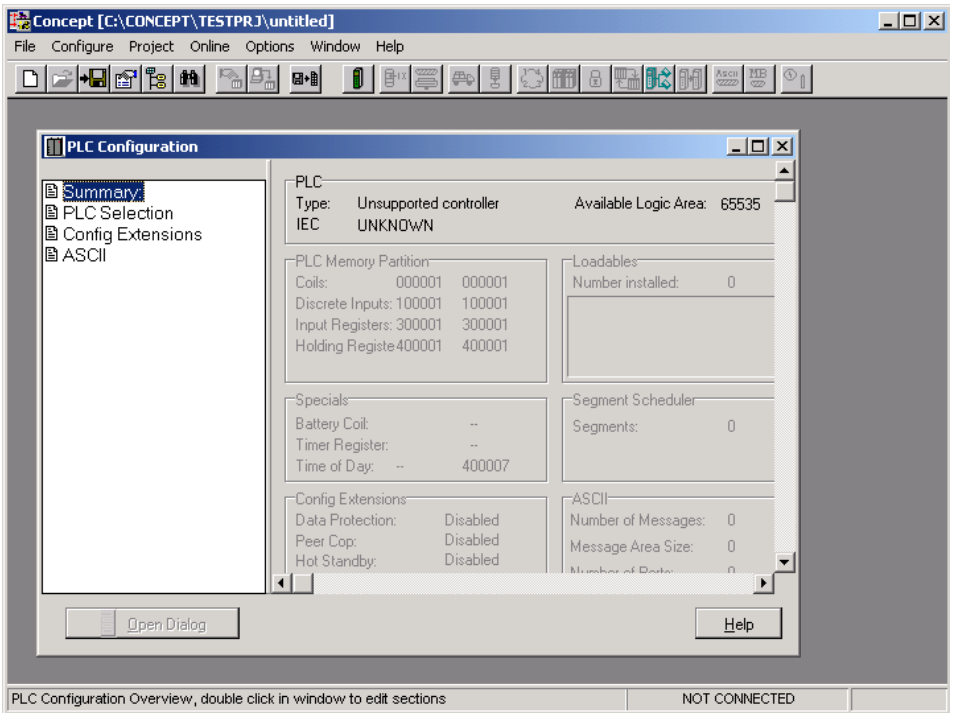
6 When the process is complete, open the File menu and choose Exit to save your changes.

2.2 Create a New Project

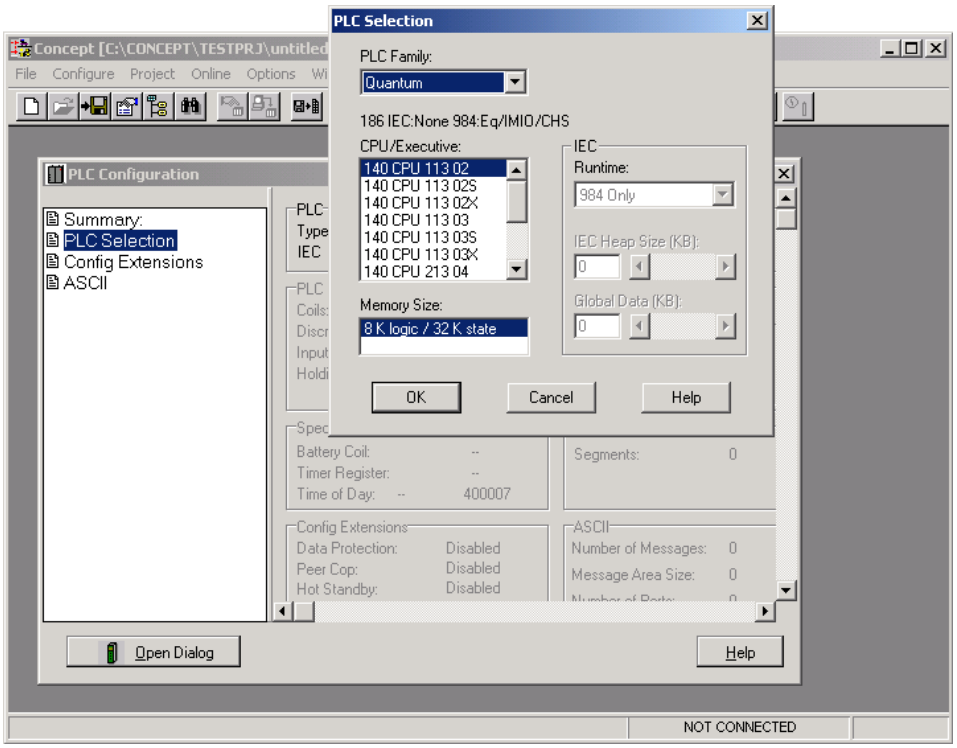
This phase of the setup procedure must be performed on a computer that has the Concept configuration software installed.

1 From your computer, choose **START / PROGRAMS / CONCEPT V2.6 XL.EN / CONCEPT**. This action opens the **CONCEPT** window.

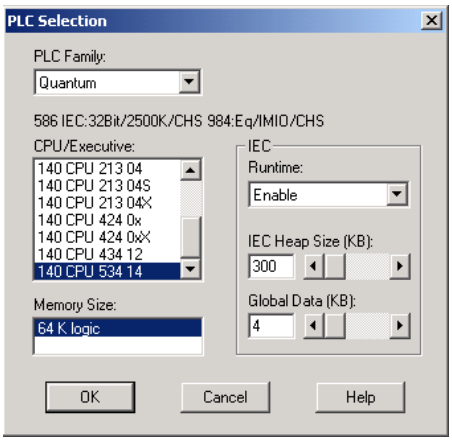
2 Open the File menu, and then choose **NEW PROJECT**. This action opens the **PLC CONFIGURATION** dialog box.



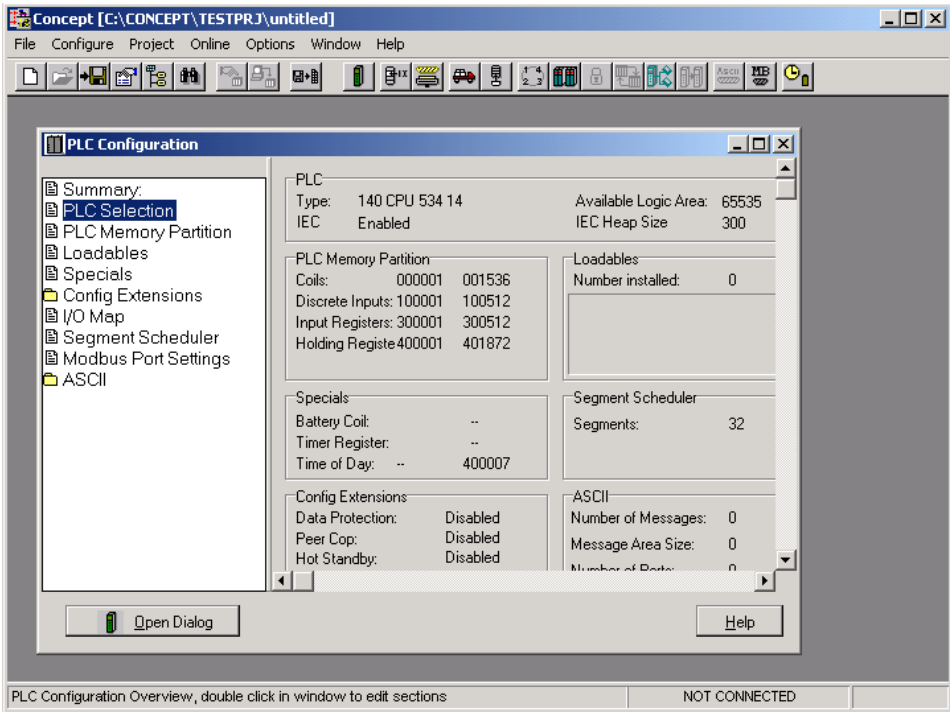
3 In the list of options on the left side of this dialog box, double-click the **PLC SELECTION** folder. This action opens the **PLC SELECTION** dialog box.



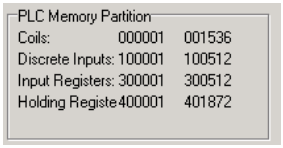
4 In the **CPU/EXECUTIVE** pane, use the scroll bar to locate and select the PLC to configure.



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

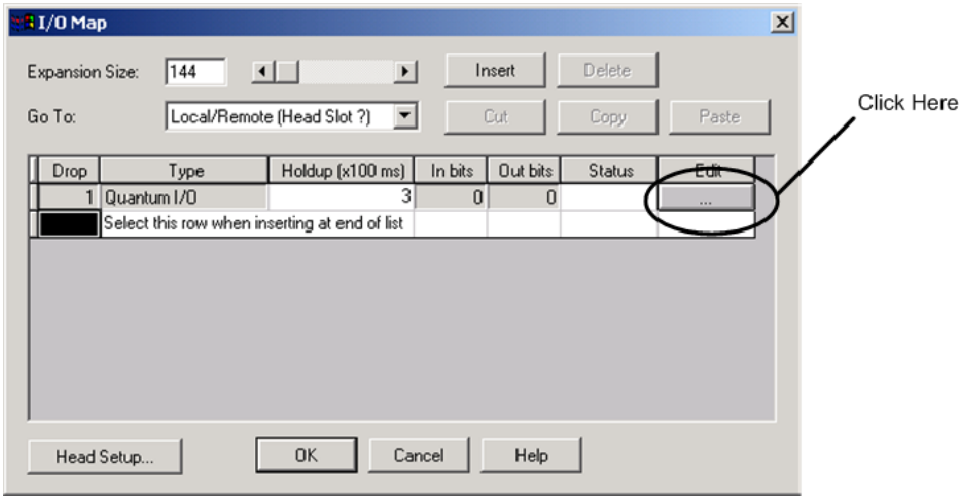


6 Make a note of the holding registers for the module. You will need this information when you modify your application. The Holding Registers are displayed in the PLC Memory Partition pane of the **PLC CONFIGURATION** dialog box.

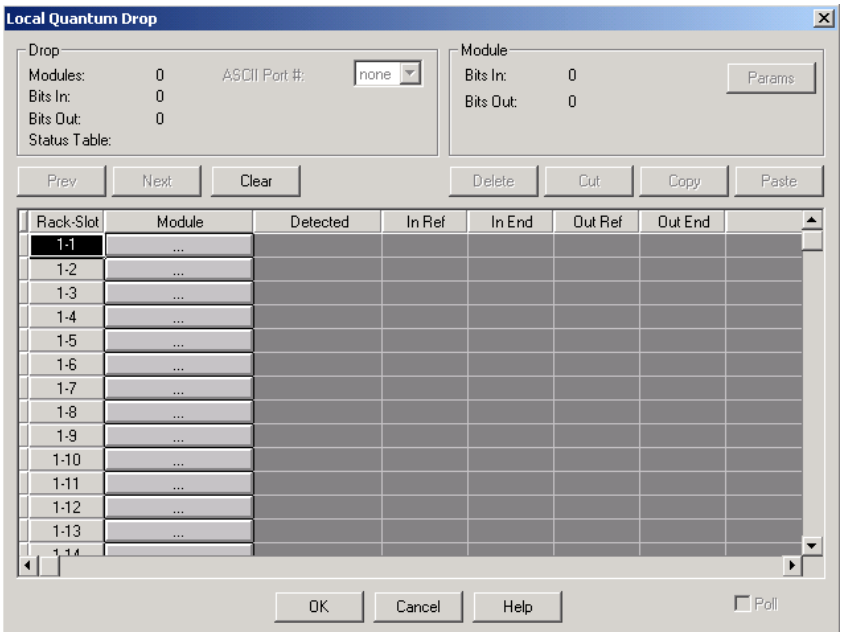


2.3 Add the PTQ Module to the Project

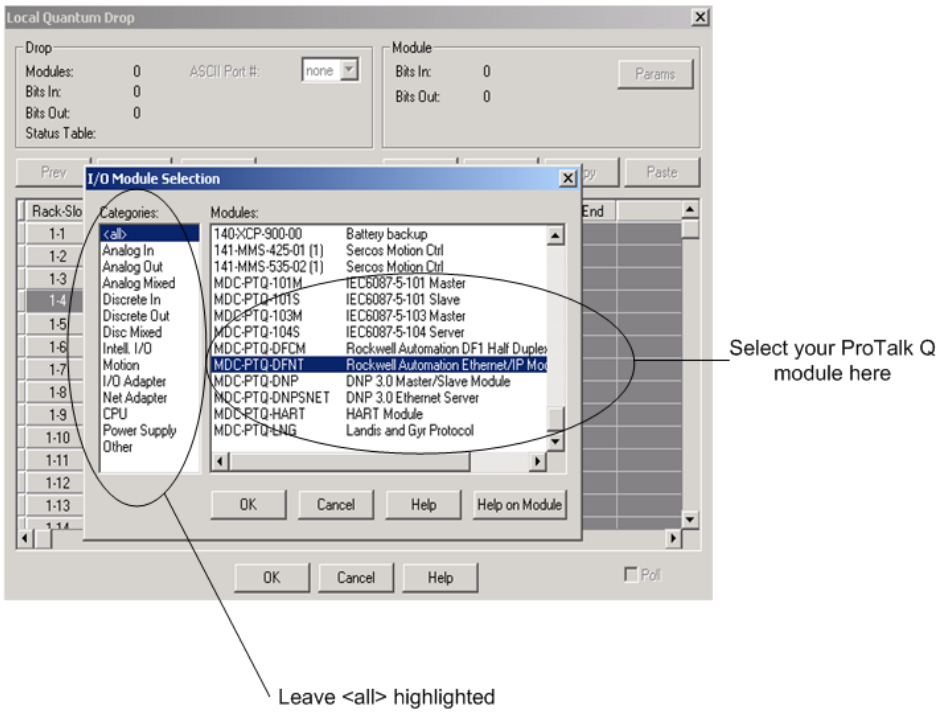
1 In the list of options on the left side of the **PLC CONFIGURATION** dialog box, double-click **I/O MAP**. This action opens the **I/O MAP** dialog box.



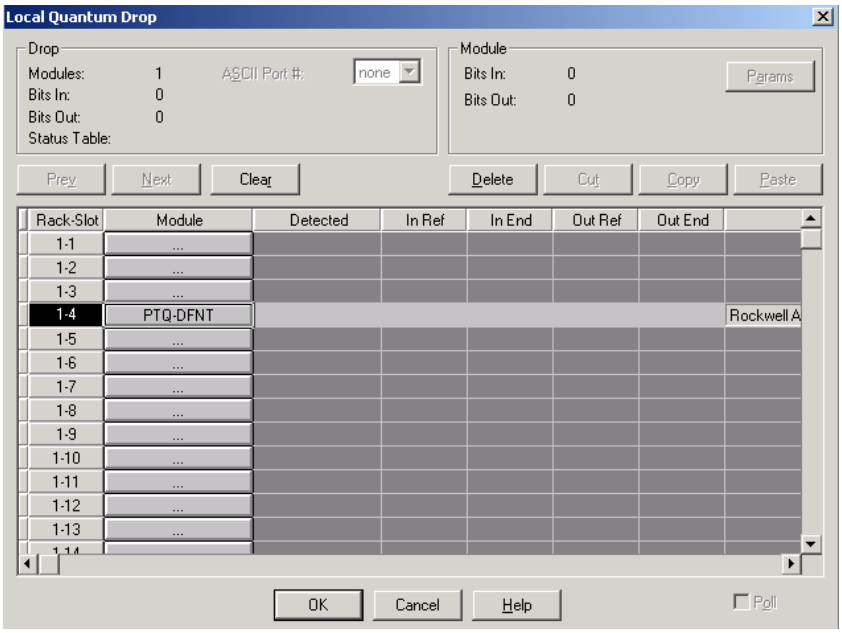
2 Click the **EDIT** button to open the **LOCAL QUANTUM DROP** dialog box. This dialog box is where you identify rack and slot locations.



3 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.

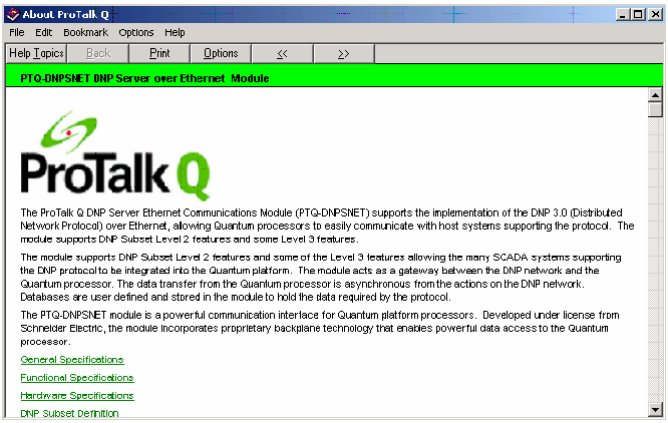


4 In the **MODULES** pane, use the scroll bar to locate and select the ProTalk module, and then click **OK.** This action copies the description of the ProTalk module next to the assigned rack and slot number of the **LOCAL QUANTUM DROP** dialog box.



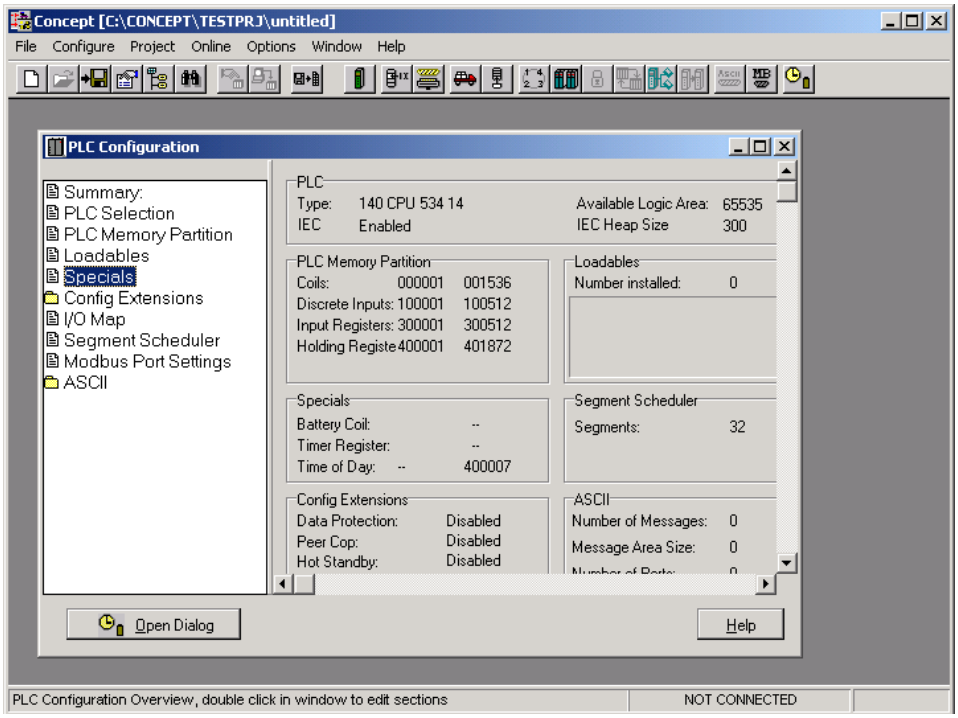
5 Repeat steps 3 through 5 for each ProTalk module you plan to install. When you have finished installing your ProTalk modules, click **OK** to save your settings. Click **YES** to confirm your settings.

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

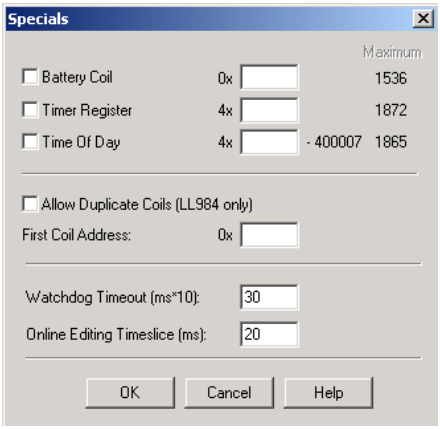


2.4 Set up Data Memory in Project

1 In the list of options on the left side of the **PLC CONFIGURATION** dialog box, double-click **SPECIALS.**

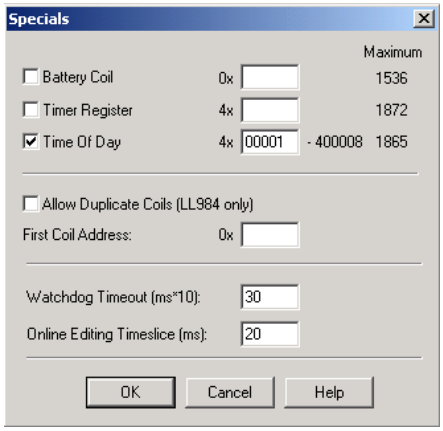


2 This action opens the **SPECIALS** dialog box.



Selecting the Time of Day

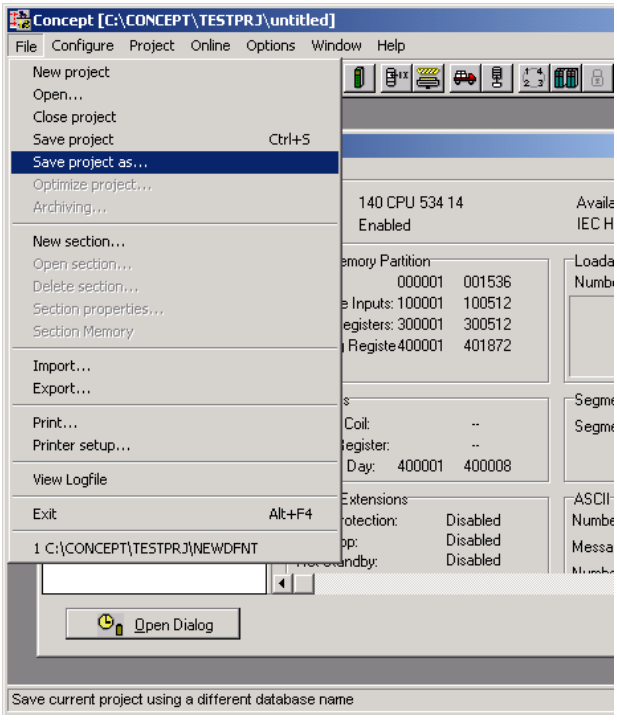
1 Select (check) the **TIME OF DAY** box, and then enter the value 00001 as shown in the following illustration. This value sets the first time of day register to 400001.



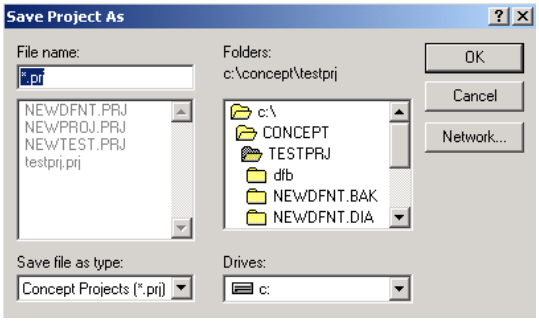
2 Click **OK** to save your settings and close the **SPECIALS** dialog box.

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.

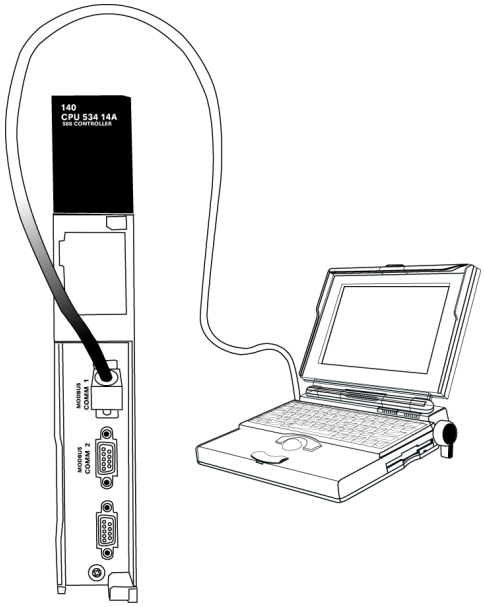


3 Name the project, and then click **OK** to save the project to a file.

2.5 Download the Project to the Processor

Next, download (copy) the project file to the Quantum Processor.

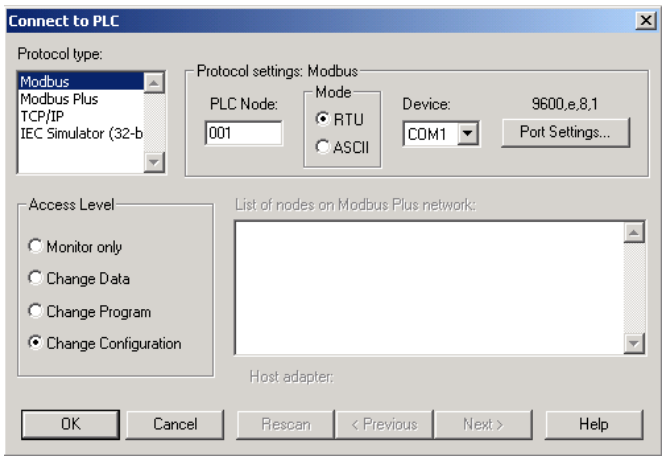
1 Use the null modem cable to connect your PC's serial port to the Quantum processor, as shown in the following illustration.



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.**

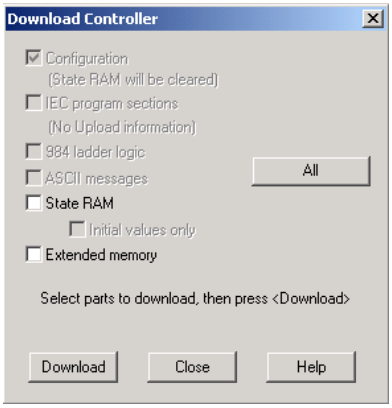
3 In the **PLC CONFIGURATION** dialog box, open the **ONLINE** menu, and then choose **CONNECT.** This action opens the **CONNECT TO PLC** dialog box.



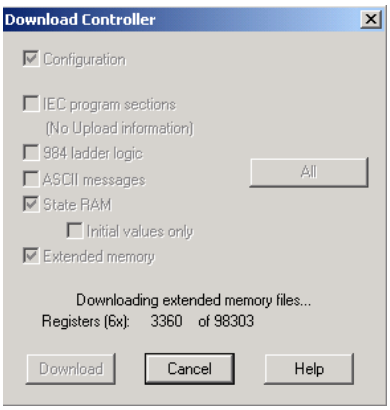
4 Leave the default settings as shown and click **OK.**

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.



6 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.

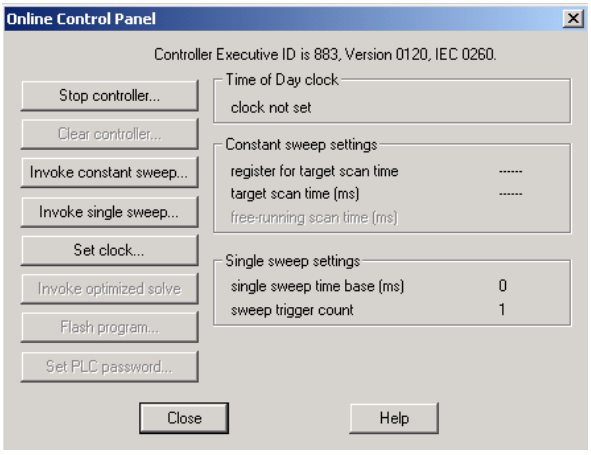


7 When the download is complete, you will be prompted to restart the controller. Click **YES** to restart the controller.

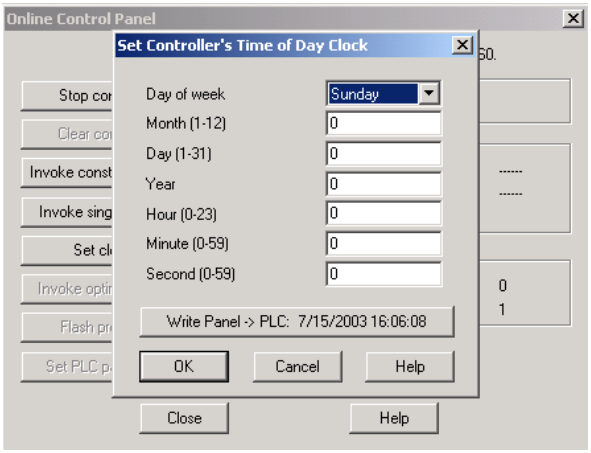
2.6 Verify Successful Download

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.

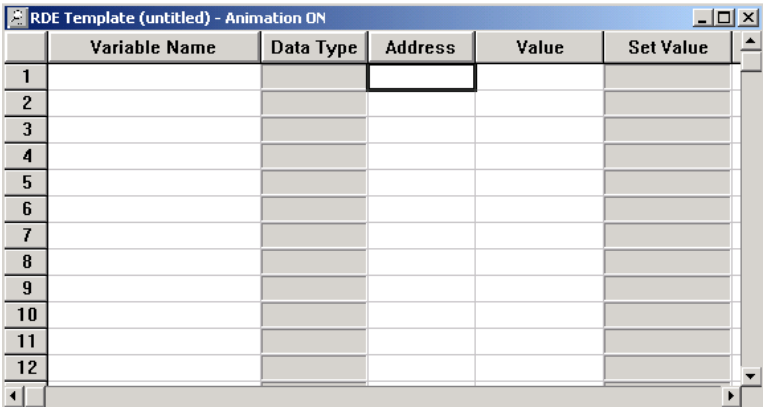


3 Click the **WRITE PANEL** button. This action updates the date and time fields in this dialog box. Click **OK** to close this dialog box and return to the previous window.

4 Click **CLOSE** to close the **ONLINE CONTROL PANEL** dialog box.

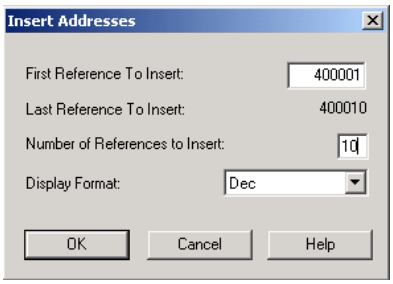
5 In the **PLC CONFIGURATION** window, open the **ONLINE** menu, and then choose **REFERENCE DATA EDITOR.** This action opens the **REFERENCE DATA EDITOR** dialog box. On this dialog box, you will add preset values to data registers that will later be monitored in the ProTalk module.

6 Place the cursor over the first address field, as shown in the following illustration.

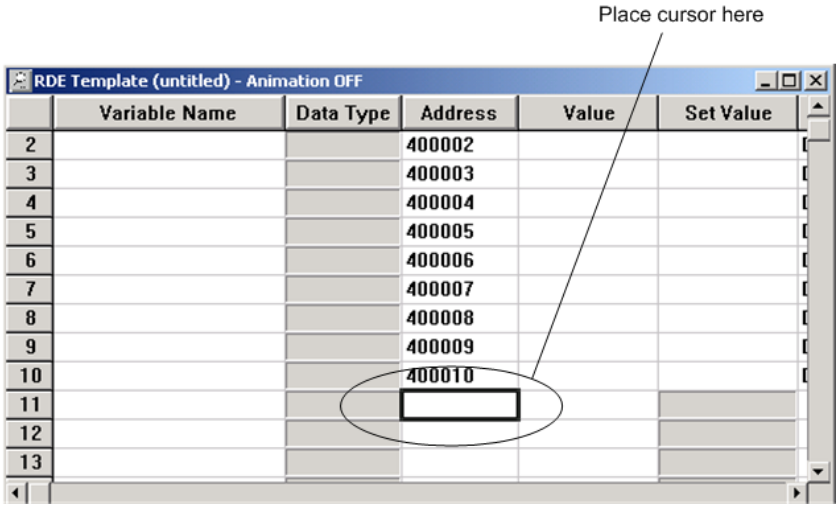


7 In the **PLC CONFIGURATION** window, open the **TEMPLATES** menu, and then choose **INSERT ADDRESSES.** This action opens the Insert addresses dialog box.

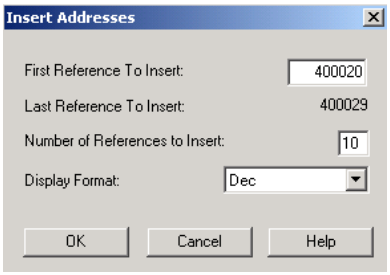
8 On the **INSERT ADDRESSES** dialog box, enter the values shown in the following illustration, and then click **OK.**



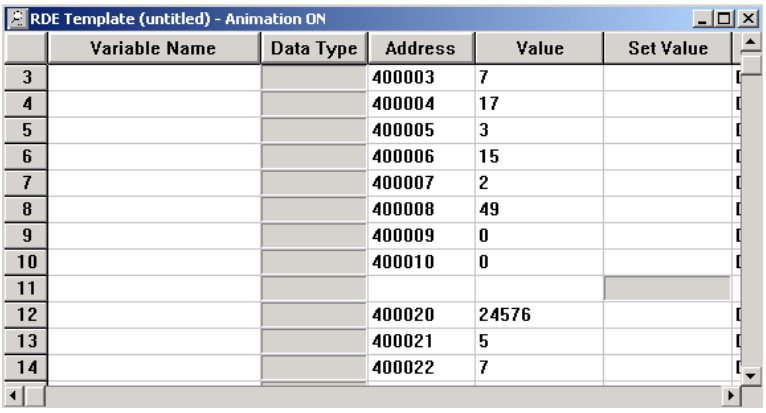
9 Notice that the template populates the address range, as shown in the following illustration. Place your cursor as shown in the first blank address field below the addresses you just entered.



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 and 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.

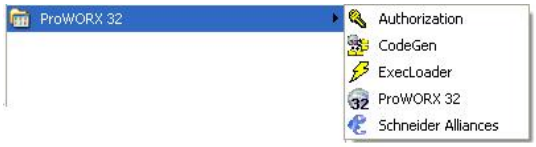
You are now ready to complete the installation and setup of the ProTalk module.

3 Configuring the Processor with ProWORX

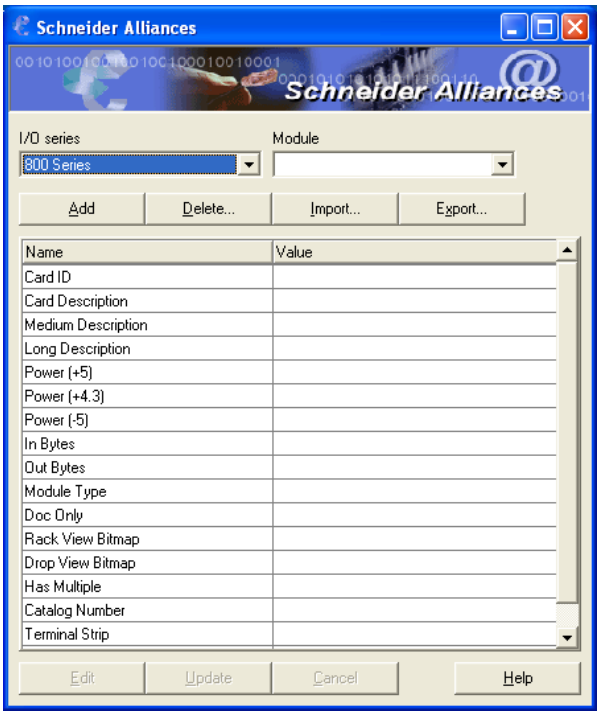
When you use ProWORX 32 software to configure the processor, use the example SAF file provided on the ProTalk Solutions CD-ROM.

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 consider this when monitoring the status of the PTQ module.

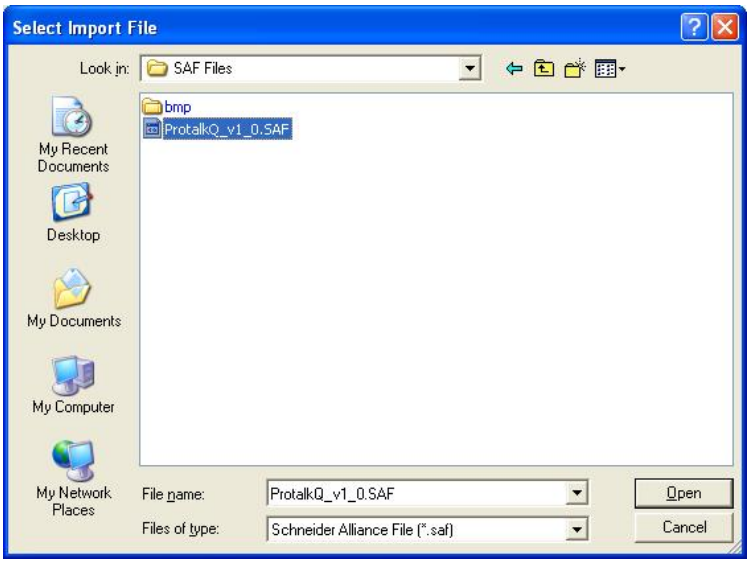
1 Run the **SCHNEIDER_ALLIANCES.EXE** application that is installed with the ProWORX 32 software:



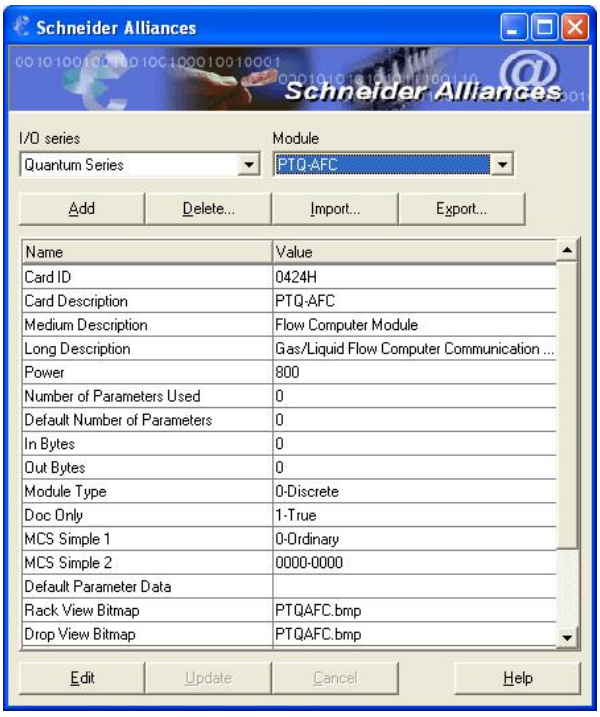
2 Click on **IMPORT…**



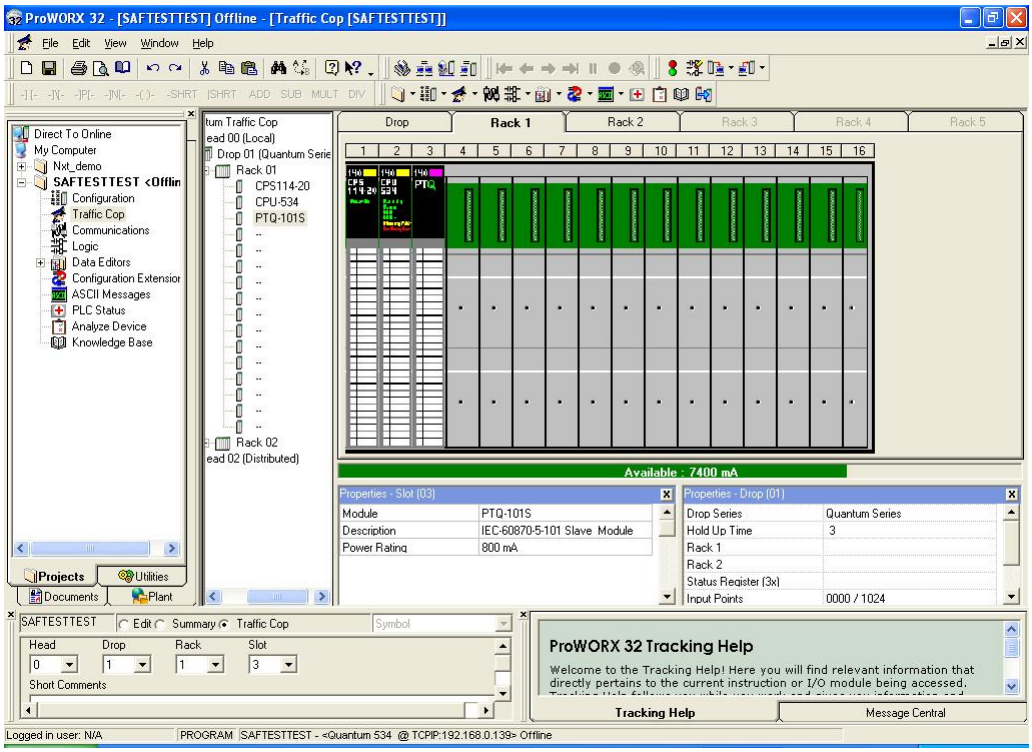
3 Select the .SAF File that is located on the CD-ROM shipped with the PTQ module.



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

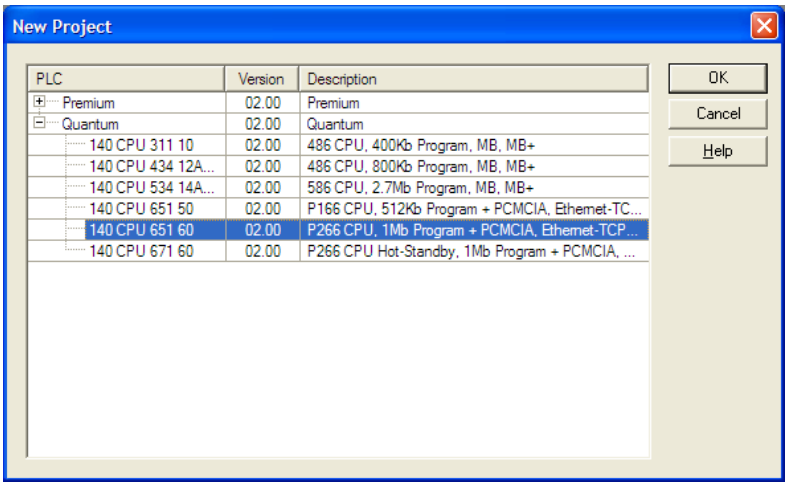
In This Chapter

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.

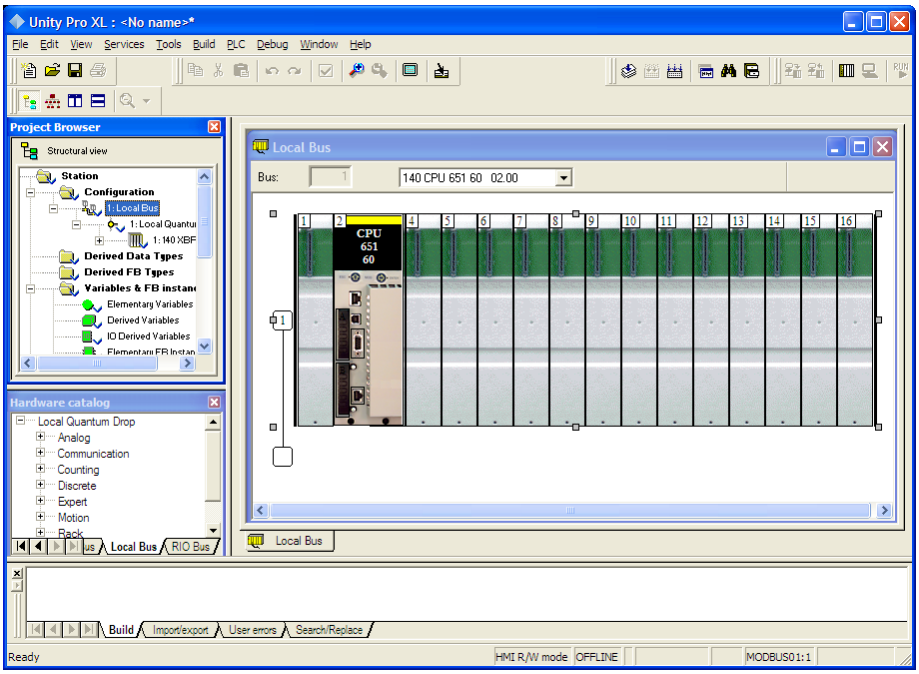
4.1 Create a New Project

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

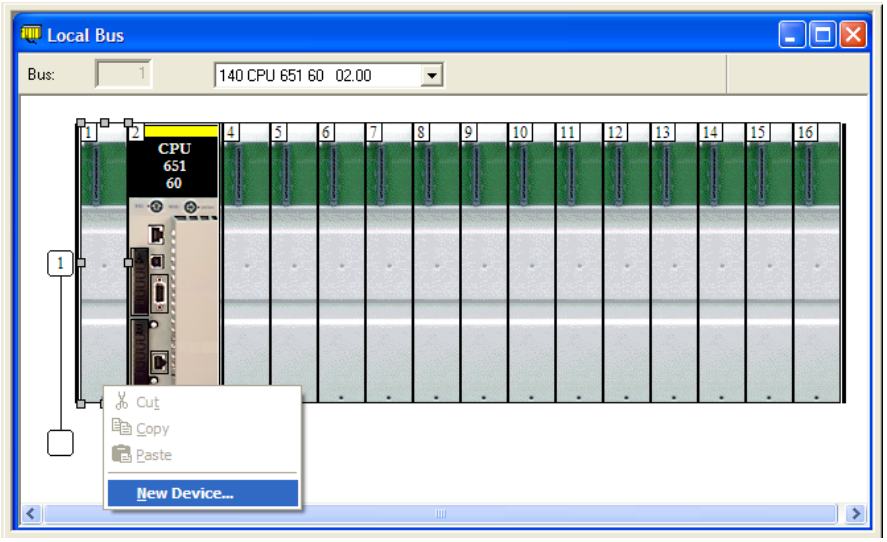
1 In the **NEW PROJECT** dialog box, choose the CPU type. In the following illustration, the CPU is 140 CPU 651 60. Choose the processor type that matches your own hardware configuration, if it differs from the example. Click **OK** to continue.



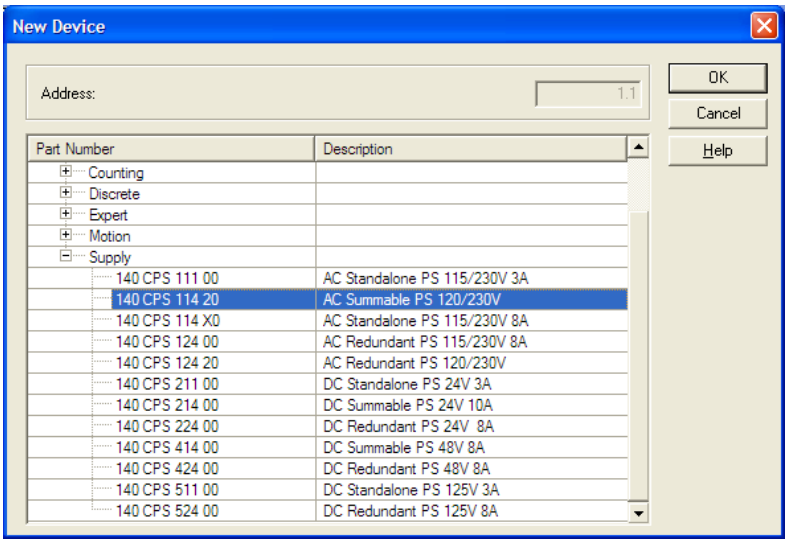
2 Next, add a power supply to the project. In the **PROJECT BROWSER**, expand the **CONFIGURATION** folder, and then double-click the **1:LOCALBUS** icon. This action opens a graphical window showing the arrangement of devices in your Quantum rack.



3 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**.



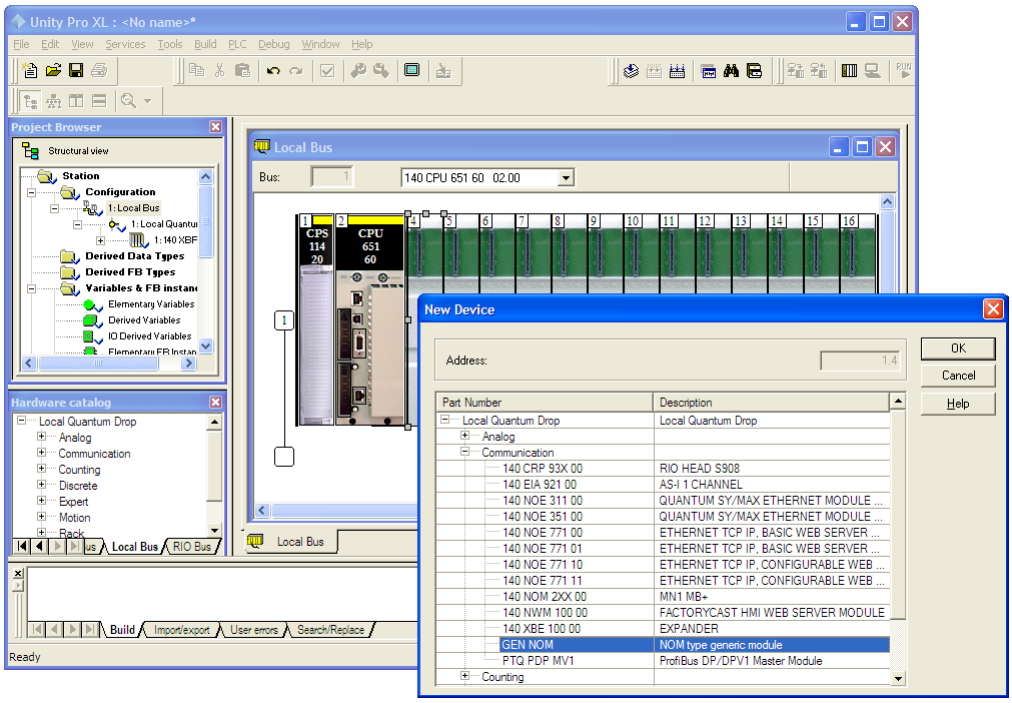
4 Expand the **SUPPLY** folder, and then select your power supply from the list. Click **OK** to continue.



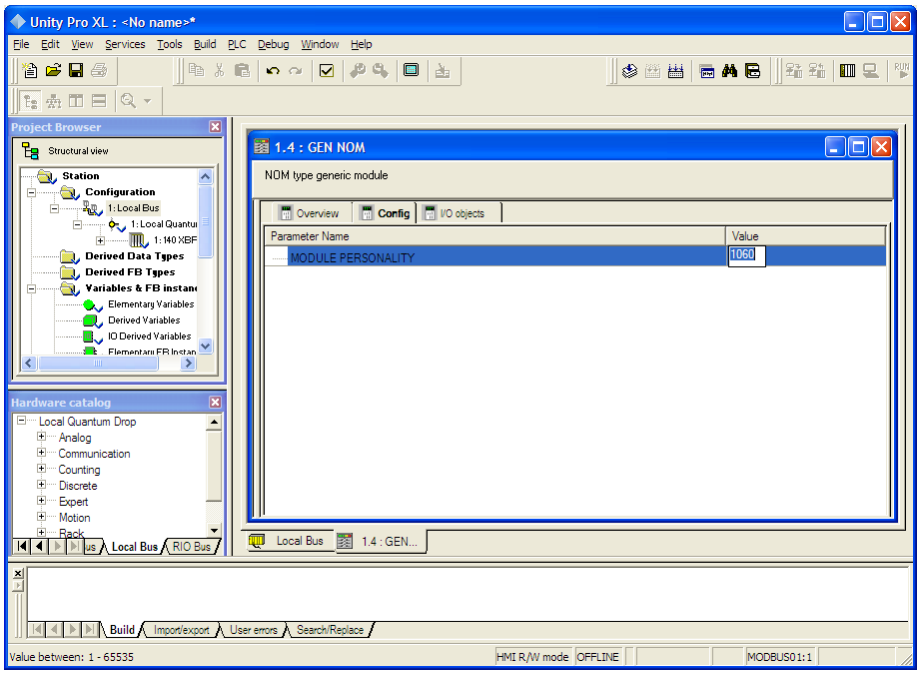
5 Repeat these steps to add any additional devices to your Quantum Rack.

4.2 Add the PTQ Module to the Project

1 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.



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



3 Before you can save the project in UnityPro XL, you must validate the modifications. Open the **EDIT** menu, and then choose **VALIDATE.** If no errors are reported, you can save the project.
4 Save the project.

4.3 Build the Project

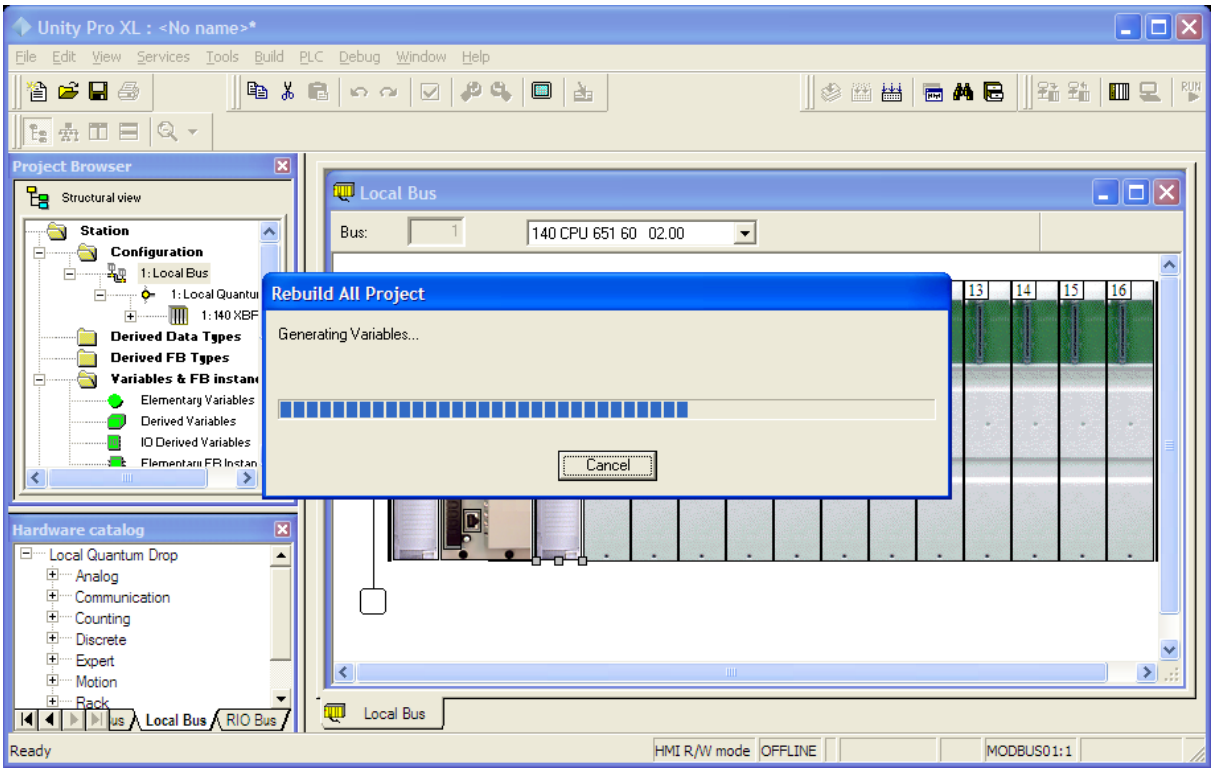
Whenever you update the configuration of your PTQ module or the processor, you must import the changed configuration from the module, and then build (compile) the project before downloading it to the processor.

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.

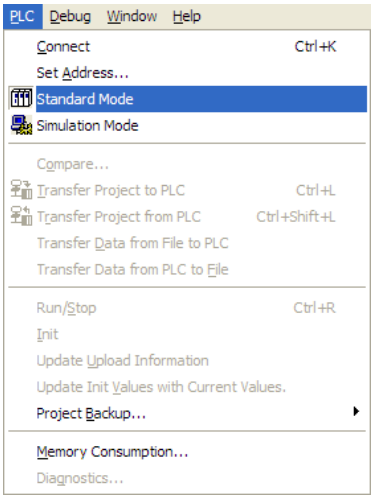
4.4 Connect Your PC to the Processor

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.

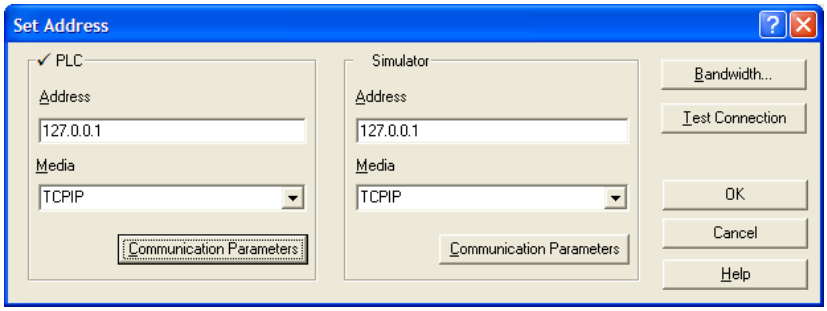
Note: If you have never connected from the PC to your processor before, you must verify that the necessary port drivers are installed and available to UnityPro XL.

To verify address and driver settings in UnityPro XL:

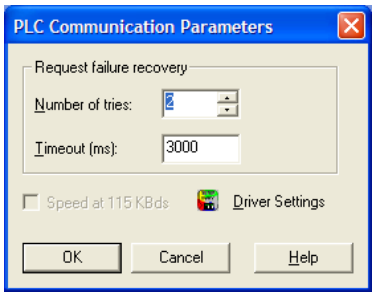
1 Open the **PLC** menu, and choose **STANDARD MODE**. This action turns off the PLC Simulator, and allows you to communicate directly with the Quantum or Unity hardware.



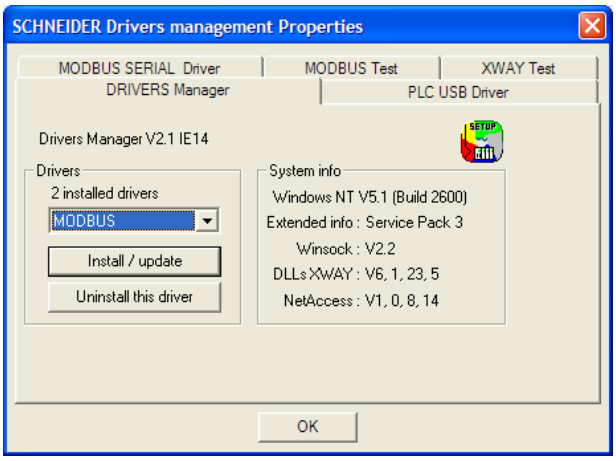
2 Open the **PLC** menu, and choose **SET ADDRESS...** This action opens the **SET ADDRESS** dialog box. Open the **MEDIA** dropdown list and choose the connection type to use (TCPIP or USB).



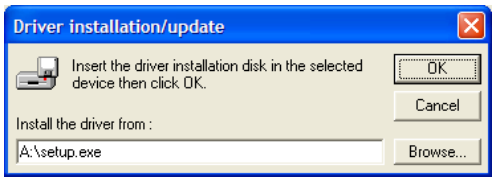
3 If the **MEDIA** dropdown list does not contain the connection method you wish to use, click the **COMMUNICATION PARAMETERS** button in the PLC area of the dialog box. This action opens the **PLC COMMUNICATION PARAMETERS** dialog box.



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.

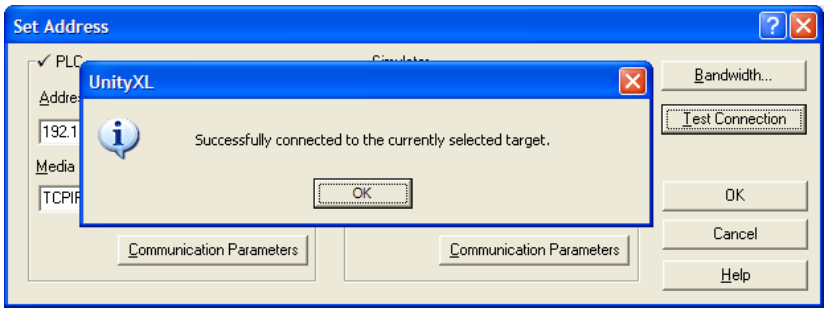
4.4.1 Connecting to the Processor with TCPIP

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**.

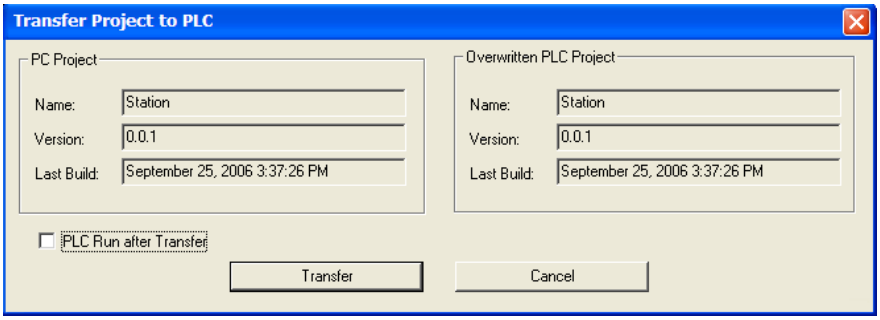
- Important: Notice that the SET ADDRESS dialog box is divided into two areas. Enter the address and media type in the PLC area of the dialog box, not the SIMULATOR area.

3. Enter the IP address in the address field. In the **MEDIA** dropdown list, choose TCPIP.
4. Click the **TEST CONNECTION** button to verify that your settings are correct.



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** 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

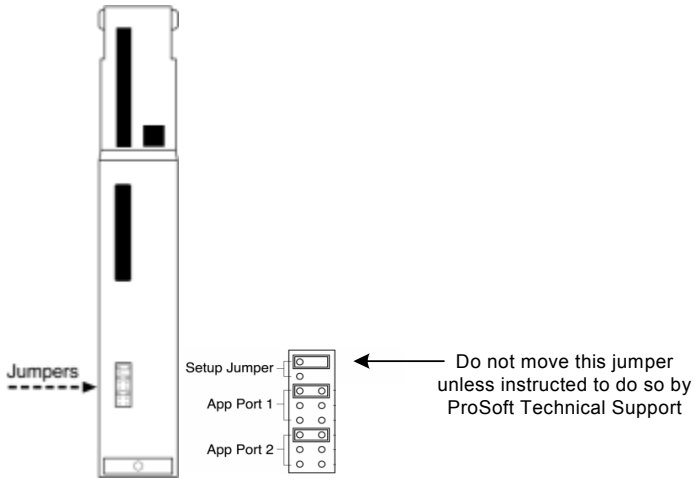
After you complete the following procedures, the ProTalk module will actively be transferring data bi-directionally with the processor.

5.1 Install the ProTalk Module in the Quantum Rack

5.1.1 Verify Jumper Settings

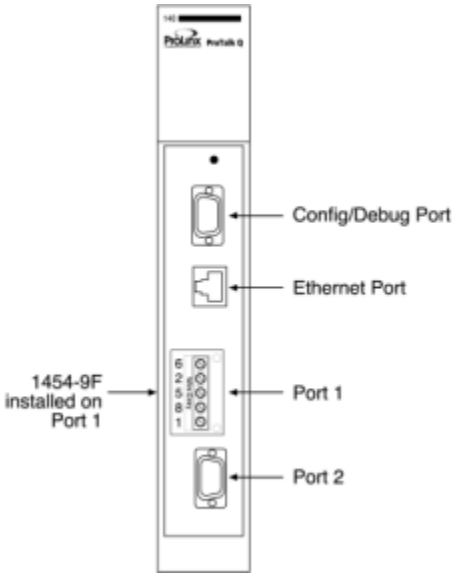
ProTalk modules are configured for RS-232 serial communications by default. To use RS-422 or RS-485, you must change the jumpers.

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



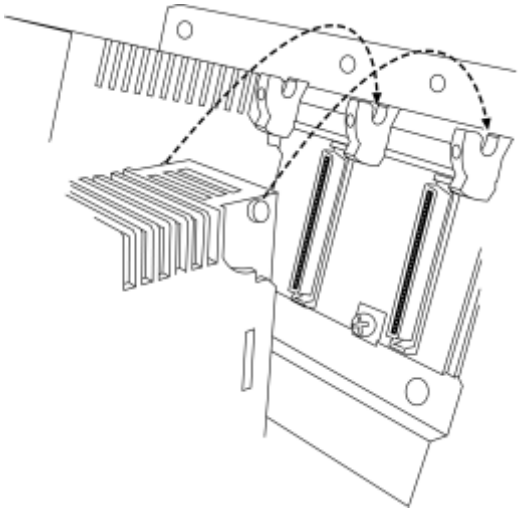
5.1.2 Inserting the 1454-9F connector

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

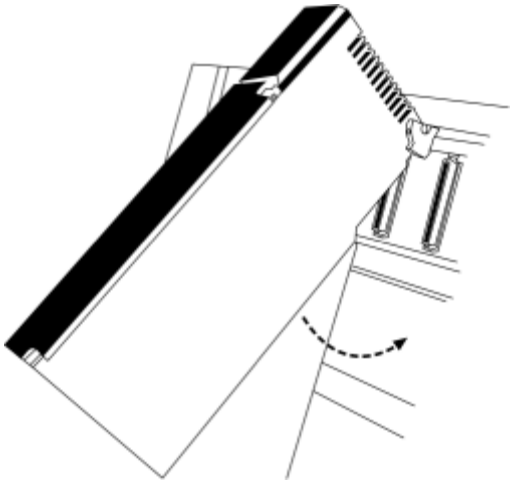


5.1.3 Install the ProTalk Module in the Quantum Rack

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.

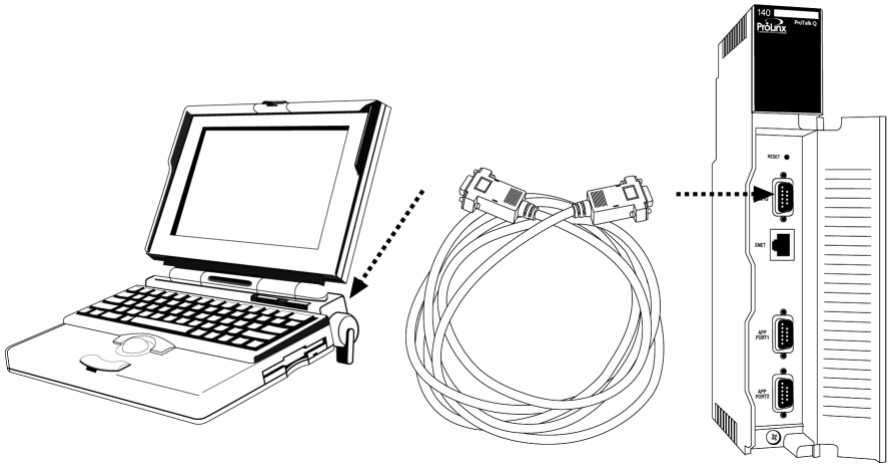


Caution: The PTQ module is hot-swappable, meaning that you can install and remove it while the rack is powered up. You should not assume that this is the case for all types of modules unless the user manual for the product explicitly states that the module is hot-swappable. Failure to observe this precaution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

5.2 Connect the PC to the ProTalk Configuration/Debug Port

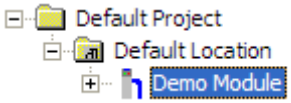
Make sure you have exited the Quantum programming software before performing these steps. This action will avoid serial port conflict.

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

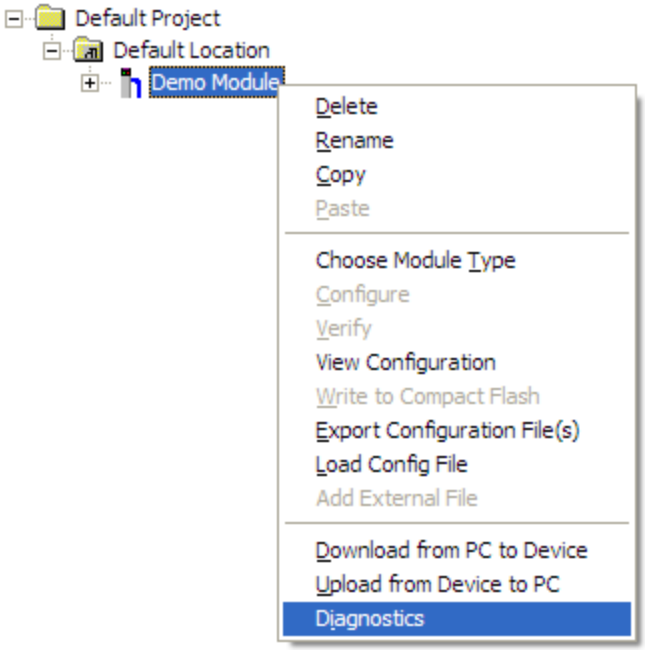


To connect to the module's Configuration/Debug serial port,

1 Start PCB, and then select the module to test. Click the right mouse button to open a shortcut menu.



2 On the shortcut menu, choose **DIAGNOSTICS.**



3 This action opens the **DIAGNOSTICS** dialog box. Press **[?]** to open the Main Menu.

```
IEC-870-5-103 MASTER COMMUNICATION MODULE
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  I=IEC-103 Master Menu
  P=Backplane Command List
  R=Receive Configuration File
  S=Send Configuration File
  V=Version Information
  Esc=Exit Program
```

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

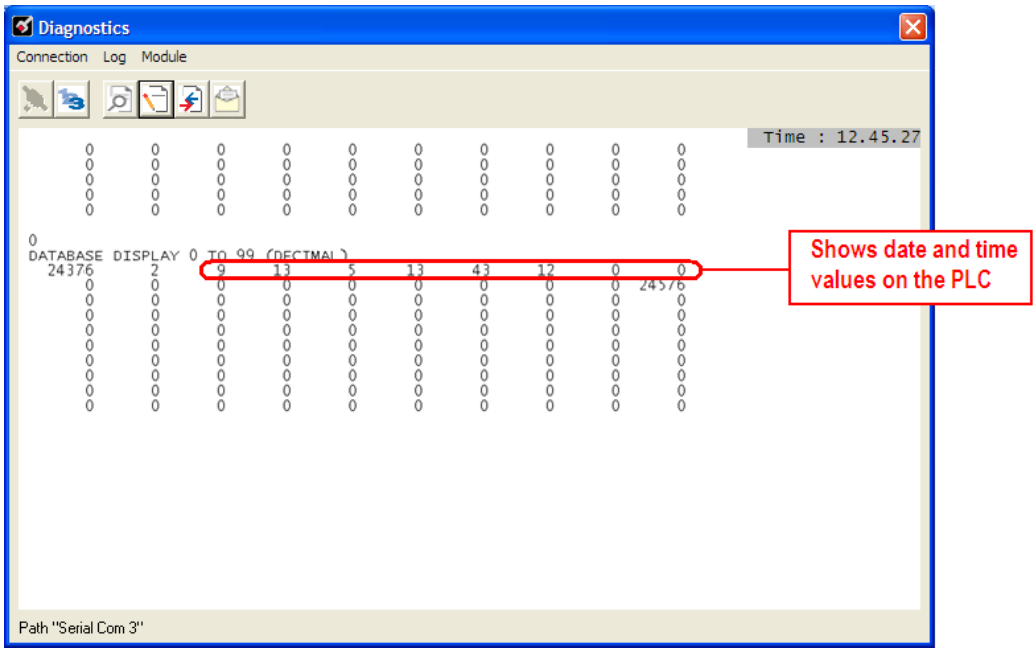
5.3 Verify Communication Between the Processor and the Module

This procedure will verify that the clock values we entered in the processor's data memory (page 21) 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.



Value	Description
9	Month (September)
13	Day of the Month
5	Year (2005)
13	Hour (13:00 or 1:00 P.M.)
43	Minutes
12	Seconds

In this example, the register values read from the PLC indicate that the date and time returned is September, 13, 2005, 1:43:12 p.m.

3 Type **[0]** again. The values should be different from 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 Configuring the PTQ-103M Module

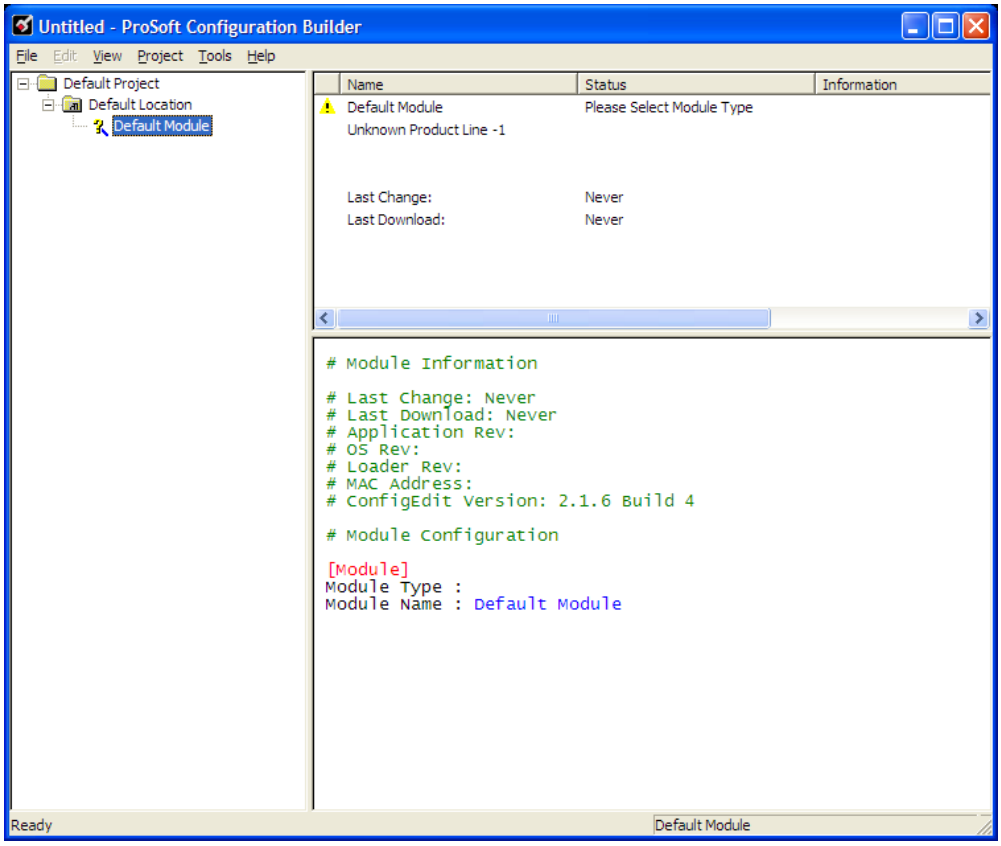
In This Chapter

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.

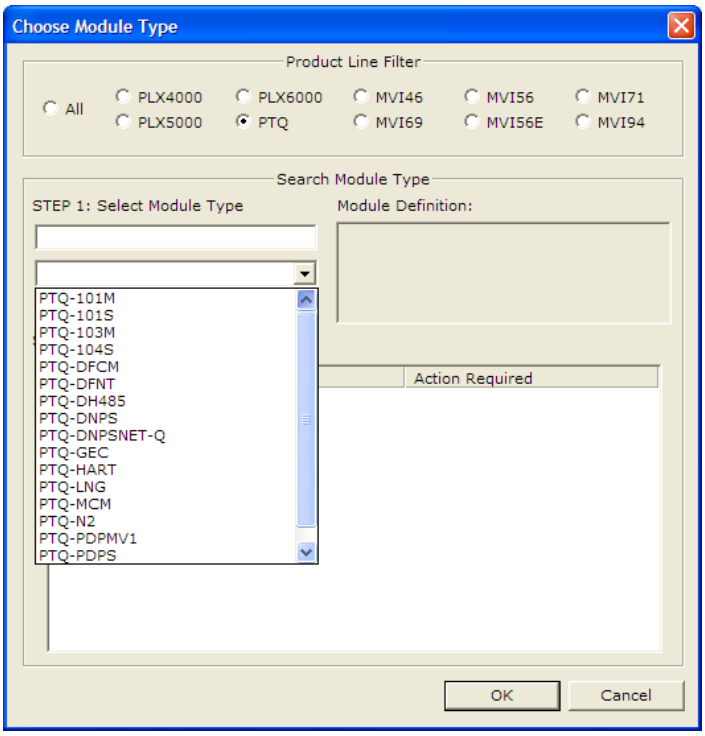
6.1.1 Set Up the Project

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-103M 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.

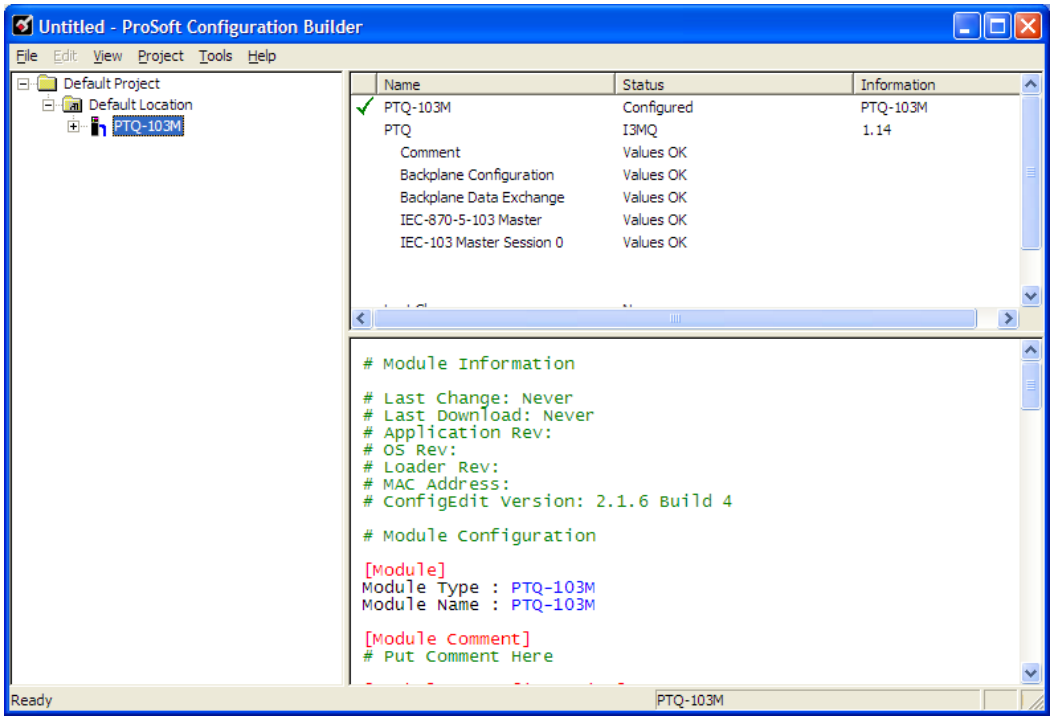


3. In the **PRODUCT LINE FILTER** area of the dialog box, select **PTQ.** In the **SELECT MODULE TYPE** dropdown list, select PTQ-103M, and then click **OK** to save your settings and return to the **PROSOFT CONFIGURATION BUILDER** window.

The next task is to set the module parameters.

6.1.2 Set Module Parameters

Notice that the contents of the information pane and the configuration pane changed when you added the PTQ-103M 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 to expand module information.
2. Double-click the 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.

Printing a Configuration File

To print a configuration file:

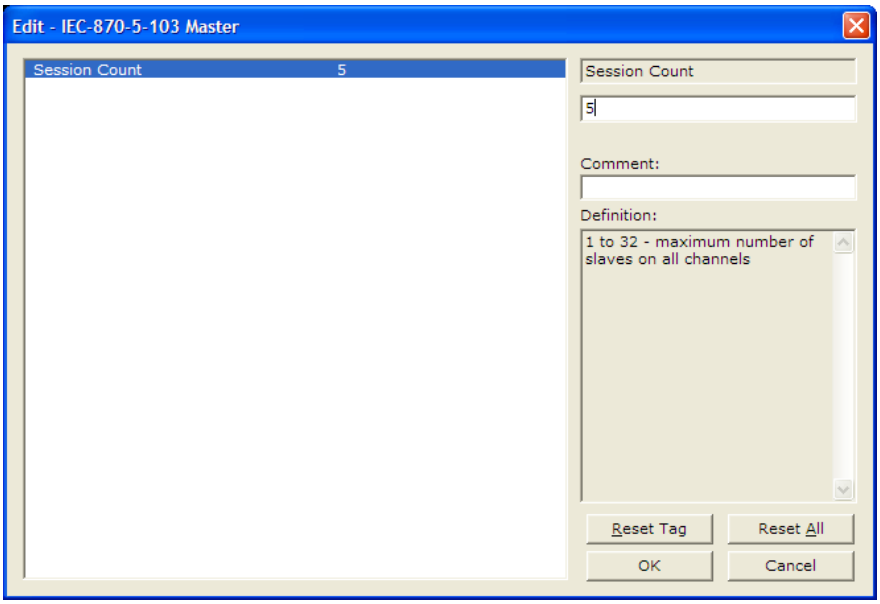
1. Select the **MODULE** icon, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose **VIEW CONFIGURATION.** This action opens the **VIEW CONFIGURATION** window.
3. On the **VIEW CONFIGURATION** window, open the **FILE** menu, and choose **PRINT.** This action 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 Quick Start

Step 1: Configure the Number of Slaves (Sessions)

The IEC 60870-5-103 protocol is a master-slave protocol where the slaves are typically protection equipments for substations. The PTQ-103M module supports a total 16 slaves (sessions) connected to the module's two application ports.

Note: The actual number of available sessions (slaves) will depend on the total number of sessions and sectors (configured. The recommended maximum number of sessions is sixteen.



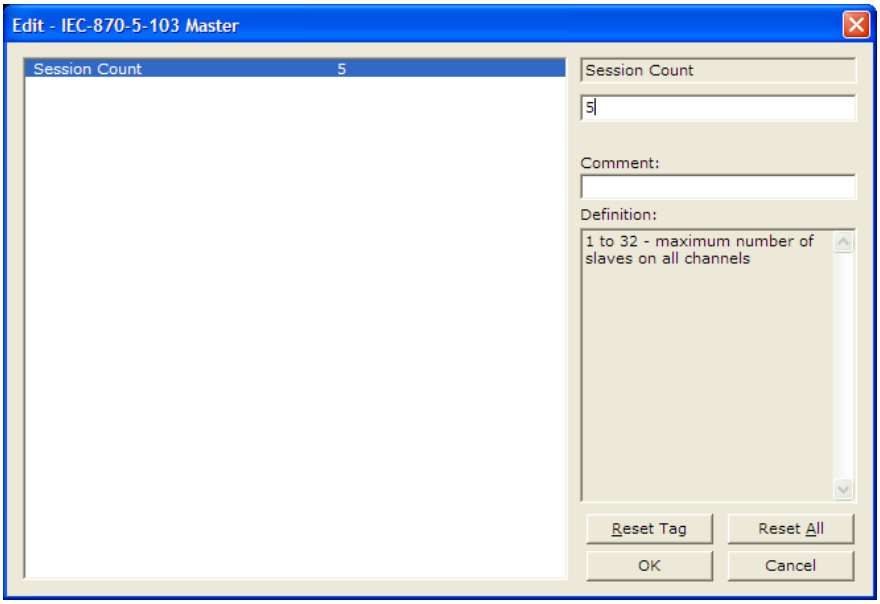
In the example above, the module will only poll sessions 0 to 4. The module would not poll sessions 5 to 31.

In Step 3, you will configure each session as an actual slave in the network.

Step 2: Configure the Port Communication Parameters

The user should configure the port communication parameters in order to enable data transfer between the master and the slave(s). The port communication parameters include baud rate, parity, RTS ON, RTS OFF, and Minimum Delay. The IEC 60870-5-103 protocol uses two baud rates: 19200 or 9600 kb/s and even parity.

Refer to the [IEC-870-5-103 Master Port 0] section in the configuration file in order to configure the communication parameters for the 103M port:



You must also configure the port jumpers to select the correct communication mode: RS-232, RS-422, or RS-485 (page 43).

Step 3: Configure the Session (Slave) Poll Parameters

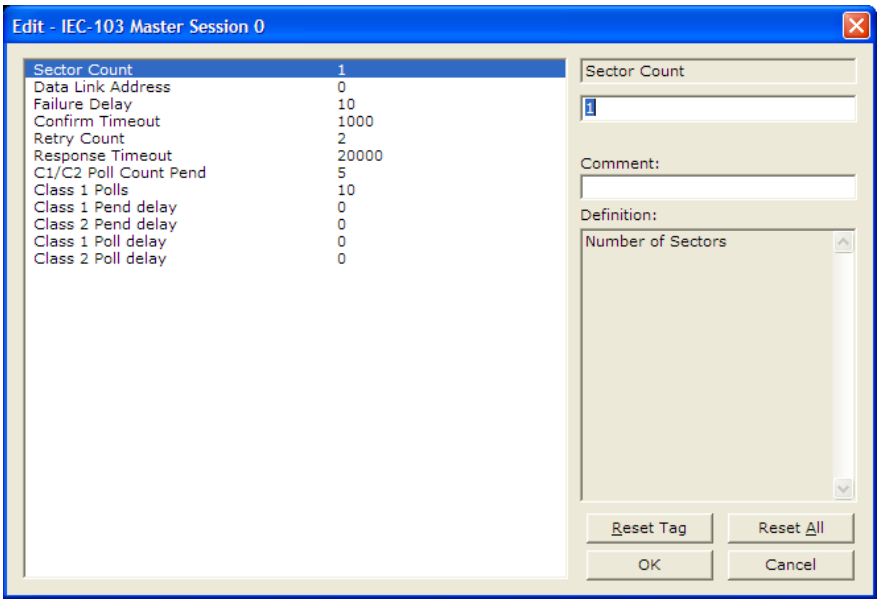
According to the IEC 60870-5-103 protocol, the master cyclically polls data from the slaves. The data is classified as Class 1 or Class 2. Events belong to Class 1, and analog data to Class 2. The module can request data through Class 1 or Class 2 requests. Responses to control command and general interrogation commands are also sent as Class 1 data.

Refer to the [IEC-103 Master Session x] section in the configuration file in order to configure how each slave will be polled.

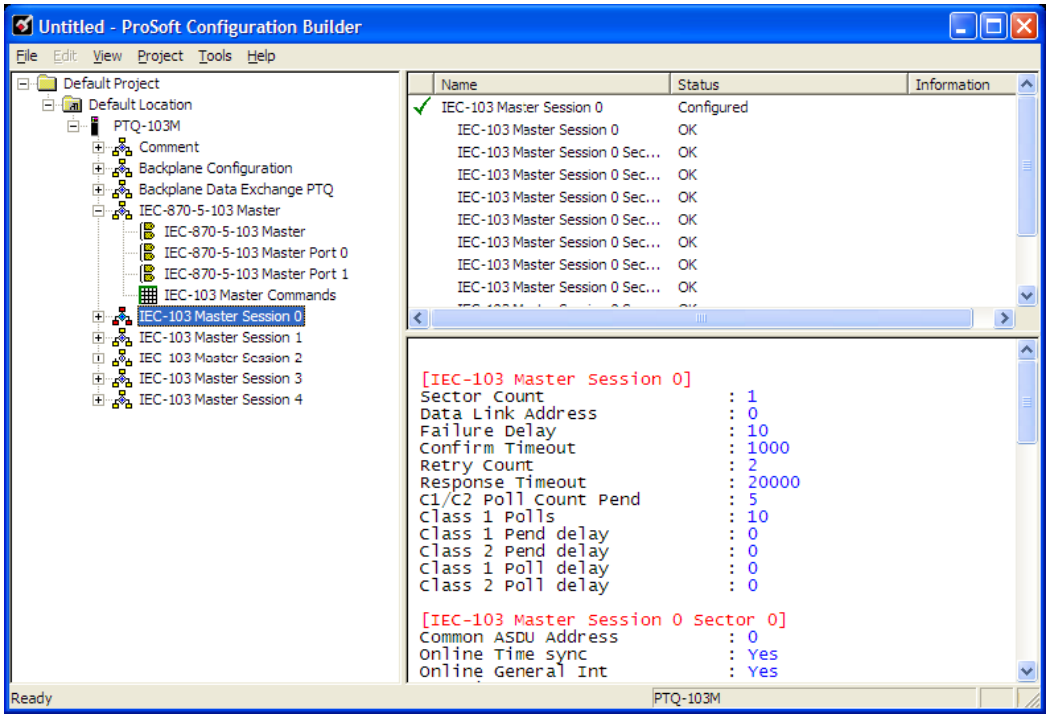
These parameters include the *Data Link Address,* which is the slave address that identifies each piece of protection equipment in the network. There should be a unique number for each slave in the network. There are also certain parameters that pertain to how the Class 1 and Class 2 polls will be used for data transfer.

You must enter the number of sectors for each session using the Sector Count parameter. The module accepts up to five sectors per session.

Note: Actual number of available sectors per session will depend on the total number of sessions and sectors configured. The recommended maximum number of sectors is three.



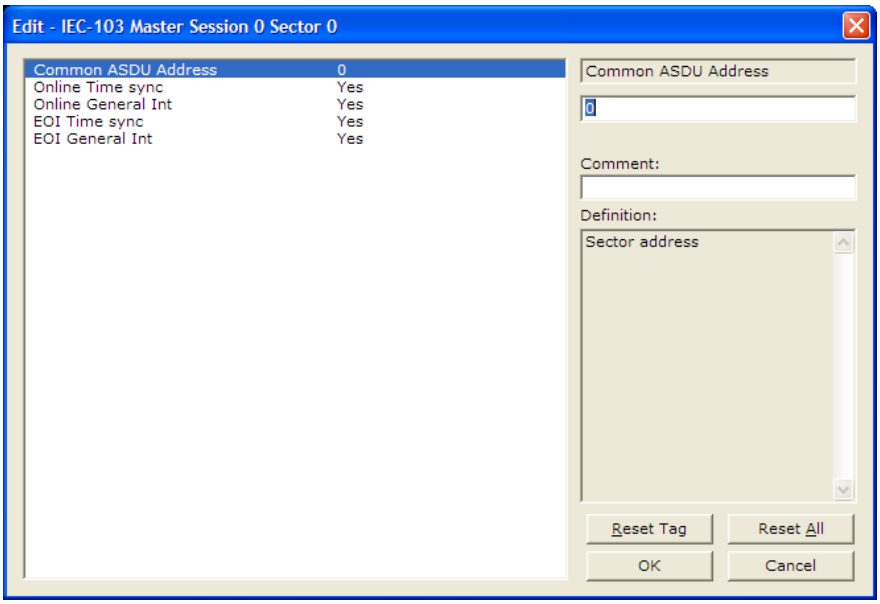
Repeat this step for each session. For example, if you configured 5 sessions during **Step 1,** configure Sessions 0 to 4.



Step 4: Sector (Data Set) Configuration

For each session (slave), you must configure one or more *sectors.* A sector is a data set defined by the vendor. Each sector is identified by the **COMMON ASDU ADDRESS** parameter in the [IEC-103 Master Session x Sector 0] area in the configuration. This area also contains some parameters that will affect the module initialization procedure.

Note: The actual number of available sectors per session will depend on the total number of sessions and sectors configured. The recommended maximum number of sectors is three.



Repeat this step for each sector used by the application. The module will only use the sectors configured in the previous step.

Step 5: Monitor Point Configuration (Monitor Direction)

When a slave receives a Class 1 or Class 2 request from the master, it responds with a message containing data. Each piece of equipment is normally configured to respond with specific points when it is being polled with a Class 2 request. During a Class 2 response, the slave may set a control bit (ACD) to inform the master that there are new events to be transmitted. Then, the master will send a Class 1 poll to read the events from the slave.

The IEC 60870-5-103 protocol states that the data is transferred between the master and slave using an ASDU (Application Service Data Unit) format. Each format is given by:

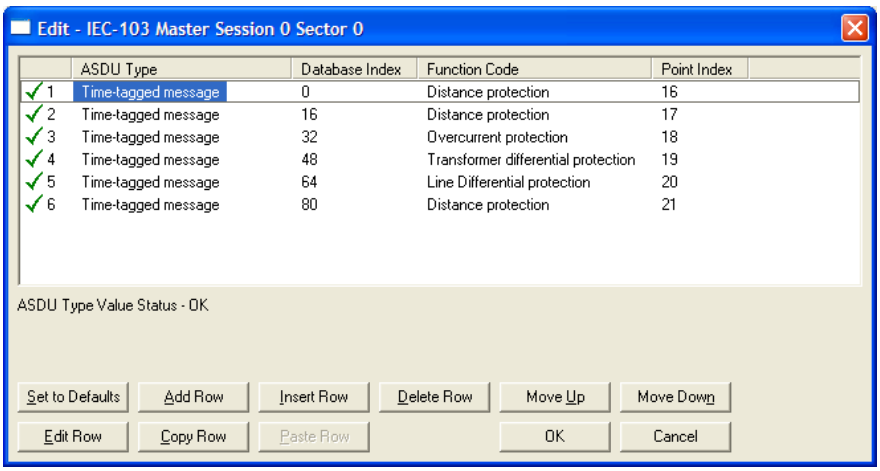
Type Identification
Variable Structure Qualifier
Cause Of Transmission
Common Address of ASDU

Type Identification
Function Type
Information Number
Data…
Data…
…

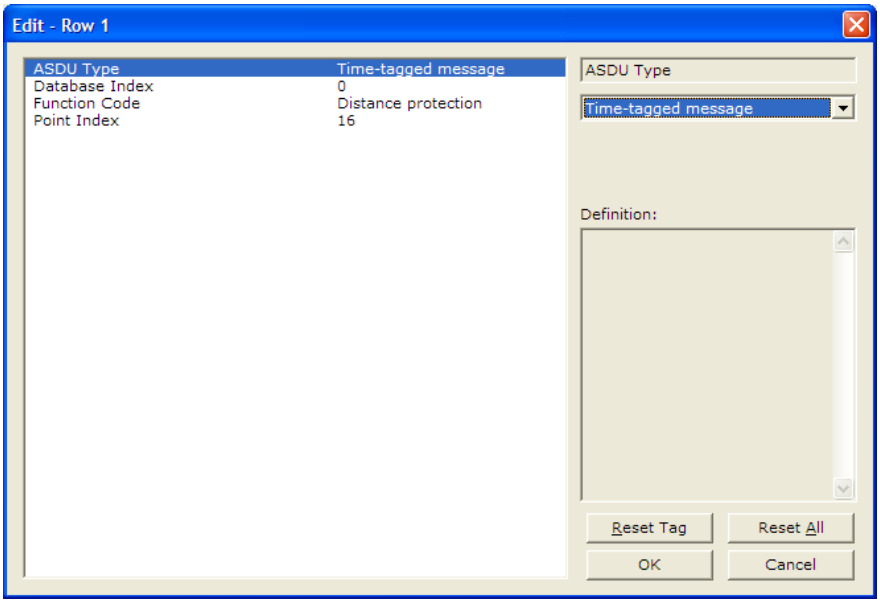
Refer to the protection equipment specification for the following information about each point:

- **Type:** Type of the message
- **Function Type:** Type of protection function
- **Information Number:** Point Identification

This information identifies each point in the configuration. You must configure the points that will be updated in the module database when a Class 2 or Class 1 response containing data is sent from the slave. Refer to **[IEC-103 MASTER SESSION X SECTOR 0]** to configure each point.



For each point, configure the following values.



ASDU Type: ASDU type for the point

Function Type: Function type for the point

Point Index: Information number for the point

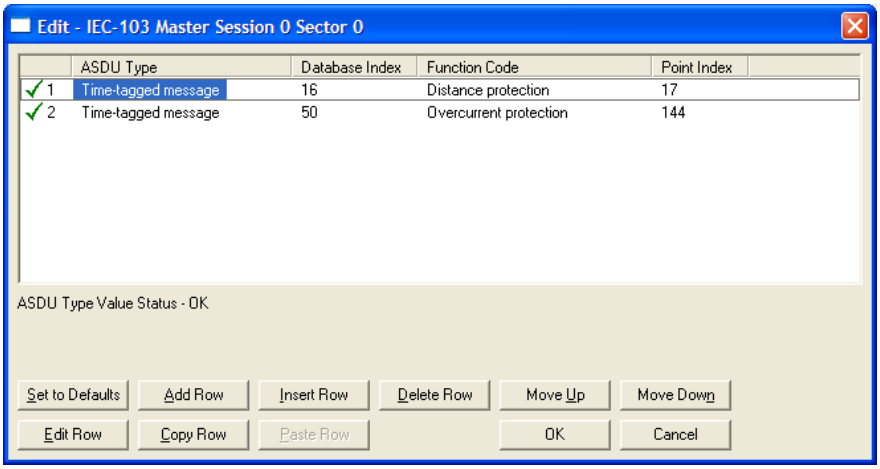
Database Index: The module database location where the value will be copied. The type of addressing will depend on the ASDU type:

ASDU Type	DB Addressing
1	Bit address with each point occupying 2 bits
2	Bit address with each point occupying 2 bits
3	Word address with each point occupying 4 words
4	Double-word address for the single float value
5	Byte address with each point occupying 12 bytes
9	Word address with each point occupying 9 words

For example, to configure the following points,

- Time-tagged message point with information number 17 (teleprotection active) and distance protection function (128). The value will be copied to bits 0 and 1 in word 1 (second word) in the module database.
- Measurands I point with information number 144 (measurands I) and overcurrent protection function (160). The value will be copied to word 50 in the module Database.

use the following configuration.



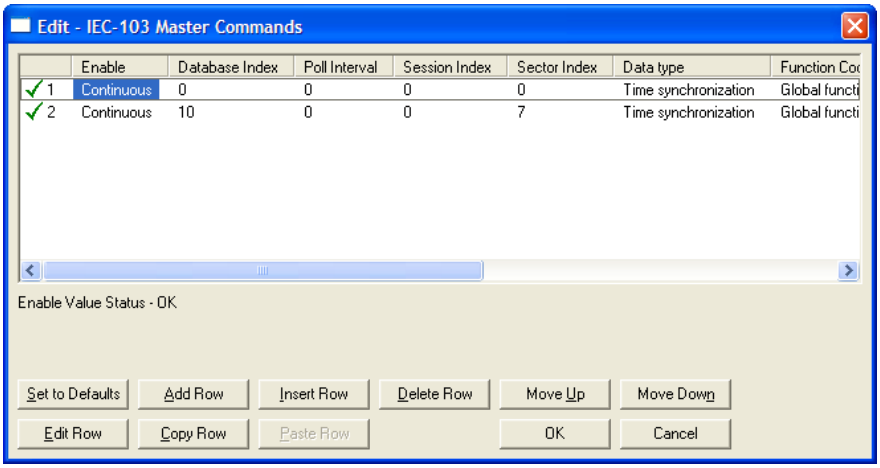
Every time the module responds with a Class 1 or Class 2 poll with these points, the module will copy the value to the database.

All the points configured in this section are sent from the slave to the master. The protocol specification refers to this data flow as the *Monitor Direction*.

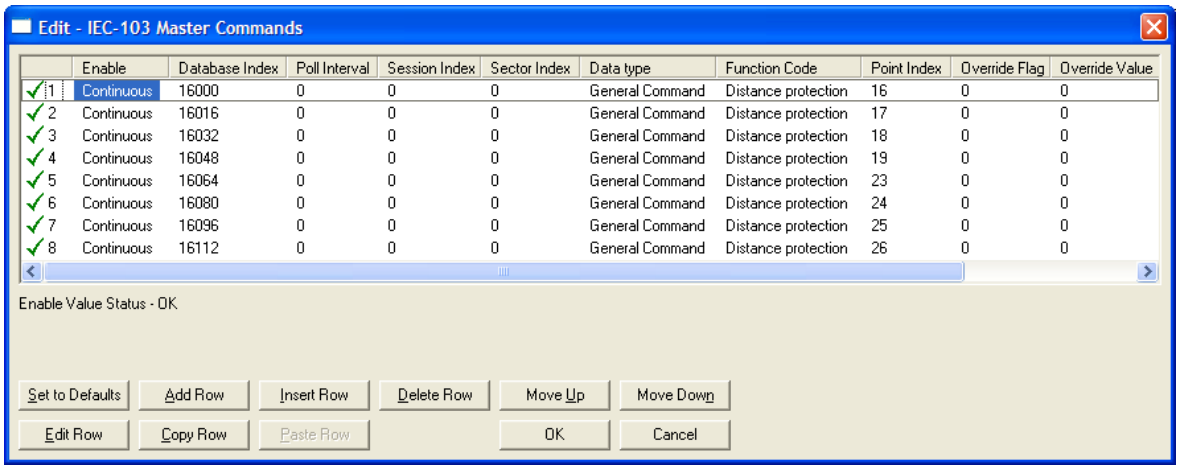
Repeat this step for each sector.

Step 6: Command Configuration (Control Direction)

You can also configure the master to send commands to slaves. The IEC 60870-5-103 protocol specification refers to this data flow as *Control Direction*. The commands include general commands, interrogation requests, and time synchronization requests. In order to configure a command, refer to the **[IEC-103 MASTER COMMANDS]** section:



To send a General Command, you can associate the source data with a register in the module database to be sent to the remote slave. The following example will send 8 commands to the slave configured as Session 0/Sector 0. Use bit addressing to send a General Command.



	Enable	Database Index	Poll Interval	Session Index	Sector Index	Data type	Function Code	Point Index	Override Flag	Override Value
1	Continuous	16000	0	0	0	General Command	Distance protection	16	0	0
2	Continuous	16016	0	0	0	General Command	Distance protection	17	0	0
3	Continuous	16032	0	0	0	General Command	Distance protection	18	0	0
4	Continuous	16048	0	0	0	General Command	Distance protection	19	0	0
5	Continuous	16064	0	0	0	General Command	Distance protection	23	0	0
6	Continuous	16080	0	0	0	General Command	Distance protection	24	0	0
7	Continuous	16096	0	0	0	General Command	Distance protection	25	0	0
8	Continuous	16112	0	0	0	General Command	Distance protection	26	0	0

Refer to the device specification for the Point Index (Information Number) listing available for control direction.

The module can also send a periodic General Interrogation command in order to initialize and refresh the event-updated points in its database. The slave keeps a list of all data subject to General Interrogation.

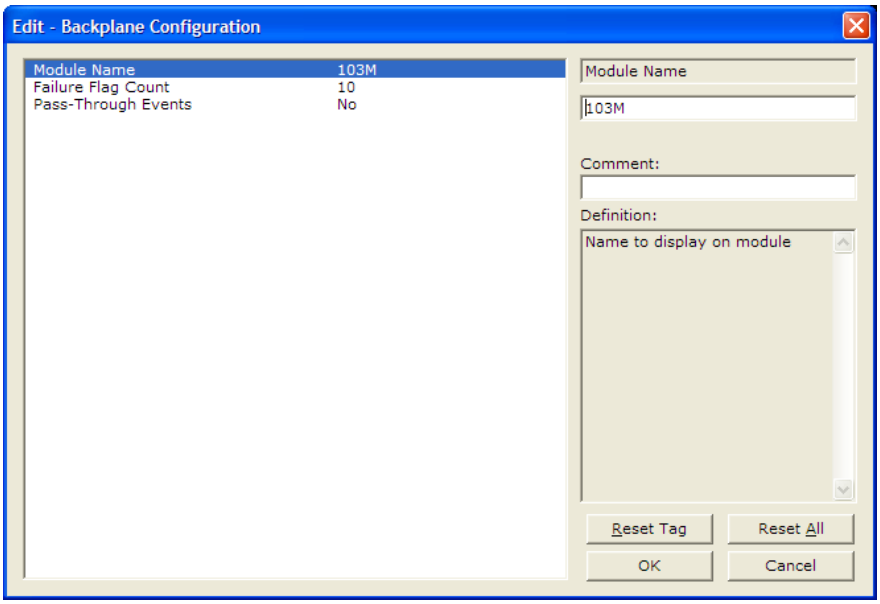
Step 7: Set the module's Data and Time (optional)

If the module will be sending time synchronization commands to the slave, you must set the date and time on the module (page 116).

Step 8: Transfer the Configuration from the Computer to the module (page 72).

6.3 [Backplane Configuration]

This section provides the module with a unique name, identifies the method of failure for the communications for the module if the processor is not in run mode, and describes how to initialize the module upon startup.



6.3.1 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.

6.3.2 Failure Flag Count

0 through 65535

This parameter specifies the number of successive transfer errors that must occur before the communication ports are shut down. If the parameter is set to 0, the communication ports will continue to operate under all conditions. If the value is set larger than 0 (1 to 65535), communications will cease if the specified number of failures occur.

6.3.3 Pass-Through Events

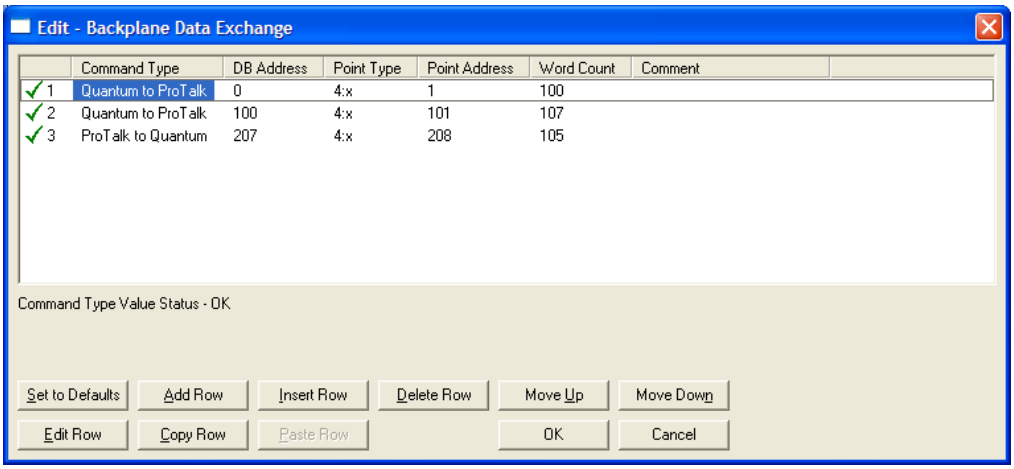
Yes, No

Pass event messages to processor. This setting must be enabled if you plan to use Command Type 9903 for pass-through messages.

6.4 [Backplane Data Exchange]

The PTQ-103M module uses Command Functions to transfer 64-word blocks of data between the processor and the module, and to execute special functions such as setting the module's clock, or forcing a reboot.

The following illustration shows a typical **[BACKPLANE DATA EXCHANGE]** section. Please refer to Backplane Data Exchange for a detailed discussion on how to configure data exchange (page 101).



6.4.1 Command Type

Command	Description
Disabled	The command will not be executed
Quantum to ProTalk	The command will copy data from the Quantum processor to the PTQ-103M module
ProTalk to Quantum	The command will copy data from the PTQ-103M module to the Quantum processor
Control Block	The command will execute a Special Function (page 110)
Pass-Through events	The command will execute a Pass-Through Function (page 119)

6.4.2 DB Address

This is the address of the starting word in the PTQ-103M database.

6.4.3 Point Type

Point Type	Description
0x	coil data
1x	input coil data
3x	input registers
4x	holding registers

6.4.4 Point Address

Point address (1 based)

0x and 1x must be at start of word (that is, 1, 17, 33, ...)

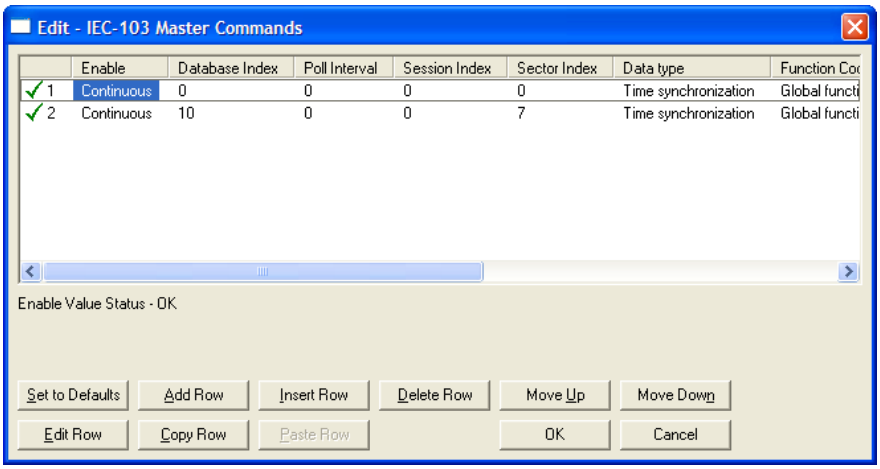
6.4.5 Word Count

1 to 130

Number of words to transfer

6.5 [IEC-103 Master Commands]

This section can contain up to 1000 user defined commands to be executed by the module and sent to the controlled devices. There is no need to place Class 1 or Class 2 polls in this list for the controlled devices as the master driver for each port will execute these automatically when the port is idle. In order for the port to be idle, make sure that there is idle time available, and that the commands do not constantly utilize the ports.



6.5.1 Enable Code

0 = Disabled

1 = Enabled with Poll Interval (seconds) utilized

2 = Conditional (executed when point in database changes)

This field defines whether the command is to be executed, and under what conditions.

To disable the command, set this parameter to 0. You can still execute commands through the processor, using a Command Control block.

To enable the command, set this parameter to 1.

- Set the Poll Interval Time to 0 to execute the command during each scan of the command list.
- Set the Poll Interval Time to a value in seconds, to execute the command at the specified interval (page 64).

To execute the command only if the internal data associated with the command changes, set this parameter to 2. This value is valid only for write commands.

6.5.2 Database Index

Database Index is the location in the module's database to use as the source for the data in the command. Refer to Data Type for specific information on addressing (page 64).

The data type field determines the meaning of the database index as follows:

Type	Description	DB Index type
6	Clock synchronization	NA
7	General interrogation	NA
20	General Command	Bit address

6.5.3 Poll Interval

This parameter specifies the minimum frequency at which the module should execute the command. The value is entered in units of seconds. For example, to execute a command every 10 seconds, enter a value of 10 in the field. A value of 0 for the parameter implies that the command should be executed every scan of the list.

6.5.4 Session Index

Session Index represents the session index in the module to associate with the command. This index is set when the session is read in from this file. The range of values for this field is 0 to 31.

6.5.5 Sector Index

Sector Index represents the sector index for the specific session. The range of values for this field is 0 to 2.

6.5.6 Data Type

Data type file represents the ASDU type as follows:

Type	Description
6	Clock synchronization
7	General interrogation
20	General Command

6.5.7 Function Code

Code	Definition
128	Distance protection
160	Overcurrent protection
176	Transformer differential protection
192	Line Differential protection
205	For SIEMENS ASDU type 205
255	Global function type

6.5.8 Point Index

Point Index specifies the address in the remote slave device of the point to interact with.

Index Value	Description
1	Bit address with each point occupying 2 bits
2	Bit address with each point occupying 2 bits
3	Word address with each point occupying 4 words
4	Double-word address for the single float value
5	Byte address with each point occupying 12 bytes
9	Word address with each point occupying 9 words

6.5.9 Override Flag

0 or 1

Override Flag field is used for general commands to determine the value to be written. If the override flag is clear (0), the value in the database will be utilized. If the override flag is set (1), the value specified in the override value field will be used.

6.5.10 Override Value

If the Override Flag is set to "Yes", you can use this setting to always force a control parameter to a fixed value. Use Enable code "Conditional" and the database value for the command to determine when the value should be written.

6.6 [IEC-870-5-103 Master]

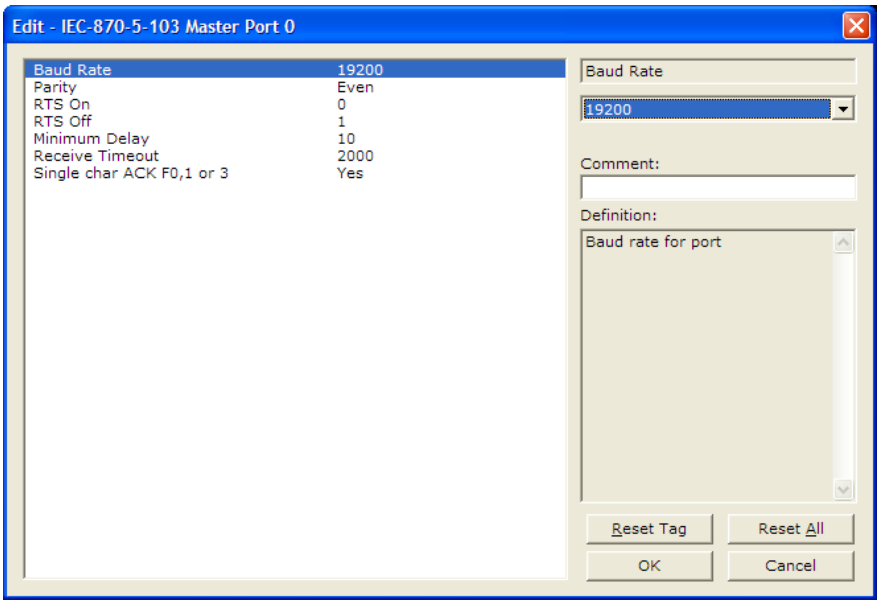
6.6.1 Session Count

1 to 16

This parameter specifies the maximum number of sessions (slaves) to interface with the module's IEC 60870-5-103 application ports. This value represents the total number of slaves on all ports.

6.7 [IEC-870-5-103 Master Port x]

These settings configure the communication parameters for each application port on the module. The following illustration shows typical settings for a Master port.



6.7.1 Baud Rate

19200 or 9600

This parameter specifies the baud rate to be used on the communication channel (port).

6.7.2 Parity

N, O, E, M, or S

This parameter sets the parity to be used on the port. The values correspond to the following settings: N=None, O=Odd, E=Even, M=Mark and S=Space.

Note: The IEC 60870-5-103 specification supports only Even Parity.

6.7.3 RTS On

0 to 65535 milliseconds

This parameter sets the number of milliseconds to delay after RTS is asserted before the data will be transmitted.

6.7.4 RTS Off

0 to 65535 milliseconds

This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low.

6.7.5 Minimum Delay

1 to 65535

This parameter specifies the minimum number of milliseconds to delay before sending the message (setting RTS high). This can be used when the serial network requires time for units to turn off their transmitters.

6.7.6 Receive Timeout

1 to 65535

This value represents the number of milliseconds to wait on a port from the time the first character is received until the last character in the longest message received on the port. This parameter should be dependent on the baud rate. A value of 2000 should work with most applications.

6.7.7 Single char ACK F0, 1, or 3

Yes or No

This parameter specifies if the signal E5 character will be used for ACK messages.

6.8 [IEC-101 Master Session x]

These sections define session *x,* which runs on Port *x*. These sections define the characteristics of the specific controlled device to interface. The **SESSION COUNT** parameter in the **[IEC-870-5-103 MASTER PORT X]** section of the configuration (page 65) determines the number of Sessions (controlled devices) for this port.

The sessions are referenced by a zero based index value. For example, if the module is configured for four sessions, the configuration file should contain sections for sessions 0 to 3 (that is, [IEC-101 Master Session 0] to [IEC-101 Master Session 3].

6.8.1 Communication Port

0 or 1

This parameter sets the port to which the controlled device is connected. On this module, values of 0 and 1 are permitted.

6.8.2 Sector Count

1 to 3

This parameter sets the number of Sectors (separate databases or Multiple Application Layer ASDU addresses) contained in this Session (controlled device). This version of the application supports 1 to 3 sectors for each session.

6.8.3 Data Link Address

0 to 254

This parameter uniquely defines the data link address for this unit on the communication channel. The ranges of values are from 0 to 254. Address 255 is the broadcast address.

6.8.4 Failure Delay

0 to 2000

This parameter sets the minimum number of seconds to delay before polling this session when it is not online. This parameter is only used in unbalanced mode.

6.8.5 Confirm Timeout

0 to 2^32-1

This parameter sets the number of milliseconds to wait for a confirm response from the controlled device.

6.8.6 Retry Count

0 to 255

This parameter sets the number of retries to be performed on the controlled device when a communication occurs.

6.8.7 C1/C2 Poll Count Pend

0 to 65535

This parameter sets the maximum number of class 1 and class 2 polls performed on this session before trying the next session. This parameter prevents a session from monopolizing the communication port.

6.8.8 Class 1 Polls

0 to 100

This parameter sets the maximum number of class 1 polls performed on this session before switching to another session. This parameter prevents a session from monopolizing the communication port.

6.8.9 Class 1 Pend Delay

0 to 2^32-1

This parameter sets the minimum number of milliseconds to delay between class 1 polls for pending data.

6.8.10 Class 2 Pend Delay

0 to 2^32-1

This parameter sets the minimum number of milliseconds to delay between class 2 polls for pending data.

6.8.11 Class 1 Poll Delay

0 to 2^32-1

This parameter sets the minimum number of milliseconds to delay between each class 1 poll.

6.8.12 Class 2 Poll Delay

0 to 2^32-1

This parameter sets the minimum number of milliseconds to delay between each class 2 poll.

6.8.13 Auto Clock Req Mode

0=Sync Only, 1=Load delay/sync, 2=Acquire delay/load delay/sync

This parameter specifies the method used to perform automatic clock synchronization. 0 performs a synchronization without delay, 1 performs synchronization using the fixed Propagation Delay and 2 computes the delay and use this value when synchronization takes place.

6.8.14 Propagation Delay

0 to 65535

This parameter sets the fixed propagation delay to be utilized if the Auto Clock Req Mode parameter is set to a value of 1.

6.8.15 Response Timeout

0 to 2^32-1 milliseconds

This parameter sets the maximum number of milliseconds to wait for a confirmation from the controlled station to a request from this module.

6.8.16 ACTTERM with setpoint

Yes or No

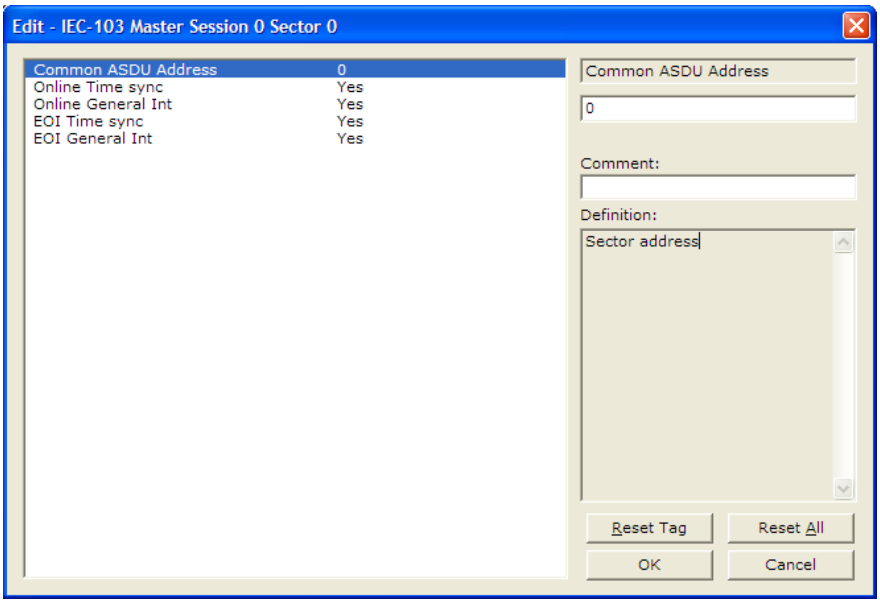
This parameter determines if an ACTTERM will be sent. If the parameter is set to Yes, then setpoint commands will issue an ACTTERM when the command is complete. If the parameter is set to No, ACTCON is the last response to a setpoint command.

6.9 [IEC-103 Master Session x Sector y]

These sections define the Sectors for each Session. The Sector Count parameter (page 67) specifies the number of sectors for the session

Each sector has a corresponding **[IEC-103 MASTER SESSION X SECTOR Y]** section, where *x* represents the session index and *y* represents the sector index.

The sectors are referenced by a zero based index value. For example, if session 0 contains 1 sector, the corresponding definition will be in the section [IEC-103 Master Session 0 Sector 0].



6.9.1 Common ASDU Address

0 to 255

This parameter sets the common ASDU address to association with this sector of the specified session. This parameter is usually set the same as the data link address when only one sector is used.

6.9.2 Online Time Sync.

Yes or No

This parameter specifies if the sector in the controlled device will be sent a time synchronization command when the unit is first recognized as being online. This should only be used for devices that do not send an EOI message after initializing.

6.9.3 Online General Int

Yes or No

This parameter specifies if the sector in the controlled device will be sent a general interrogation command when the unit is first recognized as being online. This should only be used for devices that do not send an EOI message after initializing.

6.9.4 EOI Time Sync.

Yes or No

This parameter specifies if the sector in the controlled device will be sent a time synchronization command after this module received an EOI message from the controlled unit.

6.9.5 EOI General Int

Yes or No

This parameter specifies if the sector in the controlled device will be sent a general interrogation command after this module received an EOI message from the controlled unit.

6.9.6 ASDU Type

This field contains the ASDU type code for the data contained in the message.

- 1 = Time-tagged message (bit addressed with 2 bits/point)
- 2 = Time-tagged message with relative time (bit addressed with 2 bits/point)
- 3 = Measurands I (4 word values using word address using double-word address)
- 4 = Time-tagged measurands with relative time (1 float value)
- 5 = Identification (12 characters using a byte address)
- 9 = Measurands II (9 word values using word address)

6.9.7 Database Index

Database Index is the location in the module's database to use as the source for the data in the command. Refer to Data Type for specific information on addressing (page 64).

The data type field determines the meaning of the database index as follows:

Type	Description	DB Index type
6	Clock synchronization	NA
7	General interrogation	NA
20	General Command	Bit address

6.9.8 Function Code

Code	Definition
128	Distance protection
160	Overcurrent protection
176	Transformer differential protection
192	Line Differential protection
205	For SIEMENS ASDU type 205
255	Global function type

6.9.9 Point Index

Point Index specifies the address in the remote slave device of the point to interact with.

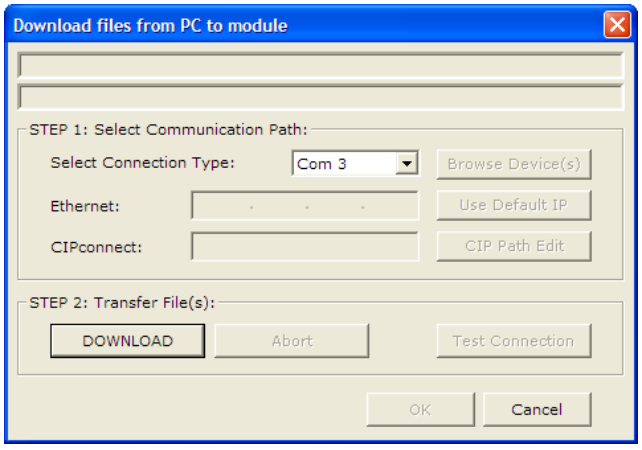
Index Value	Description
1	Bit address with each point occupying 2 bits
2	Bit address with each point occupying 2 bits
3	Word address with each point occupying 4 words
4	Double-word address for the single float value
5	Byte address with each point occupying 12 bytes
9	Word address with each point occupying 9 words

6.10 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.

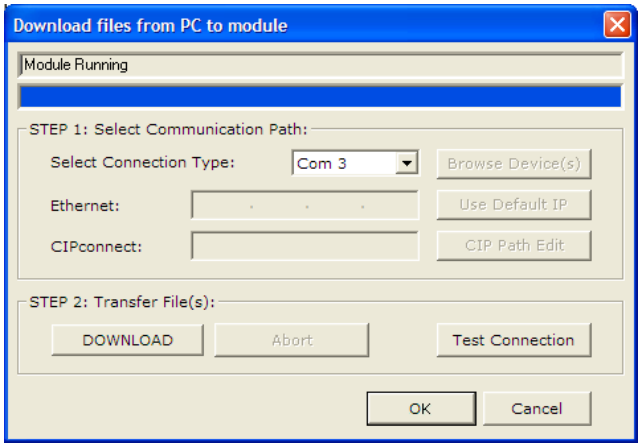
To Download the Project File

1. In the tree view in ProSoft Configuration Builder, click once to select the PTQ-103M 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 **DOWNLOAD** 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 the **DOWNLOAD** dialog box with the message *"Module Running*".



7 Diagnostics and Troubleshooting

In This Chapter

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-103M module provides the status data in each read block. This data can also be located in the module's database.

7.2 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.

7.3 The Configuration/Debug Menu

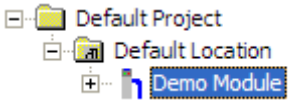
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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.

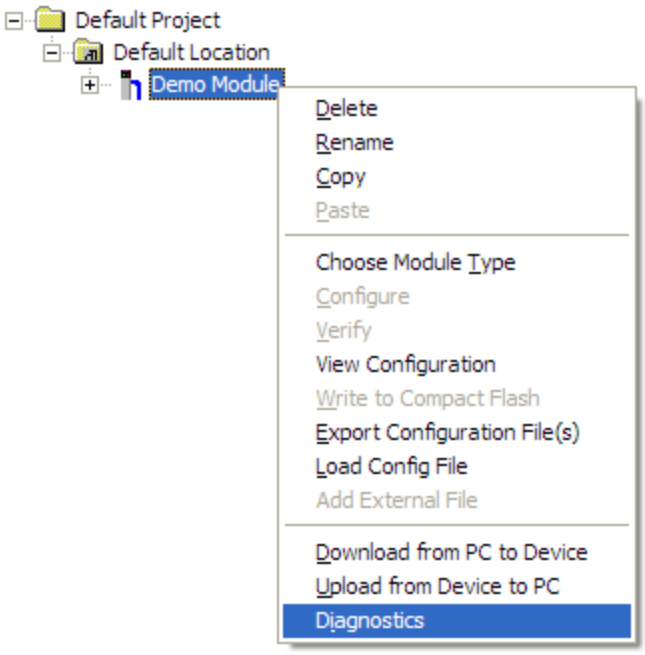
7.3.1 Using the Diagnostic Window in ProSoft Configuration Builder

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

1 Start PCB, and then select the module to test. Click the right mouse button to open a shortcut menu.



2 On the shortcut menu, choose **DIAGNOSTICS.**



3 This action opens the **DIAGNOSTICS** dialog box. Press **[?]** to open the Main Menu.

```
IEC-870-5-103 MASTER COMMUNICATION MODULE
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  I=IEC-103 Master Menu
  P=Backplane Command List
  R=Receive Configuration File
  S=Send Configuration File
  V=Version Information
  Esc=Exit Program
```

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:

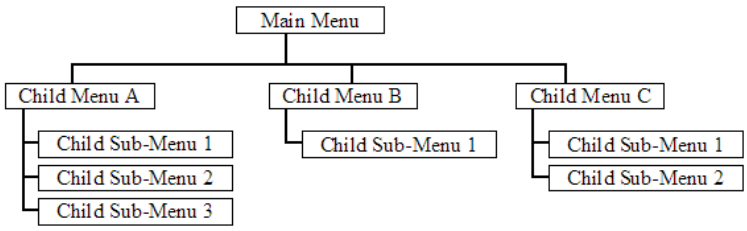
1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
2 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

7.3.2 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.

7.3.3 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 **[L]** (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.

7.3.4 Main Menu

When you first connect to the module from your 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:

```
IEC-870-5-103 MASTER COMMUNICATION MODULE
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  I=IEC-103 Master Menu
  P=Backplane Command List
  R=Receive Configuration File
  S=Send Configuration File
  V=Version Information
  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.

Viewing Block Transfer Statistics

Press **[B]** from the Main Menu 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: To determine the number of blocks transferred each second, mark the numbers displayed at a specific time. Then some seconds later activate the command again. Subtract the previous numbers from the current numbers and divide by the quantity of seconds passed between the two readings.

Viewing Module Configuration

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

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

Opening the Database Menu

Press **[D]** to open the Database View menu. Use this menu command to view the current contents of the module's database.

Opening the IEC-103 Master Menu

Press **[I]** from the Main Menu to open the IEC-870-5-103 Master Driver Menu. Use this menu command to view detailed configuration information for the module.

```
IEC-103 MASTER Menu Selected

IEC-870-5-103 MASTER DRIVER MENU
  ?=Display Menu
  A=Data Analyzer
  C=General Configuration
  I=Command List Menu
  P=Port Configuration Menu
  Q=Port Status Menu
  S=Session Menu
  V=Version
  Z=Previous Menu
```

For more information about the commands on this menu, refer to IEC-103 Master Driver Menu (page 84).

Viewing the Backplane Command List

Press **[P]** from the Main Menu to view the Backplane Data Exchange List. Use this command to display the configuration and statistics of the backplane data transfer operations.

```
BACKPLANE DATA EXCHANGE LIST -- COMMANDS 0 TO 9

TYPE  DBREG  DBTYPE  ADDRESS  COUNT  LASTERR
   0      0       0        0      0   0X0000
   0      0       0        0      0   0X0000
   0      0       0        0      0   0X0000
   0      0       0        0      0   0X0000
   0      0       0        0      0   0X0000
   0      0       0        0      0   0X0000
   0      0       0        0      0   0X0000
   0      0       0        0      0   0X0000
   0      0       0        0      0   0X0000
   0      0       0        0      0   0X0000
```

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

Receiving the Configuration File

Press **[R]** to download (receive) the current configuration file from the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File.

Sending the Configuration File

Press **[S]** to upload (send) an updated configuration file to the module. For more information on receiving and sending configuration files, please see Uploading and Downloading the Configuration File.

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.

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 memory to configure the module.

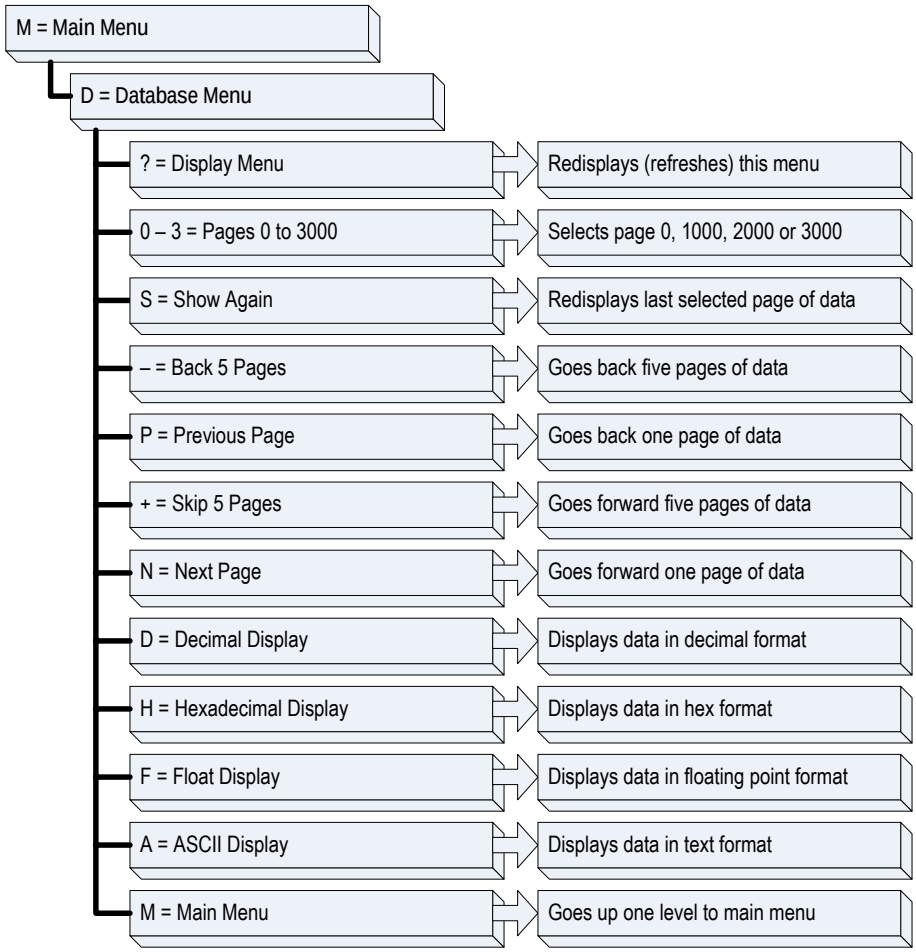
Warm Booting the Module

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.

7.3.5 Database View Menu

Press **[D]** from the Main Menu to open the Database View menu. Use this menu command to view the current contents of the module's database. Press **[?]** to view a list of commands available on this menu.



Viewing Register Pages

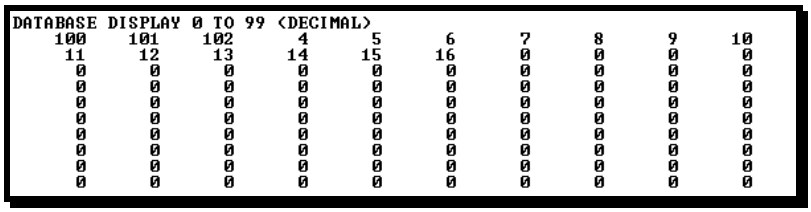
To view sets of register pages, use the keys described below:

Command	Description
[0]	Display registers 0 to 99
[1]	Display registers 1000 to 1099
[2]	Display registers 2000 to 2099

And so on. The total number of register pages available to view depends on your module's configuration.

Displaying the Current Page of Registers Again

Press **[S]** from the Database View menu to show the current page of registers again.



This screen displays the current page of 100 registers in the database.

Moving Back Through 5 Pages of Registers

Press **[-]** from the Database View menu to skip five pages back in the database to see the previous 100 registers of data.

Moving Forward Through 5 Pages of Registers

Press **[+]** from the Database View menu to skip five pages ahead in the database to see the next 100 registers of data.

Viewing the Previous 100 Registers of Data

Press **[P]** from the Database View menu to display the previous 100 registers of data.

Viewing the Next 100 Registers of Data

Press **[N]** from the Database View menu to select and display the next 100 registers of data.

Viewing Data in Decimal Format

Press **[D]** to display the data on the current page in decimal format.

Viewing Data in Hexadecimal Format

Press **[H]** to display the data on the current page in hexadecimal format.

Viewing Data in Floating Point Format

Press **[F]** from the Database View menu. Use this command to display the data on the current page in floating point format. The program assumes that the values are aligned on even register boundaries. If floating-point values are not aligned as such, they are not displayed properly.

Viewing Data in ASCII (Text) Format

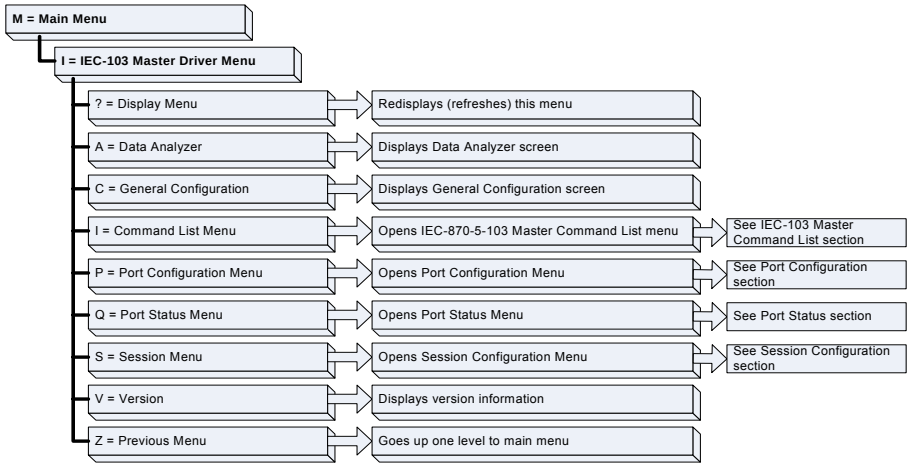
Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

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

7.3.6 IEC-103 Master Driver Menu

Press **[I]** from the Main Menu to open the IEC-870-5-103 Master Driver Menu. Use this menu command to view detailed configuration information for the module.



Opening the Data Analyzer Menu

Press **[A]** to open the Data Analyzer Menu. Use this command to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Refer to Data Analyzer for more information about this menu.

Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Viewing General Configuration

Press **[C]** from the IEC-103 Master Driver Menu to display the general configuration for the protocol. The following illustration shows an example of the Module Configuration screen:

```
MODULE CONFIGURATION
  Max Sessions    = 0 (of maximum 32)
  Online States   = 0x00000000 (bit mapped in hex format)
  Command Count   =    0 (of maximum 1000)
  Current Command =    0 (IDLE)
 MEMORY COUNTERS:
  applRec   = 0
  applTrans = 0
  anything  = 0
```

The Busy/Idle message indicates the current activity state of the module. "Idle" means it is waiting to execute a command. "Busy" means it is executing a command and is waiting for the response to the request. This does not include the normal class 1 and 2 polls as these are automatically generated.

The counter data displays the number of memory areas allocated for the application layer. When no packets are pending, the counts should all be 0. If messages are waiting to be sent, the applRec count will indicate the number waiting to be sent. If many messages are received at the same time, the applRec count will indicate the number of packets that must be processed. The "anything" count indicates any other buffer area that is allocated and must be processed by the application.

Opening the IEC-870-Master Command List Menu

Press **[I]** from the IEC-103 Master Driver Menu to open the ICE-870 Master Command List menu. Use this command to view the configured command list for the module.

```
IEC-870-5-103 MASTER COMMAND LIST, COMMANDS 0 TO 9

ENB DBIDX POLLI SES SEC ASDU FUNC POINT LAST_POLL LERROR OVR VALUE
  1     0    10   0   0   20  128    16         7      0   1     2
  0     0    11   0   0   20  128    16     14932      0   1     1
  0  1600    41   0   0   20  128    17     14932      0   0     0
  0  1602    41   0   0   20  128    18     14932      0   0     0
  1     0     5   0   0    7  255     0         4      0

**** END OF COMMAND LIST REACHED ****
```

For more information about the commands on this menu, refer to IEC-870-Master Command List Menu (page 88).

Opening the Port Configuration Menu

Press **[P]** from the IEC-103 Master Driver Menu to open the Port Configuration menu. Use this command to view the port configuration information for each of the application ports.

```
Port Configuration Menu Selected

IEC-870-5-103 MASTER CHANNEL 0 CONFIGURATION
  Baudrate          = 19200
  Parity            = EVEN
  RTS On            = 1
  RTS Off           = 0
  Mimimum Delay     = 30
  Receive Timeout   = 2000
  Single ACK        = YES
  Data Link Length  = 1
  Use Balanced Mode = NO
```

The *Port Configuration Menu* section has more information about the commands on this menu.

Opening the Port Status Menu

Press **[Q]** from the IEC-103 Master Driver Menu to open the Port Status menu. Use this command to verify the status of the master commands sent through the port. If the display indicates a communication error, you should compare the generated error code with the command error codes listed in the Appendices of this manual.

```
Port Status Menu Selected

IEC-870-5-103 MASTER PORT 0 STATUS
  Commands Executed  = 0
  Command Responses  = 0
  Command Err Count  = 0
  Request Count      = 0
  Response Count     = 0
  Error Sent Count   = 0
  Error Rec Count    = 0
  Cfg Error Word     = 0x0000
  Current Error Code = 0
  Last Error Code    = 0
```

The *Port Status Menu* section has more information about the commands on this menu.

Opening the Session Configuration Menu

Press **[S]** to open the Session Configuration menu. Use this command to view the session configuration data.

Refer to *Session Configuration Menu* (page 91) for more information about the commands on this menu.

Opening the Sector Menu

Press **[1]** from the IEC-103 Master Driver Menu to open the Sector Configuration menu. Use this command to view the Sector Configuration data.

```
SECTOR CONFIGURATION MENU
 ?=Display Menu
 D=Sector Database Menu
 S=Show Again
 P=Previous Page
 N=Next Page
 M=Return to Session Menu
```

The *Sector Configuration Menu* section has more information about the commands on this menu.

Viewing Master Driver Version Information

Press **[V]** from the IEC-103 Master Driver Menu to view the master driver version information.

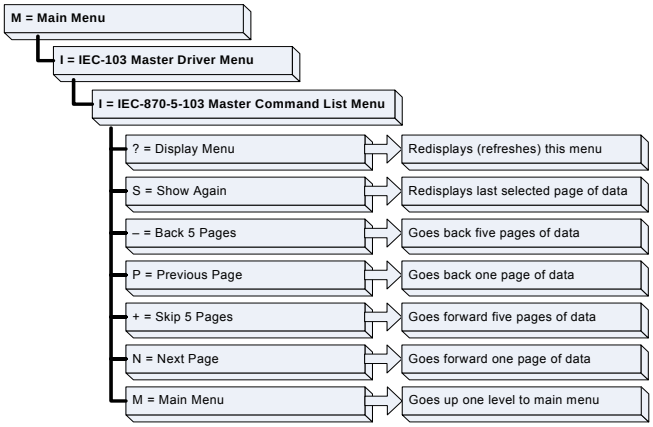
```
DRIVER VERSION INFORMATION:

  (c) 1999-2003, ProLinx Communication Gateways, Inc.

  PRODUCT NAME CODE       : IEC-870-5-103 Master Driver
  SOFTWARE REVISION LEVEL : 2.14
  SOFTWARE REVISION DATE  : 04/17/2003
  FAR CORE LEFT           : 322368
```

Returning to the Main Menu

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

7.3.7 IEC-870-Master Command List Menu

Press **[I]** from the IEC-103 Master Driver Menu to open the ICE-870 Master Command List menu. Use this command to view the configured command list for the module.



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.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Moving Back Through 5 Pages of Registers

Press **[-]** from the Database View menu to skip five pages back in the database to see the previous 100 registers of data.

Viewing the Previous 100 Registers of Data

Press **[P]** from the Database View menu to display the previous 100 registers of data.

Moving Forward Through 5 Pages of Registers

Press **[+]** from the Database View menu to skip five pages ahead in the database to see the next 100 registers of data.

Viewing the Next 100 Registers of Data

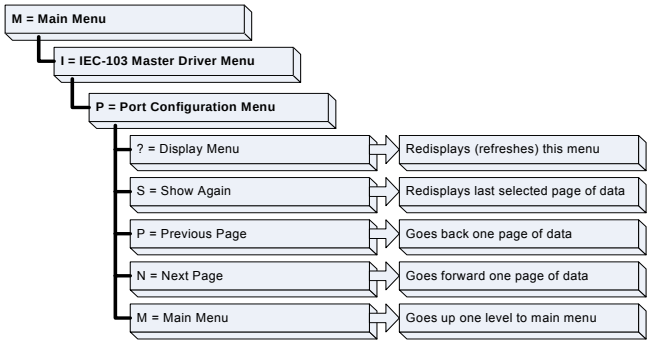
Press **[N]** from the Database View menu to select and display the next 100 registers of data.

Returning to the Main Menu

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

7.3.8 Port Configuration Menu

Press **[P]** from the IEC-103 Master Driver Menu to open the Port Configuration menu. Use this command to view the port configuration information for each of the application ports.



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.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Displaying the Next Page

Press **[N]** to display the next 100 registers. Use this command to step forward through the data a page at a time.

Displaying the Previous Page

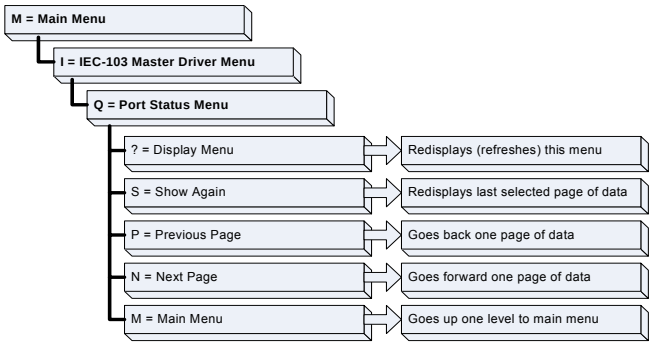
Press **[P]** to display the previous 100 registers. Use this command to step backward through the data a page at a time.

Returning to the Main Menu

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

7.3.9 Port Status Menu

Press **[Q]** from the IEC-103 Master Driver Menu to open the Port Status menu. Use this command to view the communication status information for each application port.



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.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Displaying the Previous Page

Press **[P]** to display the previous 100 registers. Use this command to step backward through the data a page at a time.

Displaying the Next Page

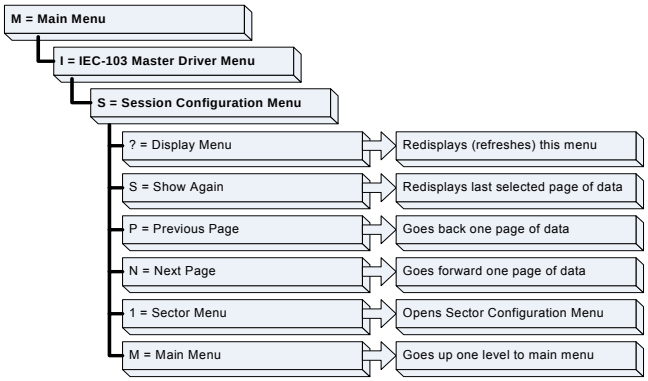
Press **[N]** to display the next 100 registers. Use this command to step forward through the data a page at a time.

Returning to the Main Menu

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

7.3.10 Session Configuration Menu

Press **[S]** from the IEC-103 Master Driver Menu to open the Session Configuration menu. Use this command to view the session configuration for each controlled device.



Online State

The Online State indicator displays 0 if the module is not online, 1 if the module is online.

Session State

The Session State indicator displays 1 if there is a configuration error, or 2 if the module is ready for communication. If the session is not in use, the Session State indicator displays 0.

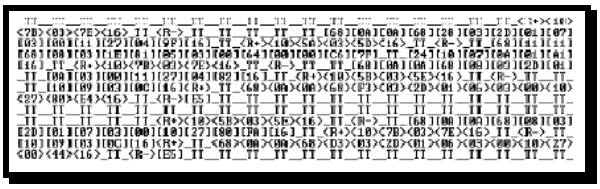
7.3.11 Data Analyzer

The data analyzer mode allows you to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Use of this feature is limited without a thorough understanding of the protocol.

Note: The Port selection commands on the Data Analyzer menu differs very slightly in different modules, but the functionality is basically the same. Use the illustration above as a general guide only. Refer to the actual data analyzer menu on your module for the specific port commands to use.
Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Analyzing Data for the first application port

Press **[1]** to display I/O data for the first application port in the Data Analyzer. The following illustration shows an example of the Data Analyzer output.



Analyzing Data for the second application port

Press **[2]** to display I/O data for the second application port in the Data Analyzer.

Displaying Timing Marks in the Data Analyzer

You can display timing marks for a variety of intervals in the data analyzer screen. These timing marks can help you determine communication-timing characteristics.

Key	Interval
[5]	1 milliseconds ticks
[6]	5 milliseconds ticks
[7]	10 milliseconds ticks
[8]	50 milliseconds ticks
[9]	100 milliseconds ticks
[0]	Turn off timing marks

Removing Timing Marks in the Data Analyzer

Press **[0]** to turn off timing marks in the Data Analyzer screen.

Viewing Data in Hexadecimal Format

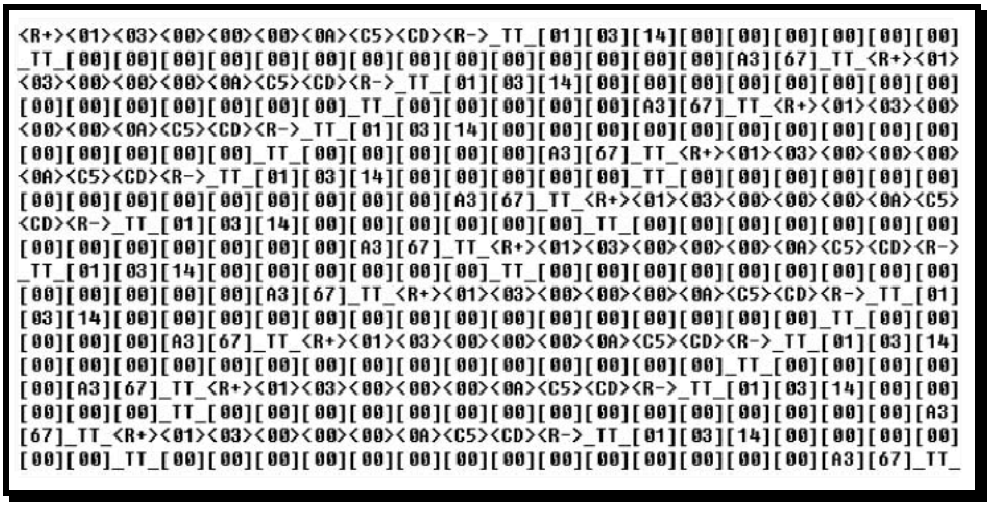
Press **[H]** to display the data on the current page in hexadecimal format.

Viewing Data in ASCII (Text) Format

Press **[A]** to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Starting the Data Analyzer

Press **[B]** to start the data analyzer. After the key is pressed, all data transmitted and received on the currently selected port will be displayed. The following illustration shows an example.



The Data Analyzer displays the following special characters:

Character	Definition
[ ]	Data enclosed in these characters represent data received on the port.
< >	Data enclosed in these characters represent data transmitted on the port.
<R+>	These characters are inserted when the RTS line is driven high on the port.
<R->	These characters are inserted when the RTS line is dropped low on the port.
<CS>	These characters are displayed when the CTS line is recognized high.
_TT_	These characters are displayed when the timing mark interval has been reached. This parameter is user defined.

Stopping the Data Analyzer

Press **[S]** to stop the data analyzer. Use this option to freeze the display so the data can be analyzed. To restart the analyzer, press **[B].**

Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Returning to the Main Menu

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

7.3.12 Data Analyzer Tips

From the main menu, press **[A]** for the "Data Analyzer". You should see the following text appear on the screen:

```
Data Analyzer Mode Selected
```

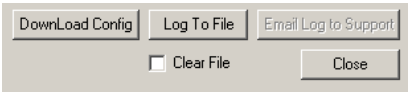
After the "Data Analyzer" mode has been selected, press **[?]** to view the Data Analyzer menu. From this menu, you can select the "Port", the "format", and the "ticks" that you can display the data in.

For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

The Tick value is a timing mark. The module will print a _TT for every xx milliseconds of no data on the line. Usually 10milliseconds is the best value to start with.

To save a capture file of your Diagnostics session

1 After you have selected the Port, Format, and Tick, we are now ready to start a capture of this data. Click the Log to File button at the bottom of the Diagnostics window.



2 When you have captured the data you want to save, click the Stop Logging button.

You have now captured, and saved the file to your PC. This file can now be used in analyzing the communications traffic on the line, and assist in determining communication errors.

Now you have everything that shows up on the HyperTerminal screen being logged to a file called ProLinxLog.txt. This is the file that you will then be able to email to ProSoft Technical Support to assist with issues on the communications network.

To begin the display of the communications data, you will then want to press **[B]** to tell the module to start printing the communications traffic out on the debug port of the module. After you have pressed **[B],** you should see something like the following:

```
[03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+><01><02>
<00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT__TT_<R+>
<01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT__TT_<R+>
<01><01><00><00><00><A0><3C><72><R->_TT__TT_[01][01][14][00][00][01][00][02]_TT_
[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]_TT__TT_<R+>
<01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+>
<01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT_
_TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01]
[00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT_
_TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00][00][01]
[00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]
_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00]
[01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]
_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT_[01][02][02][00][00][B9]
[B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00]
[00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD]
[51]_TT__TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00]
[00][01][00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00]
[B7][52]_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00]
[00][00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09]
[FB][B7]_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02]
[00][00][B9][B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_
```

The <R+> means that the module is transitioning the communications line to a transmit state.

All characters shown in <> brackets are characters being sent out by the module.

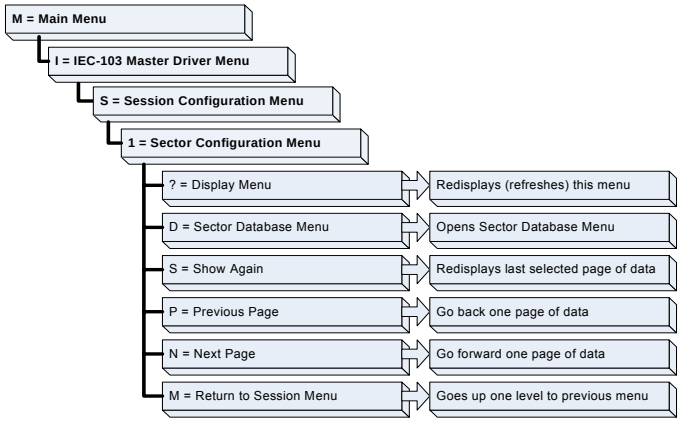
The <R-> shows when the module is done transmitting data, and is now ready to receive information back.

And finally, all characters shown in the [ ] brackets is information being received from another device by the module.

After taking a minute or two of traffic capture, you will now want to stop the "Data Analyzer". To do so, press the **[S]** key, and you will then see the scrolling of the data stop.

7.3.13 Sector Configuration Menu

Press **[1]** from the IEC-103 Master Driver Menu to open the Sector Configuration menu. Use this command to view the contents of the Sector Configuration Databases for each session (controlled device). The module supports up to three sectors (databases) per session.



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.

Opening the Sector Database Menu

Press **[D]** from the Sector Configuration menu to open the Sector Database menu. Use this command to look at the configuration and current value for each point.

The *IEC-870-Master Command List Menu* section has more information about the commands on this menu.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Displaying the Next Page

Press **[N]** to display the next 100 registers. Use this command to step forward through the data a page at a time.

Displaying the Previous Page

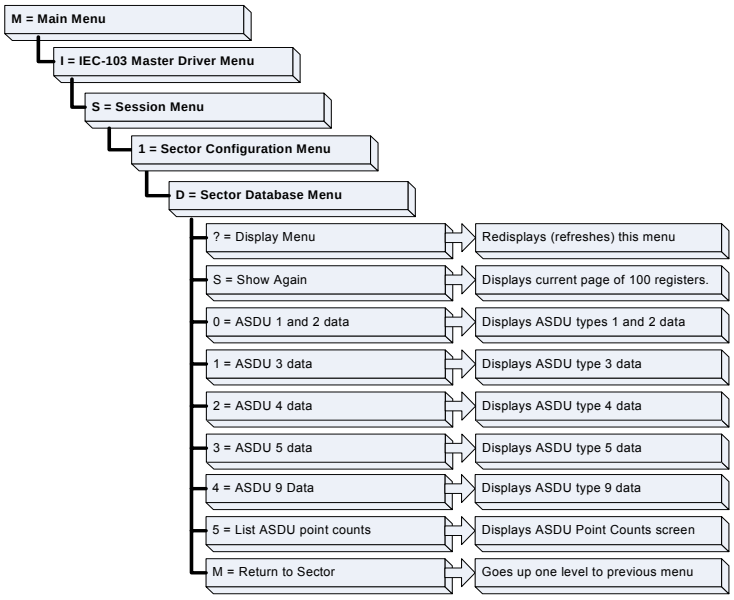
Press **[P]** to display the previous 100 registers. Use this command to step backward through the data a page at a time.

Returning to the Main Menu

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

7.3.14 Sector Database Menu

Press **[D]** from the Sector Configuration menu to open the Sector Database menu. Use this command to display the sector database values. Each session (controlled device) contains one or more data sets (sectors) that are defined by the vendor of the device.



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.

Redisplaying the Current Page

Press **[S]** to display the current page of data.

Viewing ASDU n Data

Press keys **[0]** (zero) through **[4]** to display ASDU (Application Data Service Unit) data for each of the supported data types. Refer to *[IEC-103 Master Session x Sector x]* for a list of ASDU types.

Listing ASDU point counts

Press **[5]** to display the ASDU point counts for each ASDU type.

```
SESSION 0 SECTOR 0 DATABASE SIZES:
  ASDU    COUNT
   1&2       10
     3        0
     4        0
     5        0
     9        0
```

Returning to the Main Menu

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

7.4 LED Status Indicators

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

Module	Color	Status	Indication
DEBUG	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.
PRT1	Green	On	Data is being transferred through the port
		Off	No data is being transferred through the port
PRT2	Green	On	Data is being transferred through the port
		Off	No data is being transferred through the port
CFG ERR	Red	Off	The PTQ-103M is working normally.
		On	The PTQ-103M module program has recognized an application error.
ERR2	Red	Off	The PTQ-103M is working normally.
		On	The PTQ-103M module program has recognized an application error.
ERR3	Red	Off	The PTQ-103M is working normally.
		On	The PTQ-103M module program has recognized an application 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.
BAT	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.

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

7.4.1 Error Status Table

The program maintains an error/status table that is transferred to the processor in each read block. You can use the error/status data to determine the "health" of the module. Refer to Error Status Table for data block structure (page 144).

8 Reference

In This Chapter

8.1 Product Specifications

The PTQ IEC 60870-5-103 Master Communication Module allows Quantum backplane I/O compatible processors to interface easily with IEC 60870-5-103 compatible devices. Devices commonly supporting the protocol include relays, breakers, sub-station Communication Modules and other serial communication devices used in power monitoring.

The PTQ-103M supports up to 32 total field devices between the module's two IEC 60870-5-103 Master ports. The module's communication ports can each be independently configured, allowing two separate field networks to be implemented. The field device data is exchanged between the PTQ module and the Quantum processor over the backplane.

8.1.1 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: 0x, 1x, 3x, 4x
- High speed data transfer across backplane provides quick data update times
- Does not currently support Hot-Standby processors or applications

8.1.2 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
	Serial Activity and Error Status
Configuration Serial Port (Debug)	DB-9M PC Compatible
	RS-232 only
	No hardware handshaking
Application Serial Ports (PRT1, PRT2)	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

8.1.3 Functional Specifications

The PTQ-103M module supports the IEC 60870-5-103 protocol to the following specifications:

- The IEC 60870-5-103 communication driver is built in accordance to the approved IEC specification
- User-definable module memory usage
- The module has two independent master ports, each configurable via a simple configuration file
- Supports up to 32 sessions (controlled devices) between the two ports
- Supports up to five sectors (separate databases) for each session, with individual database definition for each sector
- Total of 1000 user configurable commands to control data transfer to/from devices (controlled devices)
- Supports clock synchronization from/to the processor
- Event data received from the Control Devices updates the module database (Date and Time stamping is not stored or used by module)
- Class 1 and Class 2 delay parameter are configurable for each session
- An IEC Interoperability Document for the module is available from the web site, which fully documents data types supported by the module

8.2 Functional Overview

This section describes how the PTQ-103M module transfers data between itself and the processor, and how it implements the 103 protocol.

The standards used to build the module are listed in the following table.

Publication	Title
IEC 60870-5-103	Companion Standard for the informative interface of protection equipment.
IEC 60870-5-103 Annex A	Generic functions --Examples of constructing a directory
IEC 60870-5-1	Transmission Frame Formats
IEC 60870-5-2	Link Transmission Procedures
IEC 60870-5-3	General Structure of Application Data
IEC 60870-5-4	Definition and Coding of Application Information Elements
IEC 60870-5-5	Basic Application Functions

These documents should be obtained, reviewed, and understood in order to fully appreciate the protocol implementation. Most of the complexity of the protocol is hidden from the user and simplified in the application of the module. Detailed questions of about the protocol can be answered by reading these documents. In addition to calling our technical support group, there is also help available for the protocol using the following mail list Web Site: www.TriangleMicroWorks.com/iec870-5 (http://www.trianglemicroworks.com/iec870-5). Go to this site to join the mail list and to review questions and answers from mail list users.

8.2.1 Backplane Data Transfer

Before modifying the **[BACKPLANE DATA EXCHANGE]** section of the configuration file, familiarize yourself with the following concepts.

If you have used the parameters defined in the **[BACKPLANE CONFIGURATION]** section (page 61), you have created the following memory map. We will use this map to explain how data transfer works between the processor and the ProTalk module.

PTQ Memory Address		Application Memory Address
0	=	0
10	=	10
20	=	20
30	=	30
40	=	40
50	=	50
…		…
206	=	206
207	=	207
…	=	…
312	=	312

This example shows an application that reads 207 words from the Quantum to the module and writes 105 words from the module to the Quantum.

The module uses Command Functions to transfer data over the Quantum backplane between the module and the processor. The command functions reside in the **[BACKPLANE DATA EXCHANGE]** section of the configuration file. This method of data transfer offers some unique advantages:

- The amount of ladder logic required will be substantially reduced; in fact in many applications no ladder logic will be required.
- The module may be placed in any position in the chassis containing the PLC, and will operate with no modifications.

The **[BACKPLANE DATA EXCHANGE]** section is a series of messages that instruct the module how to transfer data to/from the Quantum. What is missing from the message is the ability to schedule its execution. This ability is normally left to the programmer in the PLC environment, however in the PTQ module this is not included so that the commands may run as fast as possible to maintain the synchronization of the two databases.

One command from the list will execute during each I/O service interval at the end of the PLC ladder logic evaluation.

For example, if your configuration contains 10 "Backplane Data Exchange" commands, it will require 10 PLC scans to process the entire list.

This section may contain up to 100 individual commands used in any combination to transfer data to/from the Quantum.

The following topics describe the use of the commands, with examples.

Command Function 1 (Read data from the Quantum)

This section provides information on how to read data from the Quantum into the module.

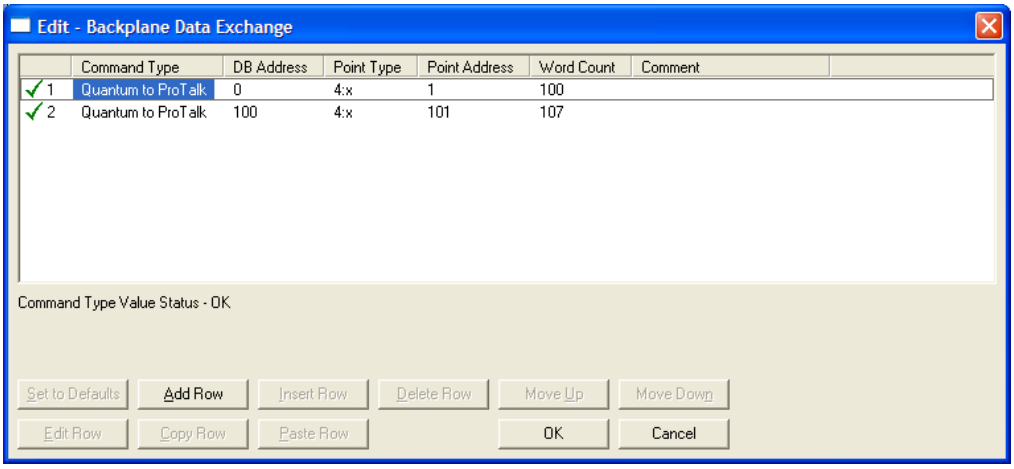
Command Function 1 (one) is designed to transfer data from the Quantum to the module on a continuous basis. The command(s) required to transfer your application data should be entered in the **[BACKPLANE DATA EXCHANGE]** section of your configuration file as required.

- This command takes the following parameters:
- **Command type:** 1 (Read data from the Quantum)
- **PTQ Database Address:** The destination for the data retrieved from the Quantum.
- **Point Type:** The type of register within the Quantum (0:x = 0, 1:x = 1 3:x = 3 or 4:x = 4)
- **Quantum Address:** The source of the data within the Quantum. The address is expressed without the use of the register range, for example 400001 would be entered as 1 (400001 - 400000 = 1 or 40001 - 40000 = 1)
- **Word Count:** The number of words to copy. The length of this copy may be any length of 1 to 130 inclusive. If your application requires the movement of additional data you may enter additional commands.

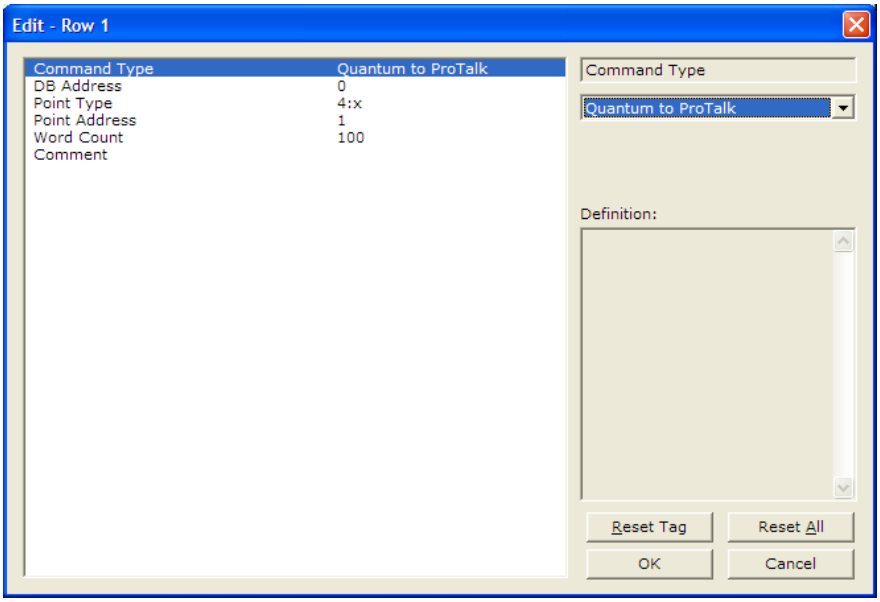
Defining Data to Send to the PTQ Database

You might be asked to provide access to 207 words of information for other devices on the network. This information resides in the PLC at addresses 400001 to 400207 and you must place this data in the first 207 words of the module database. This requires the use of Command **Function 1**.

Because the total amount of data exceeds the maximum length of any single command function, you will need two entries in the [**BACKPLANE DATA EXCHANGE**] section of your configuration. The following illustrations contain examples of how to configure the necessary commands.

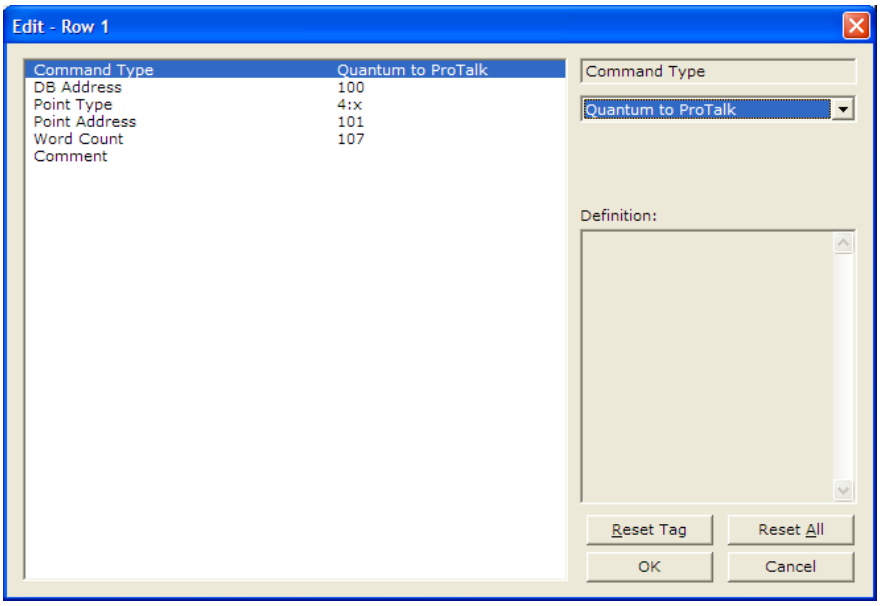


The first command states:



Field	Value	Meaning
Command Type	1	The type of operation to perform 1 = Read data from the Quantum into the PTQ
DB Address	0	The destination address within the PTQ
Point Type	4	The range of registers to read from the Quantum 4 = 4x style register
Point Address	1	The starting address of the data within the Quantum This would be Point Type + offset Example: 40000 + 1 = 40001
Word Count	100	The number of registers to transfer

The second command states:



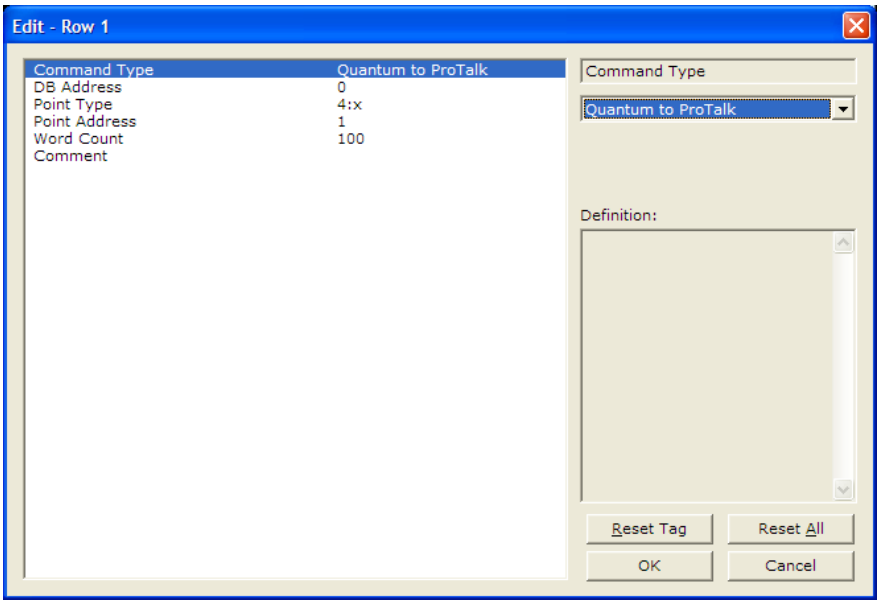
Field	Value	Meaning
Command Type	1	The type of operation to perform 1 = Read data from the Quantum into the PTQ
DB Address	100	The destination address within the PTQ
Point Type	4	The range of registers to read from the Quantum 4 = 4x style register
Point Address	101	The starting address of the data within the Quantum This would be Point Type + Quantum Address Example: 40000 + 101 = 40101
Word Count	107	The number of registers to transfer

The following diagram shows the result of this example.

Quantum Memory Address		PTQ Memory Address
40001		0
40010	First Command	9
40020		19
40030		29
40040		39
…		
40060	→	59
40070		69
40080		79
40090		89
40100		99
40101		100
40111	Second Command	110
…	→	
40200		199
40207		206
…		…

Example 3x or 4x Register Transfer

The following example shows a typical command used to read 4x data from the Quantum. In this example, registers 400001 to 400099 from the Quantum will be transferred to registers 0 to 99 within the module.

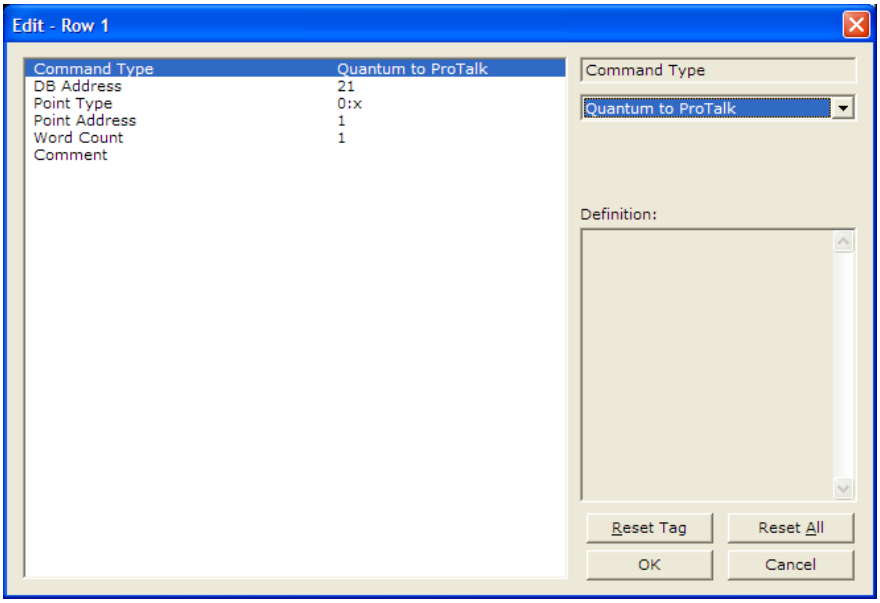


Example 0:x or 1x Register Transfer

The transfer of Coils and Input bits require some forethought as the command transfers **words** and not bits. This means that if you want to transfer bits 000005 to 000007 from the Quantum to word 21 in the module you would have to transfer the word within the Quantum containing bits 000001 to 000016 to a word within the modules memory.

Take care with the transfer of bits while planning the application, so as to optimize the usage of the available bits, and to preserve the integrity of your information.

The following illustration shows how this could be accomplished.



Command Function 2 (Write data to the Quantum)

This section provides information on how to write data from the module to the Quantum.

Command Function 2 (two) is designed to transfer data from the module to the Quantum on a continuous basis. The command(s) required to transfer your application data should be entered in the **[BACKPLANE DATA EXCHANGE]** section of your configuration as required (page 62).

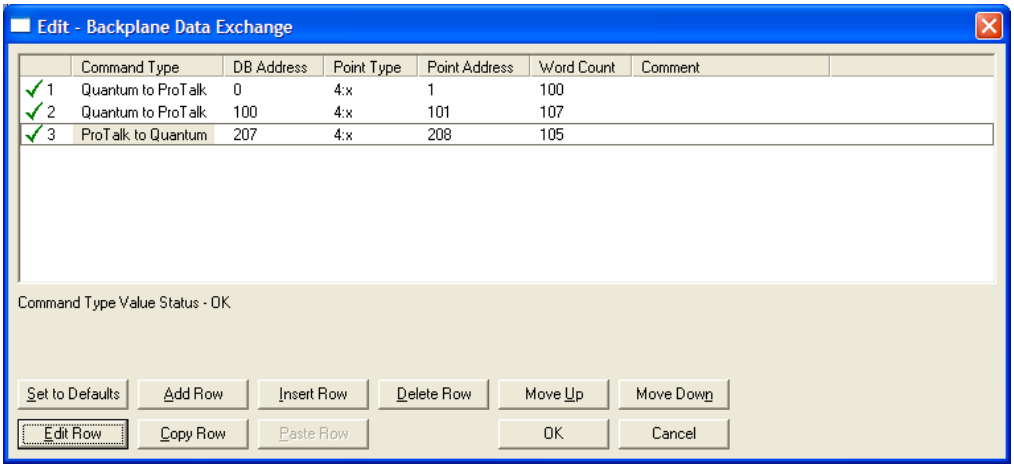
- This command takes the following parameters:
- **Command type:** 2 (Write data to the Quantum)
- **PTQ Database Address:** The source of the data within the PTQ to be sent to the Quantum.
- **Point Type:** The type of register within the Quantum (0:x = 0, 1x = 1 3x = 3 or 4x = 4)

- **Quantum Address:** The destination register within the Quantum. The address is expressed without the use of the register range, for example 400001 would be entered as 1 (400001 - 400000 = 1 or 40001 - 40000 = 1)
- **Word Count:** The number of words to copy. The length of this copy may be any length of 1 to 130 inclusive. If your application requires the movement of additional data you may enter additional commands.

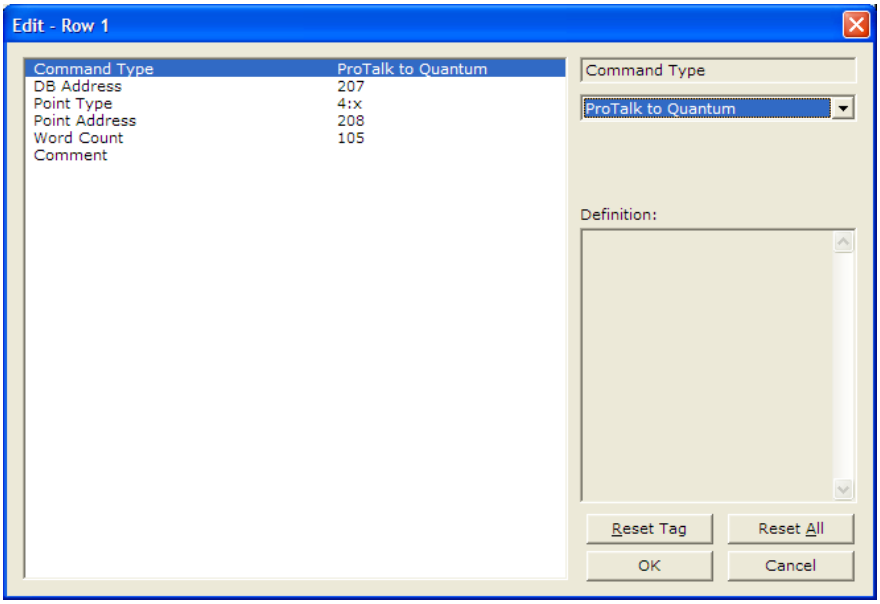
Defining Data to Retrieve from the PTQ Database.

Your application may need to retrieve 105 words of data from other devices on the network. These devices have either sent you the data if you are a slave, or you have obtained it for your use if you happen to be a master in your application. Assuming that the data resides in registers 207 to 312 within the PTQ modules memory, and you wish to place this information in addresses 400208 to 400313 within the Quantum, you could use **Command Function 2** to transfer the information.

Because the total amount of data does not exceed 130 words in length, a single command can be used to transfer the data. This additional entry will be added to our **[BACKPLANE DATA EXCHANGE]** section and it would look like the third command below:



The third command states:



Field	Value	Meaning
Command Type	2	The type of operation to perform 2 = Write data from the PTQ to the Quantum
DB Address	207	The destination address within the PTQ
Point Type	4	The range of registers to read from the Quantum 4 = 4x style register
Point Address	208	The starting address of the data within the Quantum This would be Point Type + Quantum Address Example: 40000 + 207 = 40207
Word Count	105	The number of registers to transfer

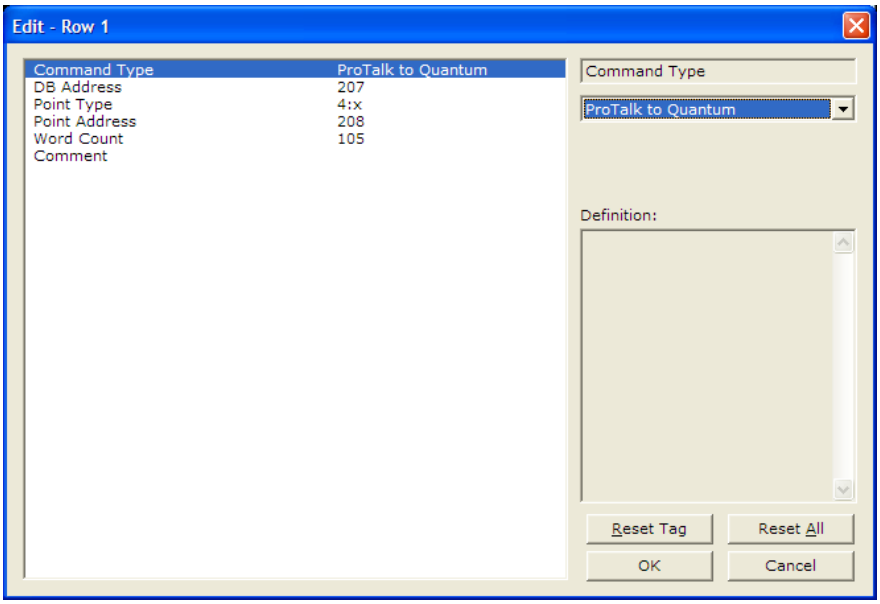
The following diagram shows the result of this example:

Quantum Memory Address		PTQ Memory Address
40001		0
40010		9
40020		19
40030		29
40040		39
…	First Command	
40060	→	59
40070		69
40080		79
40090		89
40100		99
40101		100

Quantum Memory Address		PTQ Memory Address
40111	Second Command	110
…	→	
40200		199
40207		206
…		…
40208		207
40210	Third Command	209
...	←	
40310		
40313		312

Example 3x or 4x Register Transfer

The following example shows a typical command used to write to the 4x registers within the Quantum. In this example, registers 207 to 312 from the PTQ will be transferred to registers 400208 to 400313 within the Quantum.

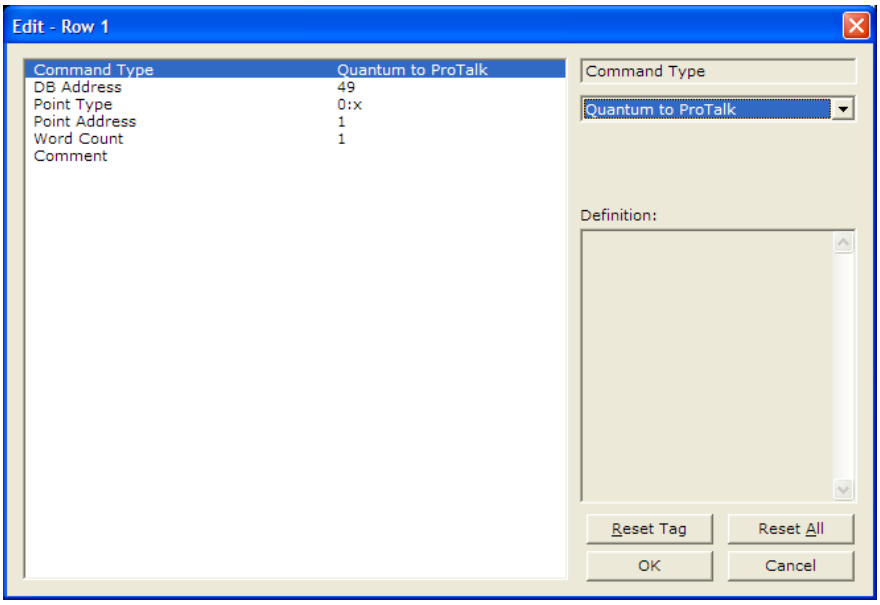


Example 0:x or 1x Register Transfer

The transfer of Coils and Inputs require some forethought as the command transfers words and not bits. This means that if you wanted to transfer the word containing the bits 805 to 806 from the module to the Quantum you would transfer the entire 50th word of the modules memory into the destination register in the Quantum. The following command transfer bits 800 to 815 (Word x Bits = Bit Address or 50 * 16 = 800) from the modules memory to word 1 of the coils (000001 to 000016) within the Quantum.

Take care with the transfer of bits while planning the application, so as to optimize the usage of the available bits, and to preserve the integrity of your information.

The following illustration shows how this could be accomplished.



Command Function 3 (Special Functions)

This section provides information on how to request the module to perform **special non-typical** functions that may be required by an application.

Command Function 3, if required, should be the first item entered in the **[BACKPLANE DATA EXCHANGE]** section of your configuration file.

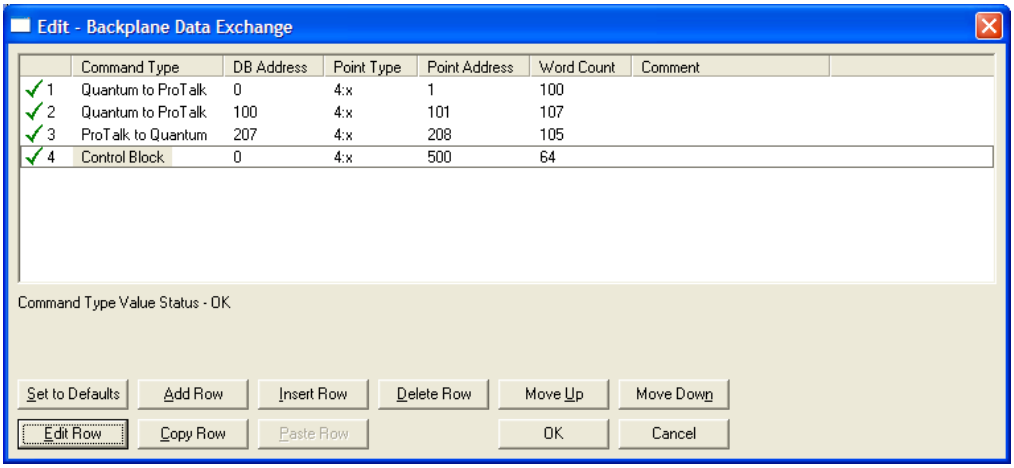
- Use Command Function 3 to send the following commands:
- Force a reboot of the PTQ module (Special Function 9998 or 9999 available on all products)
- Set / Retrieve Time and Date
- Register events with the protocol

This command takes the following parameters:

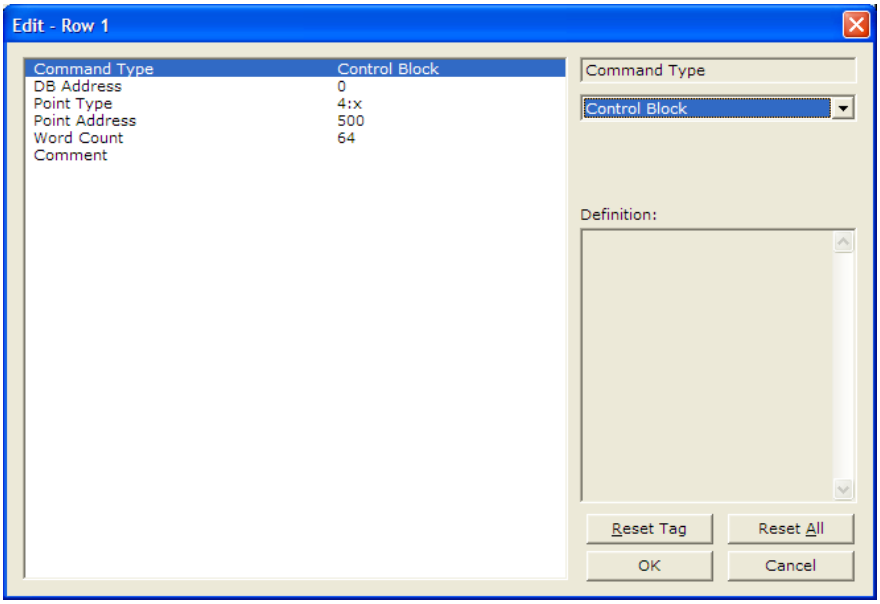
- **Command type:** 3 (Write data to the Quantum)
- **PTQ Database Address:** This value is **ALWAYS** 0. Note: This will **NOT** overwrite your application database in the PTQ but merely serves as an additional flag to notify the module of the unique nature of the command.
- **Point Type:** The type of register within the Quantum (0:x = 0, 1x = 1 3x = 3 or 4x = 4)
- **Quantum Address:** The source register within the Quantum. The address is expressed without the use of the register range, for example 400001 would be entered as 1 (400001 - 400000 = 1 or 40001 - 40000 = 1)
- **Word Count:** This value is **ALWAYS** 64. Take care to verify that 64 words of memory are available within the Quantum.

Your application may perform special functions, such as setting or retrieving the time and date, or issuing an event command to the module.

Assuming that you have chosen registers 400500 to 400563 as the target for your Command Function 3, you could enter the following command into the Backplane Data Exchange section of your configuration file.



The fourth command states:



Field	Value	Meaning
Command Type	3	The type of operation to perform 3 = Read/Write special function to the Quantum.
DB Address	0	This is ALWAYS 0 and will not overwrite your database.
Point Type	4	The range of registers to read from the Quantum 4 = 4x style register

Field	Value	Meaning
Point Address	500	The starting address of the data within the Quantum This would be Point Type + Quantum Address Example: 40000 + 500 = 40500
Word Count	64	This is ALWAYS 64 words in length.

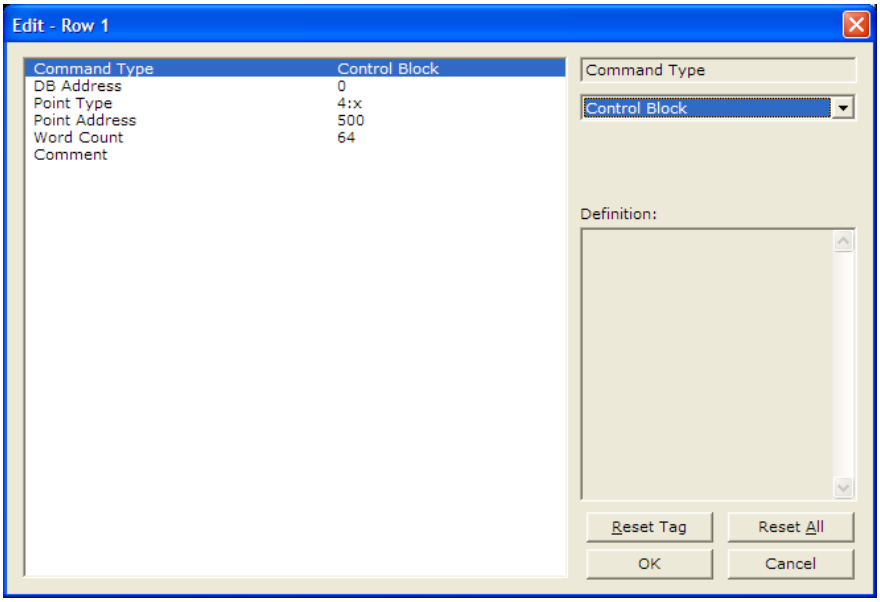
Note: This command requires two PLC scans to complete. When you issue a Function 3 we will examine the "Quantum Address" registers, process the information, clear the registers and post the status if applicable.

The following diagram shows this example:

Quantum Memory Address		PTQ Memory Address
40001		0
40010	First Command	9
40020		19
40030		29
40040		39
…		
40060	→	59
40070		69
40080		79
40090		89
40100		99
40101		100
40111	Second Command	110
…	→	
40200		199
40207		206
…		…
40208		207
40210	Third Command	209
…	←	…
40310		309
40313		312
40500		N/A
40510	Fourth Command	N/A
40520	1st Scan	N/A
40530	→	N/A
40540	2nd Scan	N/A
40550	←	N/A
40560		N/A
40563		N/A

Example 3x or 4x Register Transfer

The following example shows a typical command used to retrieve a special function command from the Quantum. In this example, registers 400500 to 400563 from the Quantum will be used to provide the information required by the module.



The following section shows the functions that may be performed by using Command Function 3.

User Constructed Command Block (9901)

Block identification code 9901 issues one or more user constructed commands. When the module receives a block 9901 identification code, it will place the included commands into the command queue.

Word Offset in Block	Data Fields	Description
0	Block ID	This field contains the block identification code of 9901 for the block.
1	Command Count	This field defines the number of user commands contained in the block. The valid range for the field is 1 to 6.
2 to 11	Command #1	Data required to build the user defined command in the command queue.
12 to 21	Command #2	Data required to build the user defined command in the command queue.
22 to 31	Command #3	Data required to build the user defined command in the command queue.
32 to 41	Command #4	Data required to build the user defined command in the command queue.
42 to 51	Command #5	Data required to build the user defined command in the command queue.

Word Offset in Block	Data Fields	Description
52 to 61	Command #6	Data required to build the user defined command in the command queue.
62 to 63	Spare	Not Used.

Each 10-word record in the command list contains the following fields.

Word Offset	Definitions	Description
0	Database Index	Address in module to associate with the command.
1	Session Index	Session index defined in the module to associate with the command.
2	Sector Index	Sector index for session as defined in the module.
3	Data Type	ASDU data type associated with the command.
4	Function Code	Function Code for the command.
5	Point Index (Information Number)	Information object address for the point on which the command operates.
6	Override Flag	Override flag for general command.
7	Override Value	Override value for general command.
8	Reserved	Reserved for future use.
9	Reserved	Reserved for future use.

Refer to [IEC-103 Master Commands] (page 63) for a detailed definition of the fields in this block. They are the same as those used in constructed the commands in the command list.

There is no response block built by the module to send back to the processor after the block is processed. The commands are placed in the command queue and issued at a high priority.

Command Control Block (9902)

Block 9902 is used by the processor to send a list of commands to be placed in the command queue from the user configured command list. Commands placed in the queue with this method need not have their enable bit set in the command list.

Word Offset in Block	Data Fields	Description
0	Block ID	This field contains the block identification code of 9902 for the block.
1	Command Count	This field defines the number of user commands contained in the block. The value has a range of 1 to 60.
2 to 61	Command Numbers to enable	These 60 words of data contain the command numbers in the command list to enable. The commands in the list will be placed in the command queue for immediate processing by the module. The first command in the list has an index of 0.
62 to 63	Spare	Not Used.

There is no response to this block by the module. The module will place the selected commands into the command queue. If the command references a unit that is not defined, the command will not be placed in the command queue. Normal processing of the command list will continue after the commands specified in this block are processed.

For digital output control, the use of block 9901 and 9902 is preferred to the use of the command list. The exact state of the output can be specified in the command list and then the command can be enabled through the use of block 9902. When the user wishes to execute this command (knowing the state of the command), can enable the command with the block 9902 request.

The following tables describe the format of each 10-word data region in the block.

Command List Error Data Block (9950)

Block 9950 identification code requests the Command List Error Table from the module for the 1000 user configurable commands. The format for the block is shown below:

Word Offset in Block	Data Fields	Description
0	Block ID	This field contains the block identification code of 9950 for the block.
1	Number of Commands to Support	This field defines the number of user commands contained in the block. The value has a range of 1 to 60.
2	Start Index of First Command	This parameter sets the index in the command list where to start. The first command in the list has a value of 0. The last index in the list has a value of MaxCommands -1
3 to 63	Spare	Not Used.

The module will respond to a valid request with a block containing the requested error information.

The following table describes the format of this block.

Word Offset in Block	Data Fields	Description
0	Reserved	Reserved (0)
1	Block ID	This is the next block requested by the module.
2	Number of Commands reported	This field contains the number of commands contained in the block that must be processed by the PLC. This field will have a value of 1 to 60.
3	Start Index of First Command	This field contains the index in the command list for the first value in the file. This field will have a value of 0 to MaxCommands -1
4 to 63	Command List Errors	Each word of this area contains the last error value recorded for the command. The command index of the first value (offset 4) is specified in word 3 of the block. The number of valid command errors in the block is set in word 2 of the block. Refer to the command error list to interpret the error codes reported.
204 to 248	Spare	Not Used
249	Block ID	This field contains the value of 9950 identifying the block type to the PLC.

Get Module Time Block (9970)

Block 9970 identification code requests the module's date and time.

Word Offset in Block	Data Fields	Description
0	Block ID	This field contains the value of 9970 identifying the block type to the module.
1 to 63	Not Used	Not Used.

The module will respond to a valid block 9970 request with a block containing the requested date and time.

The following table describes the format of this block.

Word Offset in Block	Data Fields	Description
0	Reserved	0
1	Block Write ID	9970
2	Year	This field contains the 4 digit year to be used with the new time value.
3	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
4	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
5	Hours	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
6	Minutes	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
7	Seconds	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
8	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
9 to 63	Not Used.	Not Used.

Set Module Time Block (9971)

Block identification code 9971 passes the clock time from the Quantum processor to the module. The date and time provided will be used to set the module's clock.

Word Offset in Block	Data Fields	Description
0	Block ID	This field contains the block identification code of 9971 for the block.
1	Year	This field contains the 4 digit year to be used with the new time value.
2	Month	This field contains the month value for the new time. Valid entry for this field is in the range of 1 to 12.
3	Day	This field contains the day value for the new time. Valid entry for this field is in the range of 1 to 31.
4	Hours	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
5	Minutes	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.

Word Offset in Block	Data Fields	Description
6	Seconds	This field contains the minute value for the new time. Valid entry for this field is in the range of 0 to 59.
7	Milliseconds	This field contains the millisecond value for the new time. Valid entry for this field is in the range of 0 to 999.
8 to 63	Not Used.	Not Used.

Warm Boot Block (9998)

Block 9998 performs a warm-boot operation on the module. The format of the block constructed by the processor is as follows:

Offset	Description	Length
0	9998	1
1 to 63	Spare	63

In this version of the module, the warm and cold boot processes perform the same operation as many of the variables that must be initialized are fixed when the module first boots and cannot be changed after the application starts.

Cold Boot Block (9999)

Block 9999 performs a cold-boot operation on the module. The following table describes the format of the block constructed by the processor.

Offset	Description	Length
0	9990	1
1 to 63	Spare	63

In this version of the module, the warm and cold boot processes perform the same operation, because many of the variables that must be initialized are fixed when the module first boots and cannot be changed after the application starts.

Implementing Processor's Program to Support Special Functions.

The previous discussions about Command Function 1 and Command Function 2 have not required that you implement any form of logic within the PLC, however if your application requires you to use Command Function 3, you must implement some form of control logic. The following example uses structured text language to illustrate how a typical function might be implemented.

```
Example: Rebooting the module (All modules)
(*
MyTrigger is an alias for register 401000
MyFunction3 is an alias for register 400500
MyData1-MyData63 are aliases for 400501-400563
The premise for this logic is:
IF MyTrigger = SOMEVALUE THEN
   Fill the buffer;
   set MyFunction3 to the appropriate value;
   Clear MyTrigger with a 0;
END_IF;
*)
IF MyTrigger = 9999 THEN
   MyFunction3 := MyTrigger;
   MyTrigger := 0;
END_IF;
```

Example: Setting / Retrieving the time of day

```
(*
Block ID 9971 - Set Modules Time using the PLC's Time
Assumption:
The MyYear, MyMonth and so on. values for time and date represent aliases for
your time source.
MyTrigger is an alias for register 401000.
*)
IF MyTrigger = 9971 THEN;
   MyData1 := MyYear;
   MyData2 := MyMonth;
   MyData3 := MyDay;
   MyData4 := MyHour;
   MyData5 := MyMinute;
   MyData6 := MySeconds;
   MyData7 := MyMillisec;
   MyFunction3 := 9971;
   MyTrigger := 0;
END_IF;
(*
Block ID 9970 - Set PLC's time using the modules time
Assumption:
The MyYear, MyMonth and so on. values for time and date are representative of
your aliases for your time source.
MyTrigger is an alias for register 400010.
*)
IF MyTrigger = 9970 THEN;
   MyFunction3 := MyTrigger;
   IF MyFunction3 = 0 AND MyData1 = 9970 THEN;
          MyYear := MyData2;
          MyMonth := MyData3;
          MyDay := MyData4;
          MyHour := MyData5;
          MyMinute := MyData6;
          MySeconds := MyData7;
          MyTrigger := 0;
   END_IF;
END_IF;
```

Both of these examples use structured text for the process control logic, but follow the same basic program flow.

1. Copy the data related to the block function into registers 400501 to 400563 as required.
2. Copy the BLOCK ID number of the special function into register 400500.
3. Clear your permissive condition.

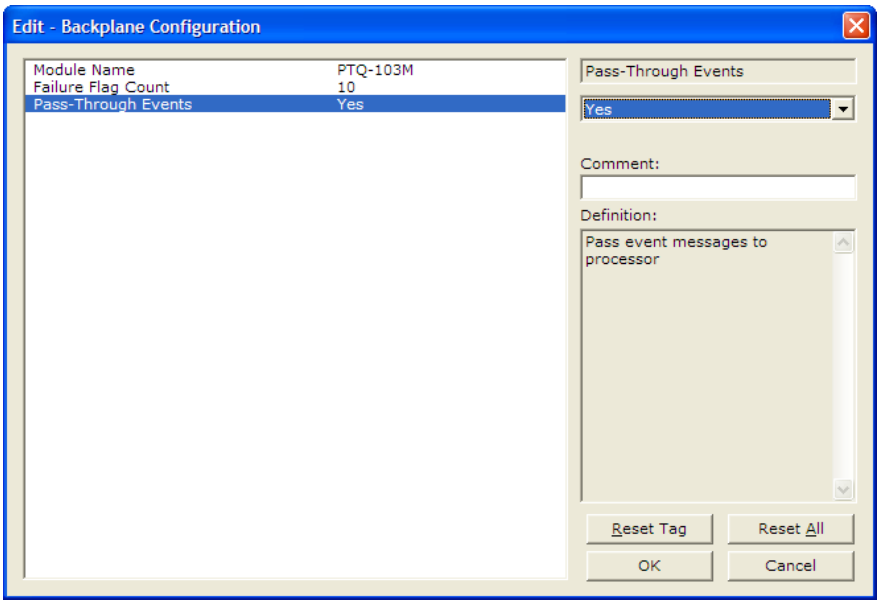
The module will read the data in and either clear the registers in the array, or return the requested data and clear the **Block ID** register (400500 in this example).

Command Function 9903 (Event Pass-Through)

The event pass-through functionality allows the module to pass events to the processor after receiving them from the IEC-8707-5-103 slave devices. Events are messages associated to supported ASDU types that contain timestamps (Hour:Minute:Seconds:Milliseconds).
Note: The event pass-through functionality is only available for version 1.12 or later.
To verify the firmware revision of your module, select the "V" key from the main menu and look for the SOFTWARE REVISION LEVEL value. If your module does not have version 1.12 installed please contact the ProSoft Technology tech support team for information on how to upgrade your module.

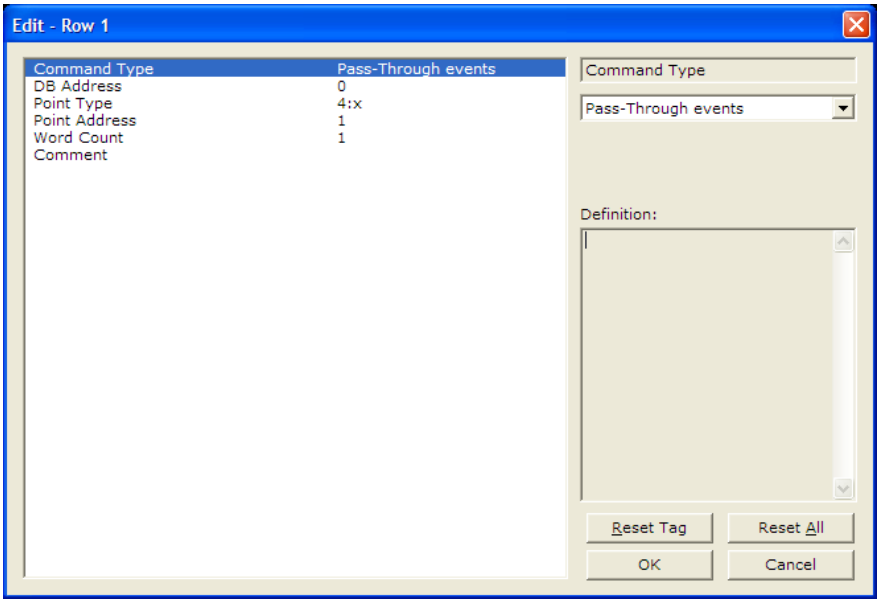
```
VERSION INFORMATION:

  IEC-870-5-103 MASTER COMMUNICATION MODULE (PTQ-I103M)
  (c) 1999-2005, ProLinx Communication Gateways, Inc.

  PRODUCT NAME CODE          : I3MQ
  SOFTWARE REVISION LEVEL    : 1.12
  OPERATING SYSTEM REVISION  : 0505
  RUN NUMBER                 : 1201
  PROGRAM SCAN COUNTER       : 6182
  FREE MEMORY                : 308048

  BACKPLANE DRIVER VERSION   : 1.04
  BACKPLANE API VERSION      : 1.11
  MODULE NAME : ProTalk Multi-Vendor Interface
  VENDOR ID   : 309       DEVICE TYPE    : 12
  PRODUCT CODE: 87        SERIAL NUMBER : 00000419
  REVISION    : 1.01
```

To use event pass-through, you must enable it in the **[BACKPLANE CONFIGURATION]** section in ProSoft Configuration Builder.



Next, configure a backplane data exchange for block 9903 (event pass-through block). Here you will configure the Quantum memory location where the block will be copied after the module receives an event.

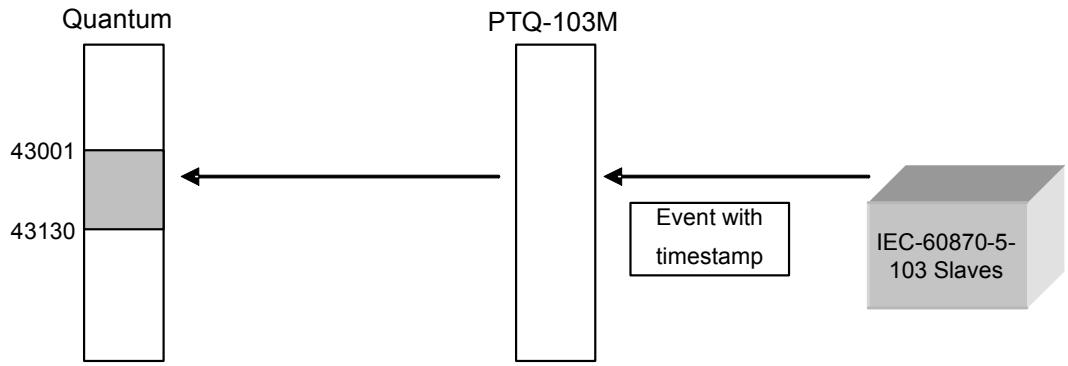
The following illustration shows an example of a pass-through command.



Where:

- **Cmd Type:** Pass-through events
- **DB Address:** this is the database address in the module associated to this block. Because the block will be handled in processor logic (not in the module) this value will not generally be used. Just make sure that this range is located in a database area that is not being used for any other purpose (130 words total)
- **Point Type:** always select 4 (holding register) for event pass-through
- **Point Address:** this is the starting memory address where the event pass-through block will be copied to (according to the selected point type). For this procedure we will be considering a value of 43001.
- **Word Count:** always use a value of 130 for the event pass-through block. Therefore for this example all registers between 43001 and 43130 will be reserved for the event pass-through block. Make sure that no other processor applications are overwriting to this area.

The following illustration shows the basic idea of the event pass-through functionality. After the module receives the event from the remote device it will build block 9903, which will be copied to the processor at the configured memory address.



Event Pass-Through Block Format

Each block can contain up to 12 events. The number of events per block will typically depend on the rate between how fast the module receives the events and how fast these can be passed to the processor (typically depends on the processor scan rate).

The following table describes the format of the block that is copied from the module to the processor.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Event Count	This field contains the number of events present in the block. Values of 1 to 20 are valid.
1 to 10	Event 1	Event message
11 to 20	Event 2	Event message
21 to 30	Event 3	Event message
31 to 40	Event 4	Event message
41 to 50	Event 5	Event message
51 to 60	Event 6	Event message
61 to 70	Event 7	Event message
71 to 80	Event 8	Event message
81 to 90	Event 9	Event message
91 to 100	Event 10	Event message
101 to 110	Event 11	Event message
111 to 120	Event 12	Event message

The following table describes the format of each 10 word data region in the block.

Word Offset	Definitions	Description
0	Session Index/ Sector Index	This field contains the session and sector indices used to define the controlled unit in the module from which the event was generated. The MSB contains the session index and the LSB contains the sector index.
1	ASDU Type	This field contains the ASDU type code for the data contained in the message.
2	Function Code/ Point Index	This field contains the function code and point index for the associated with the event message. The MSB contains the function code and the LSB contains the point index.
3	Fault Number	This is the fault number for the event if applicable. Only valid for ASDU types 2 and 4.
4	Sec/mSec	This word contains the seconds and milliseconds values with a range of 0 to 59999 time at which the message was generated by the slave device.
5	Hr/Min	This word contains the hour and minutes the message was generated by the slave. The MSB contains the hour and the LSB contains the minute value.
6	Invalid/DST	This word contains two bits that relate to the time value recorded in the slave device for the message. Bit 0 corresponds to the validity of the time (0=valid, 1=invalid) and Bit 1 defines if daylight savings time is used in the time (0=no, 1=yes).
7	Relative Time	This field contains the relative time value if applicable to the object. Only valid for ASDU types 2 and 4.
8 to 9	Value	This double-word value contains the value for the point index/function code in the event message. For ASDU types 1 and 2, this value is only 2-bits wide. For ASDU type 4, this double-word value contains the floating-point number (short-circuit location).

The processor logic should recognize an event count value greater than zero, and then read all events in the block and reset the event count value to zero to prepare for the next incoming block.

How to Set up and Use the Sample Function Block for Concept

The EVENTFB sample function block transfers the events into a buffer consisting consists of an array of elements that stores all data in a convenient format.

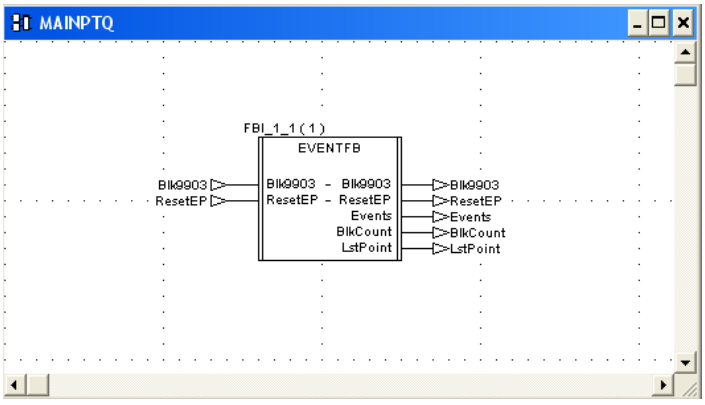
Block 9903 passes data into a compacted format, so as to occupy the minimum number of registers. For example, Block 9903 originally reserves the same register for Hour and Minute (one byte for each value), so your application would need to extract each value. The EVENTFB sample function block already extracts each event value into a separate register.

The following extract from the data type definition file describes the structure of each element of the buffer.

```
TYPE EVENT103:
STRUCT
     Session: BYTE;            (* Session configured for this slave *)
     Sector: BYTE;             (* Sector configured for this session *)
     ASDU: INT;                (* ASDU Type *)
     FUN: BYTE;                (* Function *)
     PointIndex: BYTE;         (* Point Index = Information Number*)
     Fault: UINT;              (* Fault for ASDU types 2 & 4 *)
     Hour: UINT;               (* Timestamp - hour *)
     Minute: UINT;             (* Timestamp - minute *)
     Seconds: UINT;            (* Timestamp - seconds *)
     Milliseconds: UINT;       (* Timestamp - milliseconds *)
     TimeValid: BOOL;          (* Time Valid 0=Valid, 1=Invalid *)
     DST: BOOL;                (* Daylight Savings Time 0=No , 1=Yes *)
     RelativeTime: UINT;       (* Relative Time for ASDU types 2 & 4 *)
     Value: ARRAY[0..1] OF INT; (* Data value - data size depends on ASDU
type *)
   END_STRUCT;
END_TYPE
```

Incoming events are stored an array of 60 "EVENT103" elements. This is a circular buffer that can contain up to 60 events. The element index can vary from 0 to 59. If the last event updated was located at index 59, then the next event will be copied to index 0.

The following illustration shows an instance example of the EVENTFB function block:



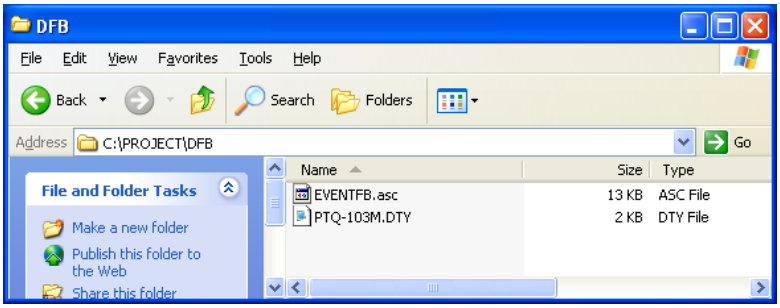
The EVENTFB function block contains the following PINs:

Pin	Pin Type	Data Type	Description
Blk9903	input/output	WORD130	Stores the memory area updated by block 9903. The start address must point to the same start address defined for block 9903 backplane data exchange (Point Address parameter).
ResetEP	input/output	INT	Move a value of one to reset the event pointer. This will cause the next event to be written to index 0 at the circular buffer. The register will be automatically reset to zero after the request was processed. This register should be only used for very specific applications (because the circular buffer automatically changes the element pointer from 59 to 0 after the maximum index was reached)
Events	output	EVENTSTRUCT	Circular buffer that stores all received events in a convenient format for the user application. It can store up to 60 events (index varies from 0 to 59). After event 59 is updated the next event to be received will be automatically updated at index 0.
BlkCount	output	INT	Incremented after a block is received (and after the events in that block have been read into the circular buffer). The maximum value for this counter is 10000 (then it is automatically reset to 0)
LstPoint	output	INT	Pointer to the last event index read from the module. For example, if last event was updated at index 5 then this value will have the same value.

Before You Begin

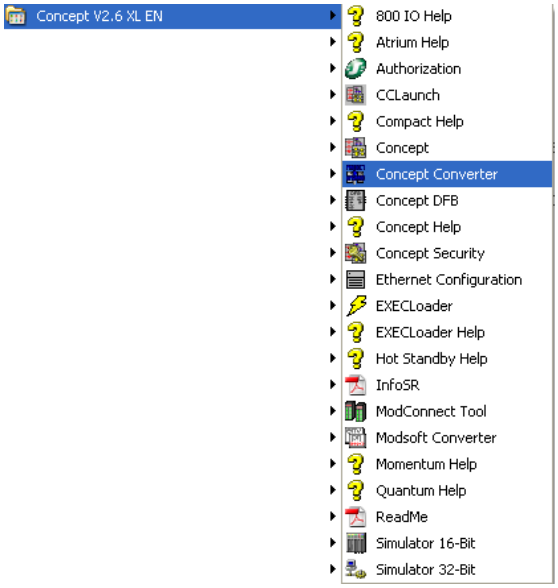
1. Make sure that your computer has the Concept Programming Unit installed.
2. The PTQ-103M firmware revision must support the event pass-thru functionality. This feature is available for version 1.12 or later. Refer to the SOFTWARE REVISION LEVEL value in the diagnostics menu of the PTQ-103M module (page 80).
   - Using Windows Explorer, create a folder for your Concept project with a "DFB" subfolder. This procedure will uses the folder C:\PROJECT\DFB, where:
   - C:\PROJECT- will store the main Concept project (.PRJ)
   - C:\PROJECT\DFB - will store the data type definition file (PTQ-103M.DTY) and the function block that will be presented later at this document.
3. Refer to the CD-ROM or to the web site for the PTQ103MConcept_Block9903.zip file and extract the two files below:
   - EVENTFB.asc (function block)
   - PTQ-103M.DTY (data type definition)

Use Windows Explorer to move these files to C:\PROJECT\DFB as follows:

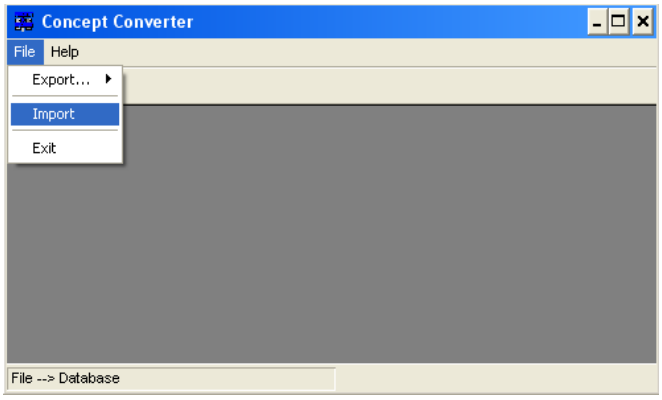


Convert the EVENTFB Function Block

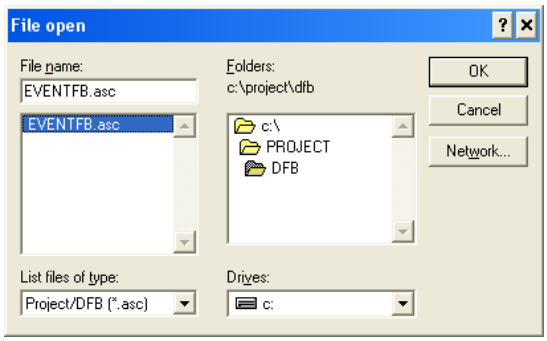
1 Click the **START** button, and navigate **TO CONCEPT V2.6 XL EN** / **CONCEPT CONVERTER**.



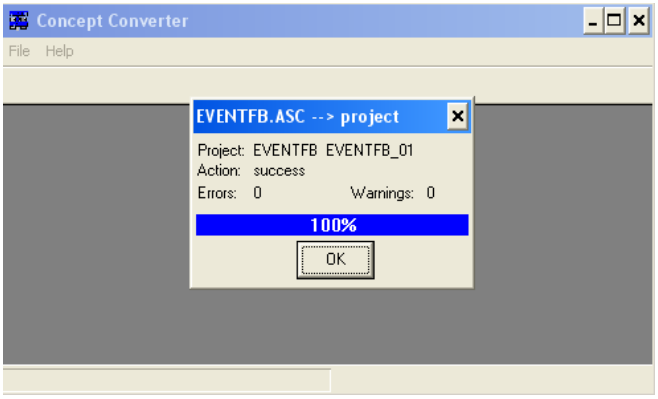
2 In the Concept Converter, open the **FILE** menu, and then choose **IMPORT**.



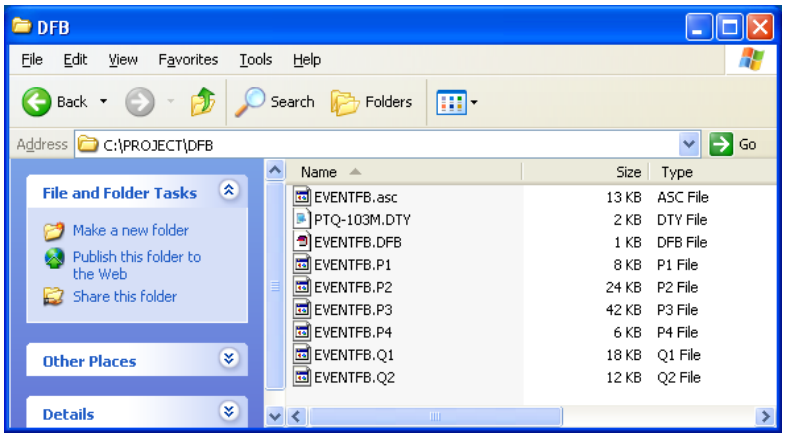
3 Select the **EVENTFB.ASC** file located in the folder **C:\PROJECT\DFB**, and then click **OK** to import the Function Block.



4 Concept Converter will import the function block. Click **OK** on the confirmation dialog box.

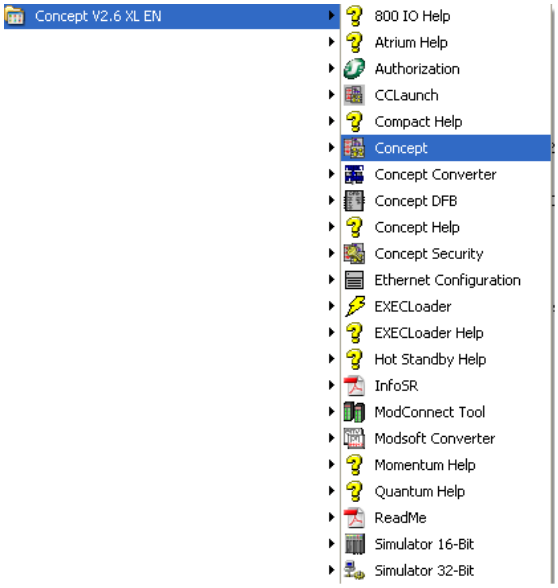


5 Close the Concept Converter tool. Now, use Windows Explorer to navigate to **C:\PROJECT\DFB,** and verify that the function block (.DFB) was exported and is ready to be used.

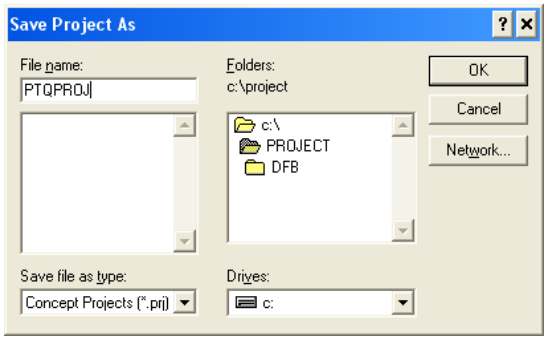


Setup the Concept Project

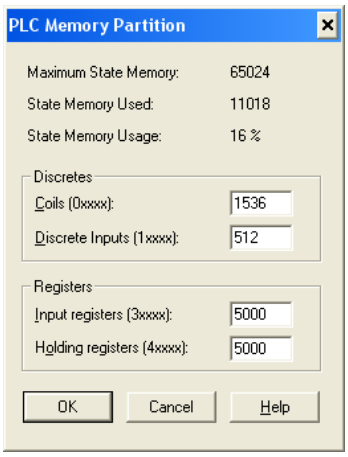
1 Click the **START** button, and navigate **TO CONCEPT V2.6 XL EN / CONCEPT**



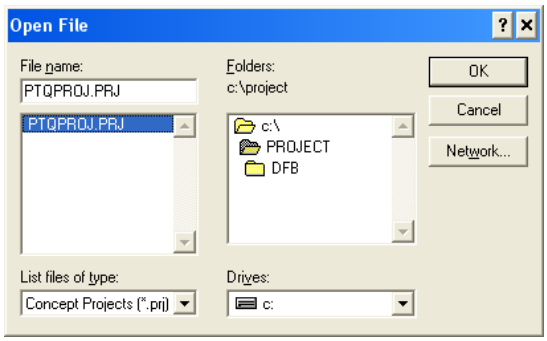
2 Create a new project and save it in the **C:\PROJECT** folder. For this example we will name the project **PTQPROJ**.



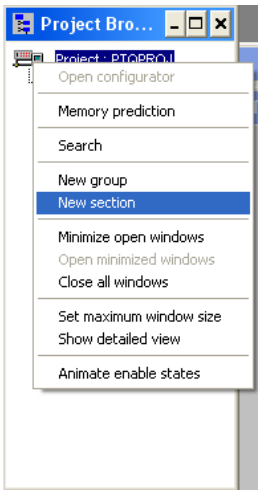
3 In the **PLC MEMORY PARTITION** dialog box, make sure that the processor memory range is sufficient for the PTQ-103M backplane usage.



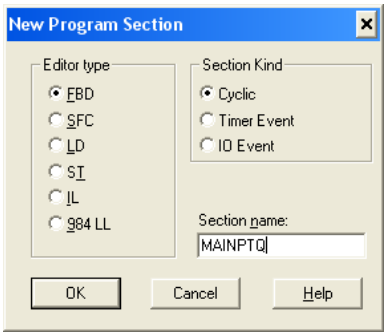
4 Open the **FILE** menu, and then chose **CLOSE PROJECT**.
5 Next, open the project again, and select the **PTQPROJ** file . This step allows the Concept application to recognize the new data types defined in the PTQ-103M.DTY file.

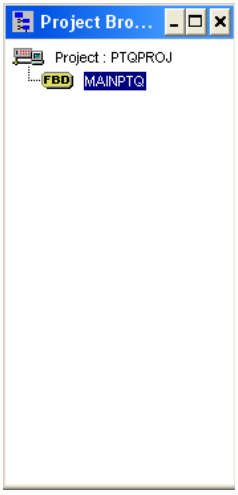


6 In the Project Browser pane, select **PROJECT: PTQPROJ** and click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW SECTION**

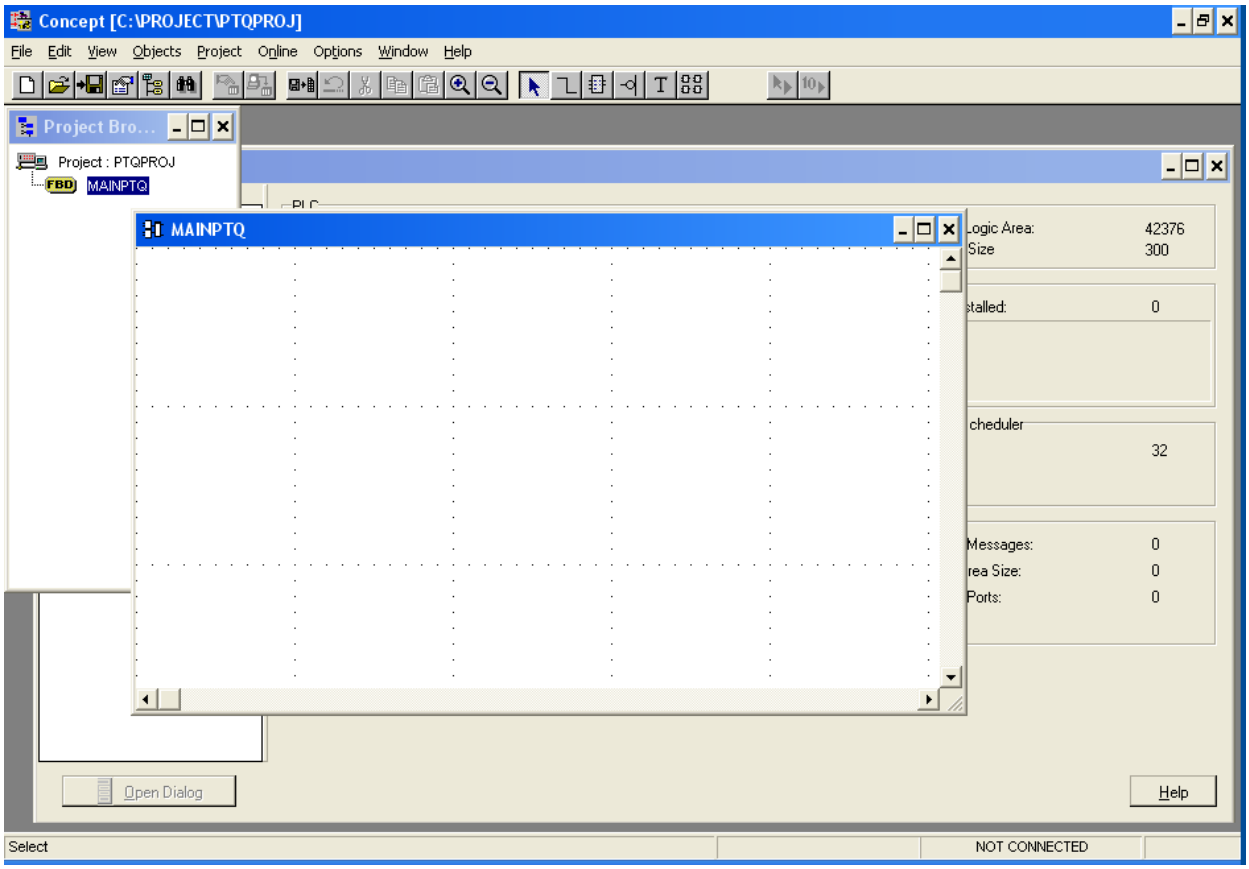


7 In the **NEW PROGRAM SECTION** dialog box, select **FBD**. In the **SECTION NAME** field, enter **MAINPTQ**, and then click **OK.**

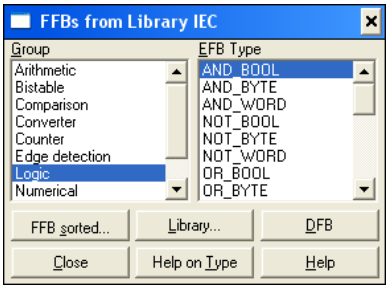




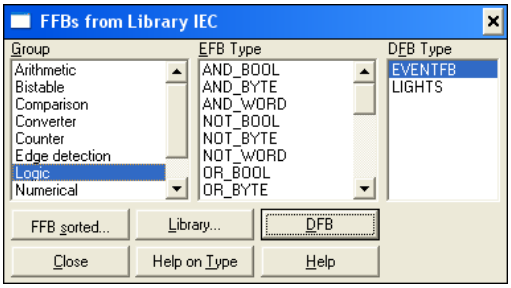
8 In the Project Browser, select and double-click the section you just added. This action opens the **FBD** section:



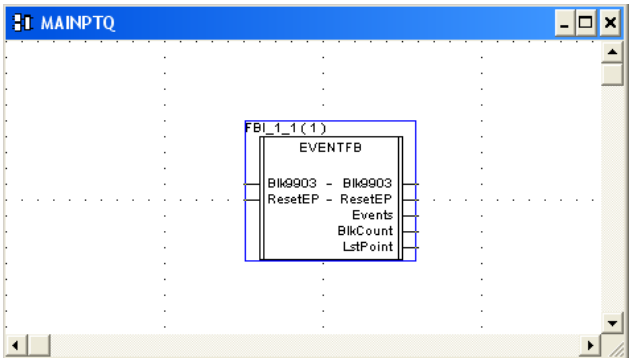
9 Select **OBJECTS-FFB SELECTION…**



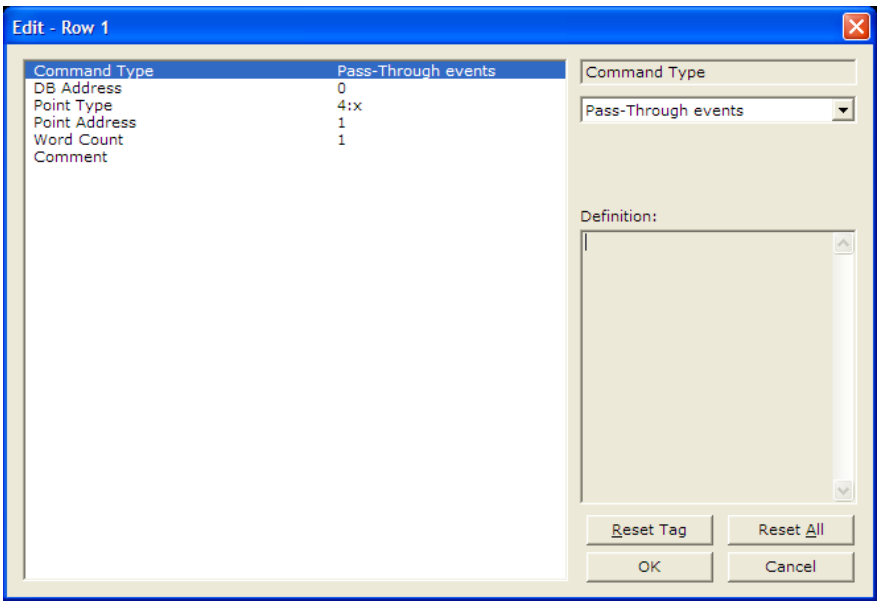
10 Click **DFB** button, and then select the **EVENTFB** function block, as shown in the following illustration. Click **CLOSE** to dismiss the dialog box.



The EVENTFB function block will now be visible in the FBD section:



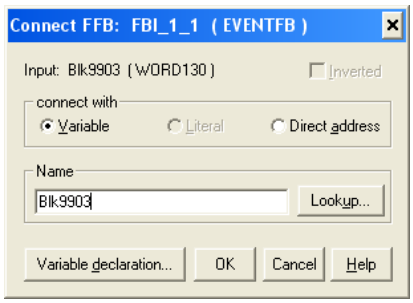
11 This step will create variables to associate to the function block PINs. We will start with the Blk9903 PIN. The variable for this PIN must point to the same start address where block 9903 will be copied. The following illustration shows an example configuration for Block 9903:



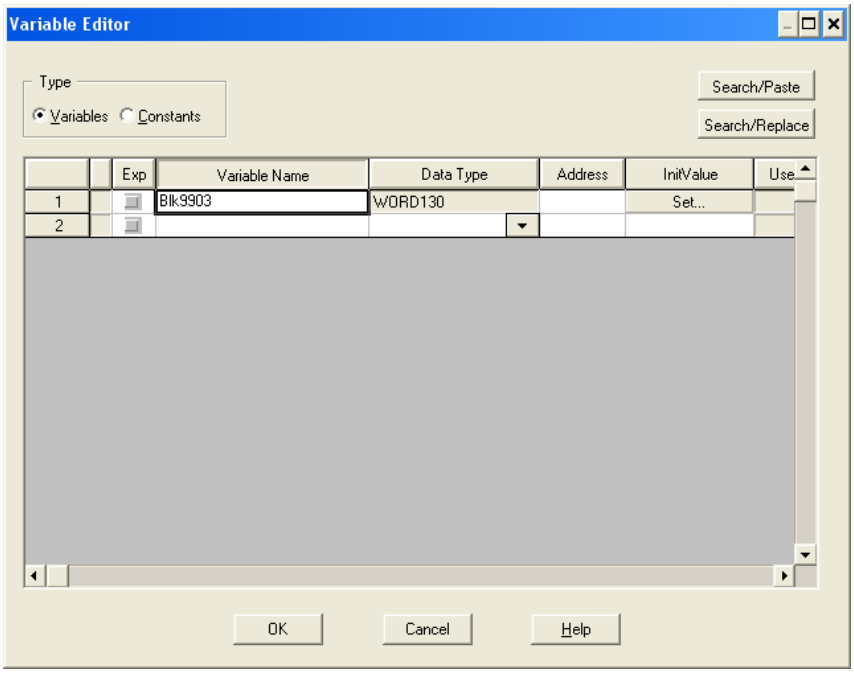
In this example, the variable associated to PIN Blk9903 must also start at the same register address (43001 for this example).

Note: this is the only variable to be associated to a PIN that requires the specification of a Quantum memory address.

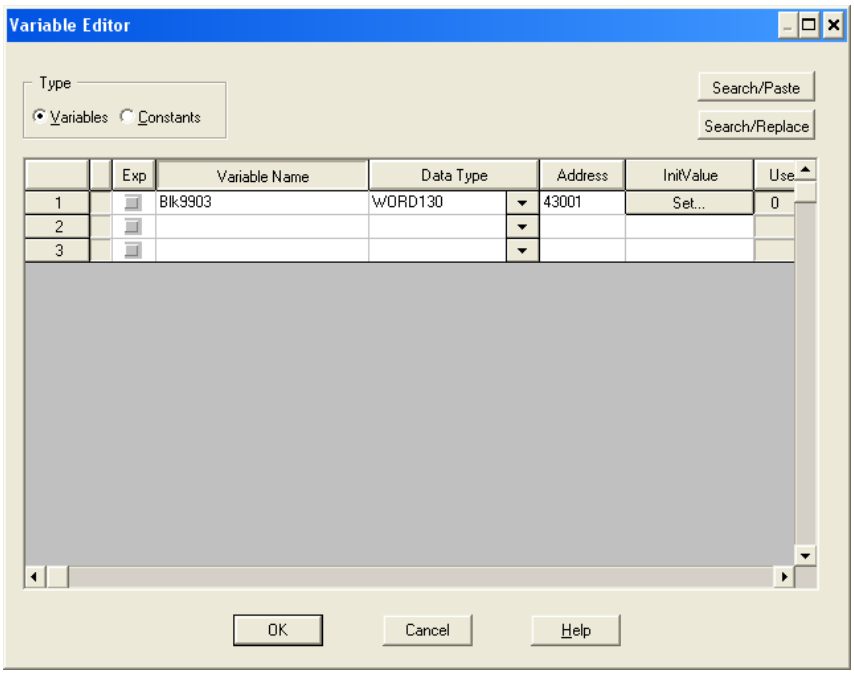
Double-click the **BLK9903 INPUT/OUTPUT PIN** and create a name for the variable to be associated. This example will use the same name as the PIN.



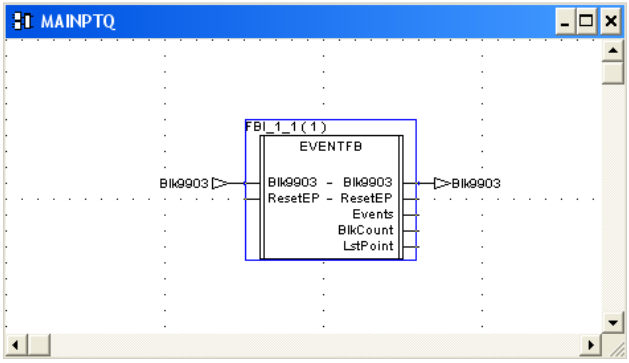
Click **VARIABLE DECLARATION**. This action opens the **VARIABLE EDITOR** dialog box.



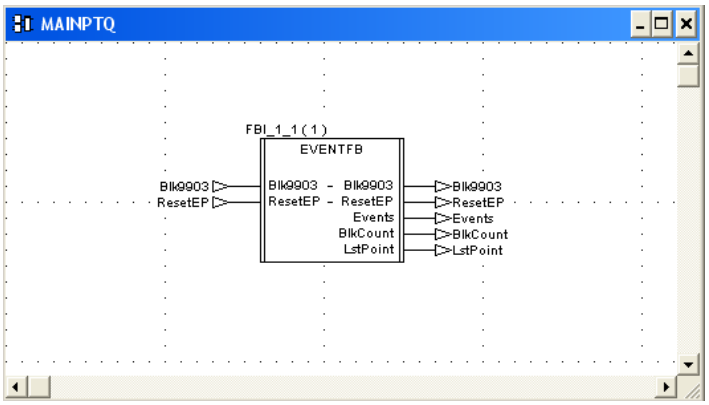
Select the memory address you have previously configured for Block 9903 (Backplane Exchange - Point Address). This example uses the value 43001:



Click **OK** to dismiss the Variable Editor dialog box. You should now see the new variable associated to the Blk9903 PIN:

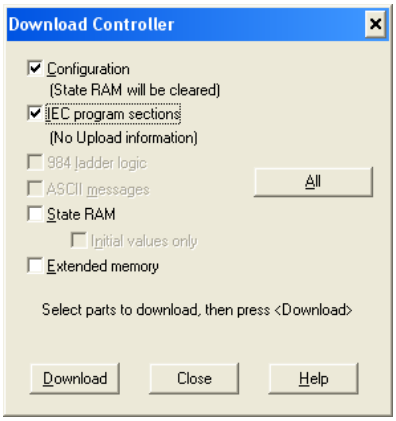


12 Repeat the previous step for the other PINS (it is not necessary to associate any memory address to the other variables).

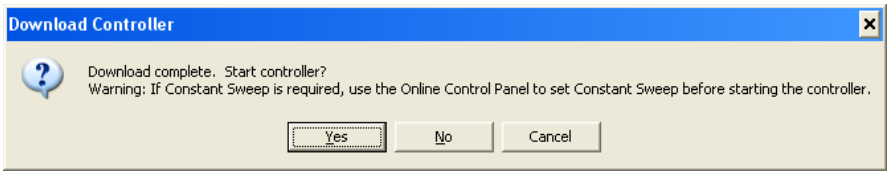


Download the Concept Project

1 Open the **ONLINE** menu, and then choose **DOWNLOAD** to download the Concept Project. Make sure that the IEC program sections checkbox is selected:



2 When the download is completed you should see the following confirmation dialog box. Click **YES** to continue.



Using the EVENTFB Function Block

In order to show how the function block can be used, we will create the following Template. This template shows the **BLKCOUNT**, **LSTPOINT**, and **RESETEP** variables, as well as the first two event elements (**EVENTS.EVENT[0]** and **EVENTS.EVENT[1]**):

RDE Template (untitled) - Animation ON

	Variable Name	Data Type	Address	Value	Set Value	Format
1						
2	BlkCount	INT		0		Dec
3	LstPoint	INT		0		Dec
4	ResetEP	INT		0		Dec
5						
6	Events.Event[0].Session	BYTE		0	0	Hex
7	Events.Event[0].Sector	BYTE		0		Hex
8	Events.Event[0].ASDU	INT		0		Dec
9	Events.Event[0].FUN	BYTE		0	0	Dec
10	Events.Event[0].PointIndex	BYTE		0	0	Dec
11	Events.Event[0].Fault	UINT		0		Uns Dec
12	Events.Event[0].Hour	UINT		0		Uns Dec
13	Events.Event[0].Minute	UINT		0		Uns Dec
14	Events.Event[0].Seconds	UINT		0		Uns Dec
15	Events.Event[0].Milliseconds	UINT		0		Uns Dec
16	Events.Event[0].TimeValid	BOOL		Off		Bool
17	Events.Event[0].DST	BOOL		Off		Bool
18	Events.Event[0].RelativeTime	UINT		0		Uns Dec
19	Events.Event[0].Value[0]	INT		0		Dec
20	Events.Event[0].Value[1]	INT		0		Dec
21						
22	Events.Event[1].Session	BYTE		0		Hex
23	Events.Event[1].Sector	BYTE		0		Hex
24	Events.Event[1].ASDU	INT		0		Dec
25	Events.Event[1].FUN	BYTE		0	0	Dec
26	Events.Event[1].PointIndex	BYTE		0	0	Dec
27	Events.Event[1].Fault	UINT		0		Uns Dec
28	Events.Event[1].Hour	UINT		0		Uns Dec
29	Events.Event[1].Minute	UINT		0		Uns Dec
30	Events.Event[1].Seconds	UINT		0		Uns Dec
31	Events.Event[1].Milliseconds	UINT		0		Uns Dec
32	Events.Event[1].TimeValid	BOOL		Off		Bool
33	Events.Event[1].DST	BOOL		Off		Bool
34	Events.Event[1].RelativeTime	UINT		0		Uns Dec
35	Events.Event[1].Value[0]	INT		0		Dec
36	Events.Event[1].Value[1]	INT		0		Dec
37						

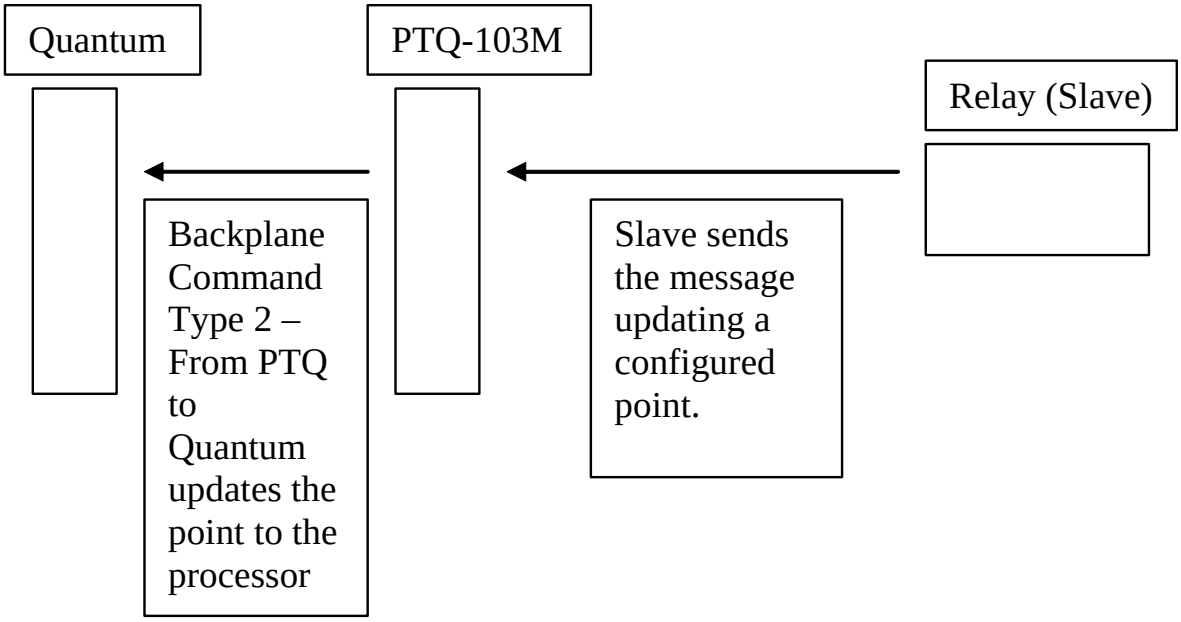
In this example, the remote device has sent two events with timestamp to the module (in different blocks 9903). The following illustration shows how the variables associated to the EVENTFB function block would be updated.

- **BLKCOUNT**: shows a value of 2 because the processor has received two blocks 9903
- **LSTPOINT**: shows a value of 1 because the last element that was updated has an index of 1 (**EVENTS.EVENT[1]**).
- **EVENTS.EVENT[0]**: shows the first event received from the module
- **EVENTS.EVENT[1]**: shows the second event received from the module

RDE Template (untitled) - Animation ON

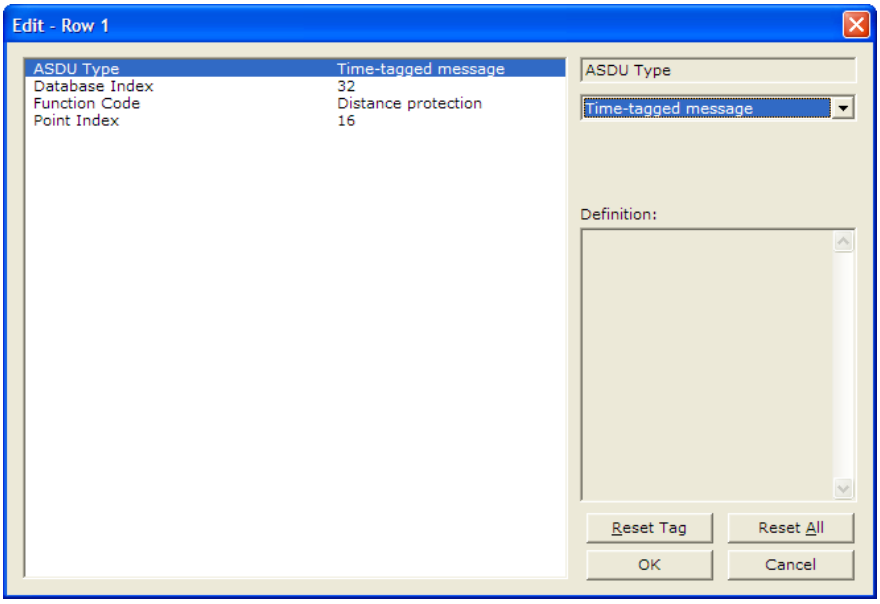
	Variable Name	Data Type	Address	Value	Set Value	Format
1						
2	BlkCount	INT		2		Dec
3	LstPoint	INT		1		Dec
4	ResetEP	INT		0		Dec
5						
6	Events.Event[0].Session	BYTE		0	0	Hex
7	Events.Event[0].Sector	BYTE		0		Hex
8	Events.Event[0].ASDU	INT		1		Dec
9	Events.Event[0].FUN	BYTE		128	0	Dec
10	Events.Event[0].PointIndex	BYTE		16	0	Dec
11	Events.Event[0].Fault	UINT		0		Uns Dec
12	Events.Event[0].Hour	UINT		13		Uns Dec
13	Events.Event[0].Minute	UINT		36		Uns Dec
14	Events.Event[0].Seconds	UINT		11		Uns Dec
15	Events.Event[0].Milliseconds	UINT		250		Uns Dec
16	Events.Event[0].TimeValid	BOOL		Off		Bool
17	Events.Event[0].DST	BOOL		Off		Bool
18	Events.Event[0].RelativeTime	UINT		0		Uns Dec
19	Events.Event[0].Value[0]	INT		2		Dec
20	Events.Event[0].Value[1]	INT		0		Dec
21						
22	Events.Event[1].Session	BYTE		0		Hex
23	Events.Event[1].Sector	BYTE		0		Hex
24	Events.Event[1].ASDU	INT		1		Dec
25	Events.Event[1].FUN	BYTE		128	0	Dec
26	Events.Event[1].PointIndex	BYTE		16	0	Dec
27	Events.Event[1].Fault	UINT		0		Uns Dec
28	Events.Event[1].Hour	UINT		13		Uns Dec
29	Events.Event[1].Minute	UINT		36		Uns Dec
30	Events.Event[1].Seconds	UINT		21		Uns Dec
31	Events.Event[1].Milliseconds	UINT		515		Uns Dec
32	Events.Event[1].TimeValid	BOOL		Off		Bool
33	Events.Event[1].DST	BOOL		Off		Bool
34	Events.Event[1].RelativeTime	UINT		0		Uns Dec
35	Events.Event[1].Value[0]	INT		2		Dec
36	Events.Event[1].Value[1]	INT		0		Dec
37						

8.2.2 Monitor Direction



Example

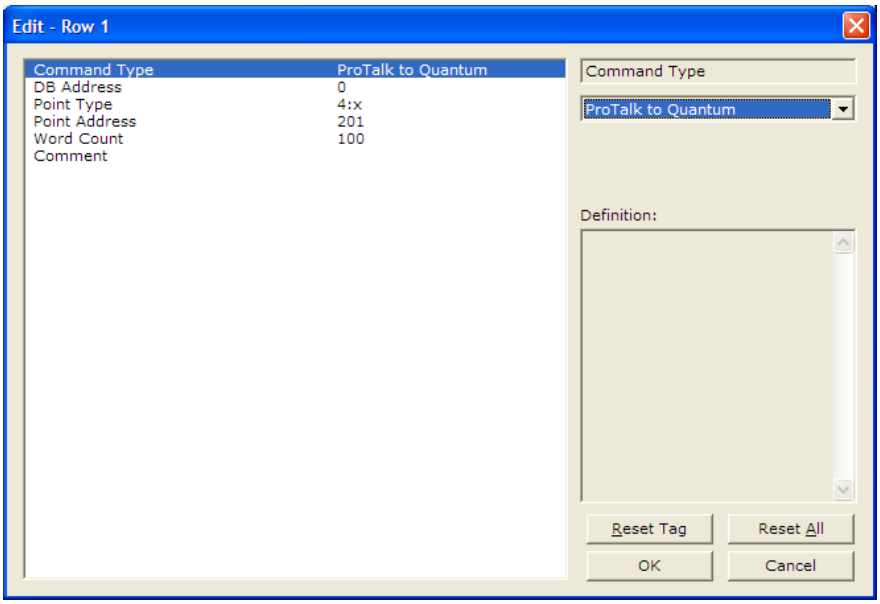
Configure the following monitor point in the **[IEC-103 MASTER SESSION X SECTOR X]** section in ProSoft Configuration Builder:



- ASDU Type = Time-tagged message
- Function Code = Distance protection
- Information Number (Point Index) = 16 (auto-recloser active)

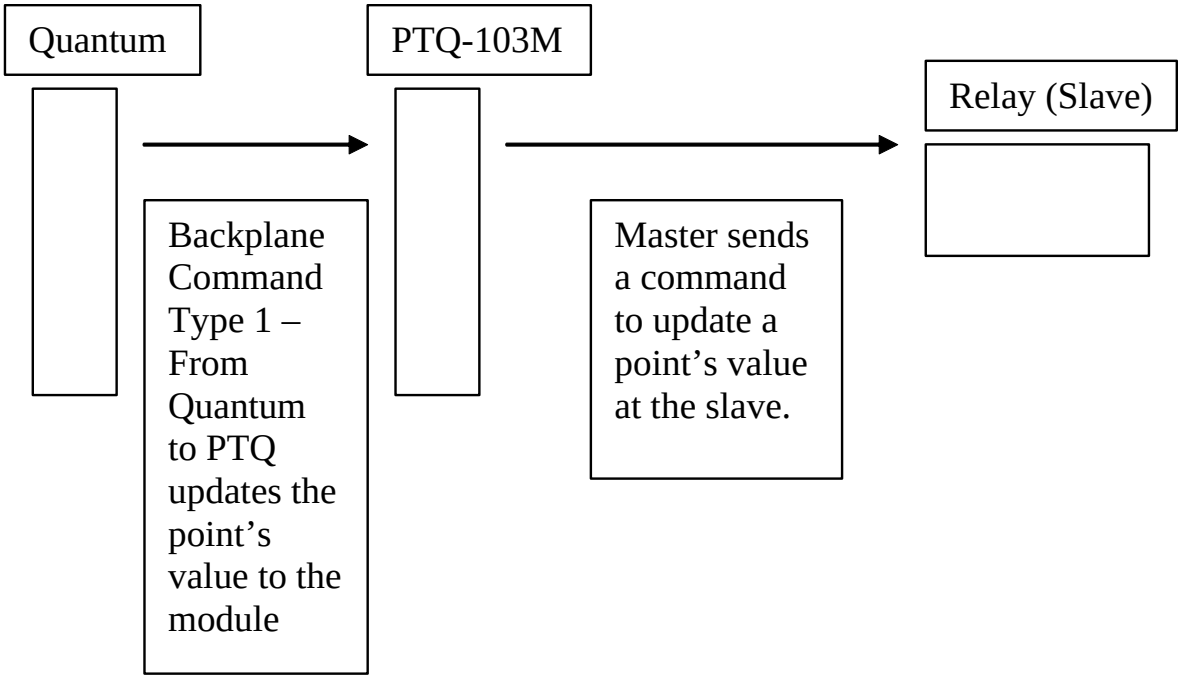
- The point's value is associated to database address 32 (bit address 32 = bit 0 from word address 2).

The following **[BACKPLANE DATA EXCHANGE]** command would update this value from the module to the Quantum processor:



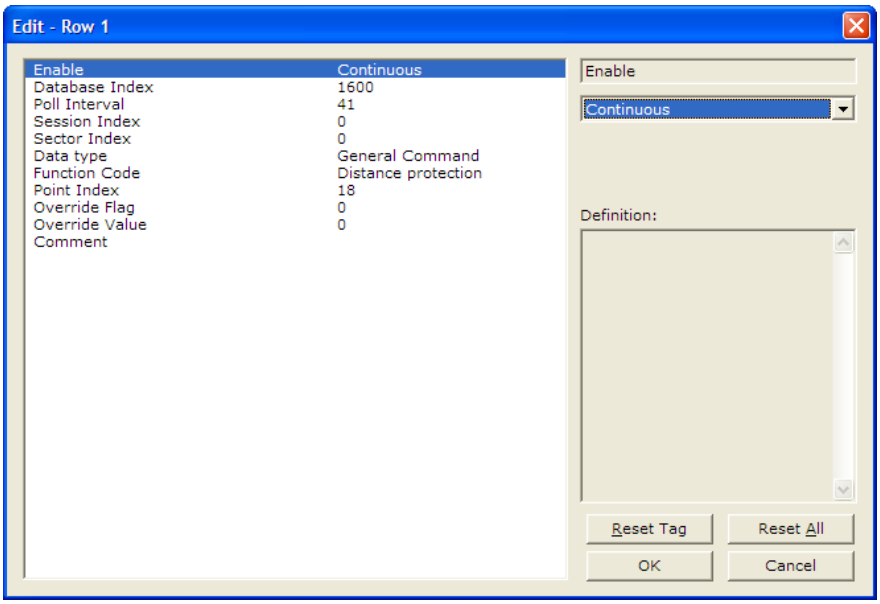
Therefore, the Quantum's holding register address 400203 will be automatically updated by the PTQ-103M's database address 2 (containing the value of the configured monitor point).

8.2.3 Control Direction



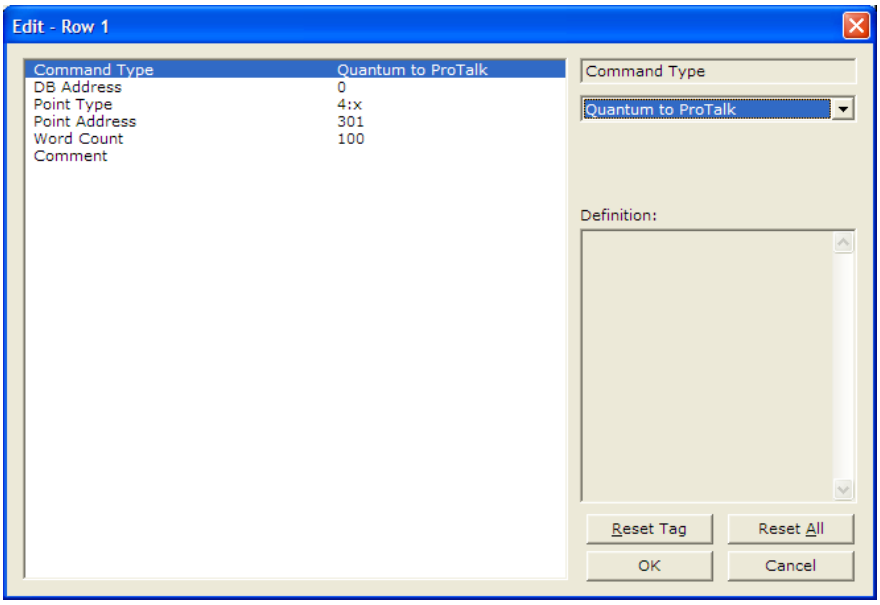
Example

Configure the following command point in the **[IEC-103 MASTER SESSION COMMANDS]** section in ProSoft Configuration Builder:



- Data Type = General Command
- Function Code = Distance protection
- Information Number (Point Index) = 18 (protection ON/OFF)
- The point's value is associated to database address 1600 (bit address 1600 = bit 0 from word address 100).

The following **[BACKPLANE DATA EXCHANGE]** command would update this value from the Quantum processor to the module:



Therefore, the Quantum's holding register address 400301 will automatically update PTQ-103M's database address 100 (containing the value of the configured command point).

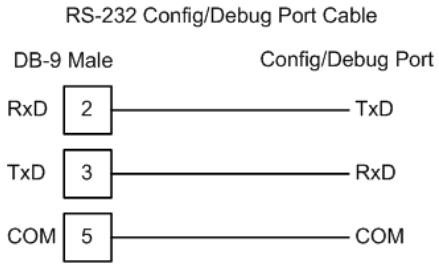
8.3 Cable Connections

The application ports on the PTQ-103M module support RS-232, RS-422, and RS-485 interfaces. Please inspect the module to ensure that the jumpers are set correctly to correspond with the type of interface you are using.

Note: When using RS-232 with radio modem applications, some radios or modems require hardware handshaking (control and monitoring of modem signal lines). Enable this in the configuration of the module by setting the UseCTS parameter to 1.

8.3.1 RS-232 Configuration/Debug Port

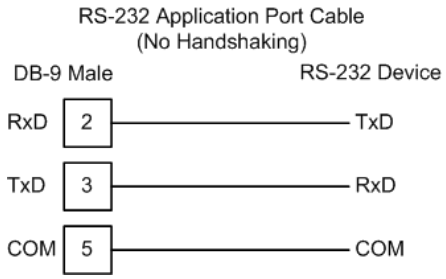
This port is physically a DB-9 connection. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following diagram:



The Ethernet port on this module (if present) is inactive.

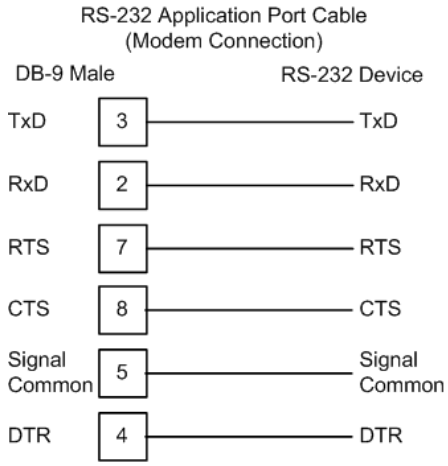
8.3.2 RS-232

When the RS-232 interface is selected, the use of hardware handshaking (control and monitoring of modem signal lines) is user definable. If no hardware handshaking will be used, the cable to connect to the port is as shown below:



RS-232: Modem Connection

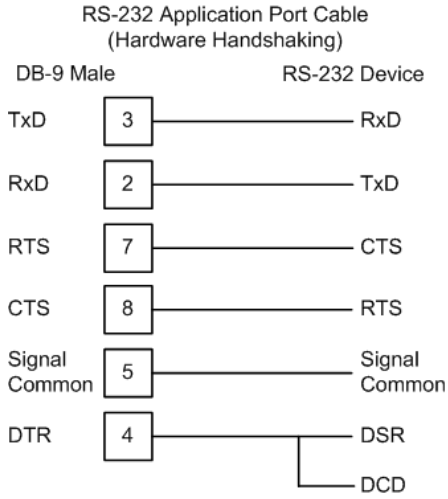
This type of connection is required between the module and a modem or other communication device.



The "Use CTS Line" parameter for the port configuration should be set to 'Y' for most modem applications.

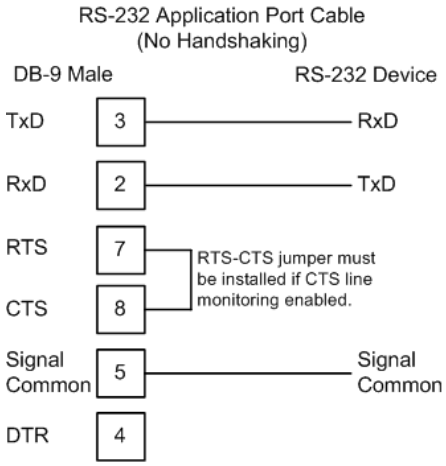
RS-232: Null Modem Connection (Hardware Handshaking)

This type of connection is used when the device connected to the module requires hardware handshaking (control and monitoring of modem signal lines).



RS-232: Null Modem Connection (No Hardware Handshaking)

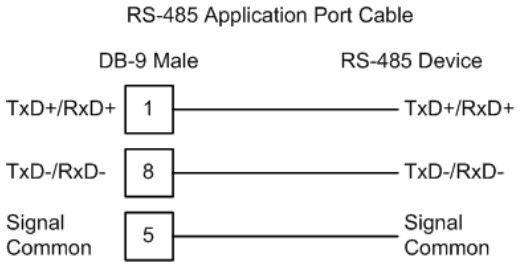
This type of connection can be used to connect the module to a computer or field device communication port.



Note: If the port is configured with the "Use CTS Line" set to 'Y', then a jumper is required between the RTS and the CTS line on the module connection.

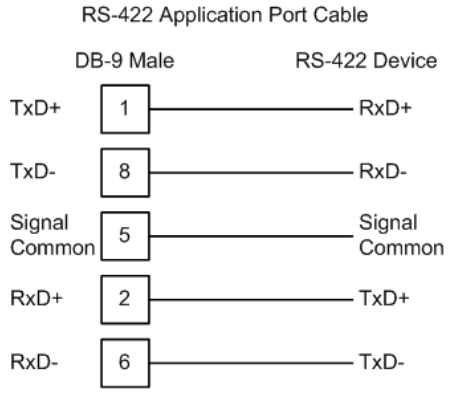
8.3.3 RS-485

The RS-485 interface requires a single two or three wire cable. The Common connection is optional and dependent on the RS-485 network. The cable required for this interface is shown below:



Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In this case, install a 120-ohm terminating resistor on the RS-485 line.

8.3.4 RS-422



RS-485 and RS-422 Tip

If communication in the RS-422/RS-485 mode does not work at first, despite all attempts, try switching termination polarities. Some manufacturers interpret +/- and A/B polarities differently.

8.4 PTQ-103M Error Status Table

This section contains a listing of the PTQ-103M module's status data area. Use backplane command 2 to transfer this data into the database.

Offset	Parameter	Description
4000	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
4001 to 4002	Product Name	This two-word data area contains the text values representing the product name. These words contain the text 'I3M5' for the PTQ-103M platform.
4003 to 4004	Revision	This two-word data area contains the text values for the revision number.
4005 to 4006	Op Sys #	This two-word data area contains the text values for the operating system number.
4007 to 4008	Run Number	This two-word data area contains the text values for the run number
4009	Reserved	Not used
4010	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
4011	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
4012	Parse Blk Cnt	This word contains the total number of write blocks successfully parsed.
4013	Error Blk Cnt	This word contains the total number of block transfer errors.
4014 to 4019	Reserved	Not used

Offset	Parameter	Description
4020	Cmd Req	This word contains the number of commands transferred out channel 0.
4021	Cmd Resp	This word contains the number of command response messages received on channel 0.
4022	Cmd Err	This word contains the number of command errors recognized on channel 0.
4023	Requests	This word contains the total number of messages transmitted on channel 0.
4024	Responses	This word contains the total number of messages received on channel 0.
4025	Err Sent	This word contains the number of error messages sent on channel 0.
4026	Err Received	This word contains the number of error messages received on channel 0.
4027	Cfg Err	This bit mapped word recognizes any configuration errors for channel 0. Refer to the configuration error word table for a definition of each bit.
4028	Current Error	This word contains the error code for the current command executing on channel 0.
4029	Last Error	This word contains the error code for the last error recognized on channel 0.
4030	Cmd Req	This word contains the number of commands transferred out channel 1.
4031	Cmd Resp	This word contains the number of command response messages received on channel 1.
4032	Cmd Err	This word contains the number of command errors recognized on channel 1.
4033	Requests	This word contains the total number of messages transmitted on channel 1.
4034	Responses	This word contains the total number of messages received on channel 1.
4035	Err Sent	This word contains the number of error messages sent on channel 1.
4036	Err Received	This word contains the number of error messages received on channel 1.
4037	Cfg Err	This bit mapped word recognizes any configuration errors for channel 1. Refer to the configuration error word table for a definition of each bit.
4038	Current Error	This word contains the error code for the current command executing on channel 1.
4039	Last Error	This word contains the error code for the last error recognized on channel 1.
4040	SessionConfig	Total number of sessions configured
4041 to 4042	OnlineState	Bitmap double-word (32 bits), where each bit indicates if associated session is online (example bit - = 1 means session is online)
4043	CmdCount	Total number of commands configured
4044	MaxCmd	Maximum number of commands supported (1000)

Offset	Parameter	Description
4045	CurrCmd	Current Command Executed
4046	CmdStatus	Command Status (0 = idle, (other = busy)
4047	EventMsgCount	Event Message Count (incremented on every event)
4048	EventMsgOverflow	Event Message Overflow

Each bit in the word corresponds to an error condition recognized when the module is configured. There is a separate word for each application port. This data is reported in the status data area.

The following table describes the structure of the configuration error word.

Bit	Code	Description
0	0x0001	Invalid baud rate selected
1	0x0002	Invalid parity selected
2	0x0004	Received timeout set to 0
3	0x0008	Invalid Port selected for a session
4	0x0010	Invalid sector count for session
5	0x0020	Could not allocate memory for sector of a session.
6	0x0040	
7	0x0080	Invalid failure delay or confirm timeout for session.
8	0x0100	
9	0x0200	
10	0x0400	
11	0x0800	
12	0x1000	
13	0x2000	
14	0x4000	
15	0x8000	

8.5 PTQ-103M IEC 60870-5-103 Master Communication Module Error Codes

The following table describes all potential errors that can be generated by the IEC 60870-5-103 Master driver:

Error	Description
51	Physical layer error: Error transmitting message
52	Physical layer error: Intercharacter timeout occurred before message fully received.
53	Physical layer error: Frame not entirely received before timeout condition.
54	Physical layer error: Invalid frame length.
101	Link layer error: Invalid checksum received
102	Link layer error: Address unknown to module
103	Link layer error: Link established
104	Link layer error: Link failed
105	Link layer error: Received primary

Error	Description
106	Link layer error: FCB error discard
107	Link layer error: FCB error repeat
108	Link layer error: Invalid start character received
109	Link layer error: Invalid second character received
110	Link layer error: Invalid ending character received
111	Link layer error: Length mismatch error
112	Link layer error: Illegal function
113	Link layer error: No confirmation received
114	Link layer error: No ACK received
115	Link layer error: Sequence unknown
116	Link layer error: Out of sequence
117	Link layer error: Remote close
118	Link layer error: Unexpected ACK
119	Link layer error: Request cancelled
201	Application layer error: Length mismatch
202	Application layer error: Address unknown
203	Application layer error: Response late
251	RBE error: Clock event buffer overflow
252	RBE error: Event buffer overflow
271	Data error: Address unknown
281	Control error: Illegal operation
282	Control error: Illegal value
283	Control error: Not selected
301	Initialization error: Database
302	Initialization error: Out of memory
401	Channel open error
501	Session error: Database
502	Session error: Configuration
601	No memory to receive message
602	Session not reserved
603	Illegal session
604	Session is reserved
605	Session is not available
701	No memory to transmit message
702	ASDU not supported
703	Duplicate request
704	Illegal sector
705	Control mode is illegal
801	Partial stop request
802	Stop request failed

Error	Description
901	Response timeout
902	Negative COT in response
903	Session is offline
904	Session is disabled
905	Select confirmation received, waiting to execute
906	Execute confirmation has not be received
907	Command Timeout.

8.6 Protocol Support

This section describes the portions of IEC 60870-5-103 protocol that are supported by the module.

Note: Shaded areas are not supported by the module.

8.6.1 List of Type Identification Codes

In Monitor Direction

Type	Description
1	time-tagged message
2	time-tagged message with relative time
3	measurands I
4	time-tagged measurands with relative time
5	identification
6	time synchronization
8	general interrogation termination
9	measurands II
10	generic data
11	generic identification
23	list of recorded disturbances
26	ready for transmission of disturbance data
27	ready for transmission of channel
28	ready for transmission of tags
29	transmission of tags
30	transmission of disturbance values
31	end of transmission

In Control Direction

Type	Description
6	time synchronization
7	general interrogation
10	generic data
20	general command

Type	Description
21	generic command
24	order for disturbance data transmission
25	ack for disturbance data transmission

8.6.2 List of Cause of Transmission Codes

In Monitor Direction

COT	Description
1	spontaneous
2	cyclic
3	reset frame count bit (FCB)
4	reset communication unit (CU)
5	start/restart
6	power on
7	test mode
8	time synchronization
9	general interrogation
10	termination of general interrogation
11	local operation
12	remote operation
20	positive ack of command
21	negative ack of command
31	transmission of disturbance data
40	positive ack of generic write command
41	negative ack of generic write command
42	valid data response to generic read command
43	invalid data response to generic read command
44	generic write confirmation

In Control Direction

COT	Description
8	time synchronization
9	initiation of general interrogation
20	general command
31	transmission of disturbance data
40	generic write command
42	generic read command

8.6.3 List of Function Types

Fun	Description	Symbol
128	Distance protection	t(z)
160	Overcurrent protection	I>>
176	Transformer differential protection	ΔI_T
192	Line differential protection	ΔIL
254	Generic function type	GEN
255	Global function type	GLB

8.6.4 Information Numbers Used in Monitor Direction

System functions

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
0	end of general interrogation		8						X
0	time synchronization		6						X
2	reset FCB		5						*
3	reset CU		5						*
4	start/restart		5						*
5	power on		5						*
* - According to main function									

Status Indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
16	auto-recloser active	X	1	X	X		X		
17	teleprotection active	X	1	X	X				
18	protection active	X	1	X	X	X	X		
19	LED reset		1	X	X	X	X		
20	monitor direction blocked	X	1	X	X	X	X		
21	test mode	X	1	X	X	X	X		
22	local parameter setting	X	1	X	X	X	X		
23	characteristic 1	X	1	X					
24	characteristic 2	X	1	X					
25	characteristic 3	X	1	X					
26	characteristic 4	X	1	X					
27	auxiliary input 1	X	1	X	X	X	X		
28	auxiliary input 2	X	1	X	X	X	X		
29	auxiliary input 3	X	1	X	X	X	X		
30	auxiliary input 4	X	1	X	X	X	X		

Supervision indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
32	measured supervision I	X	1	X	X				
33	measured supervision V	X	1	X	X				
35	phase sequence supervision	X	1	X	X				
36	trip circuit supervision	X	1	X	X	X	X		
37	I>>back-up operation	X	1	X					
38	VT fuse failure	X	1	X	X				
39	teleprotection disturbed	X	1	X	X		X		
46	group warning	X	1	X	X	X	X		
47	group alarm	X	1	X	X	X	X		

Earth fault indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
48	earth fault L_1	X	1	X	X				
49	earth fault L_2	X	1	X	X				
50	earth fault L_3	X	1	X	X				
51	earth fault forward (that is, line)	X	1	X	X				
52	earth fault reverse (that is, busbar)	X	1	X	X				

Fault indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
64	start/pick-up L_1	X	2	X	X		X		
65	start/pick-up L_2	X	2	X	X		X		
66	start/pick-up L_3	X	2	X	X		X		
67	start/pick-up N	X	2	X	X		X		
68	general trip		2	X	X	X	X		
69	trip L_1		2	X	X	X	X		
70	trip L_2		2	X	X	X	X		
71	trip L_3		2	X	X	X	X		
72	trip I>> (back-up operation)		4	X					
73	fault location X in ohms		2	X	X				
74	fault forward/line		2	X	X				
75	fault reverse/busbar		2	X	X				
76	teleprotection signal transmitted		2	X	X				

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
77	teleprotection signal received		2	X	X				
78	zone 1		2	X					
79	zone 2		2	X					
80	zone 3		2	X					
81	zone 4		2	X					
82	zone 5		2	X					
83	zone 6		2	X					
84	general start/pick-up	X	2	X	X	X	X		
85	breaker failure		2	X	X				
86	trip measuring system L_1		2			X			
87	trip measuring system L_2		2			X			
88	trip measuring system L_3		2			X			
89	trip measuring system E		2			X			
90	trip I>		2		X				
91	trip I>>		2		X				
92	trip IN>		2		X				
93	trip IN>>		2		X				

Auto-reclosure indications

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
128	CB 'on' by AR		1	X	X		X		
129	CB 'on' by long-time AR		1	X	X		X		
130	AR blocked	X	1	X	X		X		

Measurands

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
144	measurand I		3.1	X	X				
145	measurands I, V		3.2	X	X				
146	measurands I, V, P, Q		3.3	X					
147	measurands I_N, V_{EN}		3.4	X	X				
148	measurands $IL_{1,2,3,}$ $VL_{1,2,3,}$ P, Q, f		9	X					

Generic functions

Inf	Description	GI	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
240	read headings of all defined groups		10					X	
241	read values or attributes of all entries of one group		10					X	
243	read directory of a single entry		11					X	
244	read value or attribute of a single entry	(x)	10					X	
245	end of general interrogation of generic data		10					X	
249	write entry with confirmation		10					X	
250	write entry with execution		10					X	
251	write entry aborted		10					X	

8.6.5 Information Numbers used in Control Direction

System functions

Inf	Description	Type	Fun					
			128	160	176	192	254	255
			t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
0	initiation of general interrogation	7						X
0	time synchronization	6						X

General commands

Inf	Description	Com	Type	Fun					
				128	160	176	192	254	255
				t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
16	auto-recloser on/off	On/Off	20	X	X		X		
17	teleprotection on/off	On/Off	20	X	X				
18	protection on/off	On/Off	20	X	X	X	X		
19	LED reset	On	20	X	X	X	X		
23	activate characteristic 1	On	20	X					
24	activate characteristic 2	On	20	X					
25	activate characteristic 3	On	20	X					
26	activate characteristic 4	On	20	X					

Generic functions

Inf	Description	Type	Fun					
			128	160	176	192	254	255
			t(z)	I>>	ΔI_T	ΔI_L	GEN	GLB
240	read headings of all defined groups	21					X	
241	read values or attributes of all entries of one group	21					X	
243	read directory of a single entry	21					X	
244	read value or attribute of a single entry	21					X	
245	general interrogation of generic data	21					X	
248	write entry	10					X	
249	write entry with confirmation	10					X	
250	write entry with execution	10					X	
251	write entry abort	10					X	

8.6.6 Definition and Presentation of ASDUs In Monitor Direction

TYPE 1: Time-tagged

Bytes	Description	Spec
1	Value of 1	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
1	DPI (0 to 3)	7.2.6.5
4	4-octet binary time	7.2.6.28
1	SIN	7.2.6.23

TYPE 2: Time-tagged with relative time

Bytes	Description	Spec
1	Value of 2	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
1	DPI (0 to 3)	7.2.6.5
2	RET (relative time)	7.2.6.15
2	FAN (fault number)	7.2.6.6
4	4-octet binary time	7.2.6.28
1	SIN	7.2.6.23

TYPE 3: Measurands I

Bytes	Description	Spec
1	Value of 3	
1	i = value of 1, 2 or 4	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
2	Current L_2	7.2.6.8
2	Voltage L_1-L_2	7.2.6.8
2	Active power P	7.2.6.8
2	Reactive power Q	7.2.6.8
	ASD U3.1: i=1 ASDU 3.2: i=2 ASDU 3.3: i=4 ASDU 3.4: i=2 val1=I_N and val2=V_{EN}	

TYPE 4: Time-tagged measurands with relative time

Bytes	Description	Spec
1	Value of 4	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
4	SCL (short-circuit location (real))	7.2.6.20
2	RET (relative time)	7.2.6.15
2	FAN (fault number)	7.2.6.6
4	4-octet binary time	7.2.6.28

TYPE 5: Identification

Bytes	Description	Spec
1	Value of 5	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
1	COL (compatibility level 2 or 3)	7.2.6.3
1	Char 1	7.2.6.2
1	Char 2	7.2.6.2

Bytes	Description	Spec
1	Char 3	7.2.6.2
1	Char 4	7.2.6.2
1	Char 5	7.2.6.2
1	Char 6	7.2.6.2
1	Char 7	7.2.6.2
1	Char 8	7.2.6.2
1	Manufacture byte	Free assignment
1	Manufacture byte	Free assignment
1	Manufacture byte	Free assignment
1	Manufacture byte	Free assignment

TYPE 6: Time synchronization

Bytes	Description	Spec
1	Value of 6	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
7	7-Octet binary time	7.2.6.29

TYPE 8: Termination of general interrogation

Bytes	Description	Spec
1	Value of 8	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
1	SCN (scan number 0 to 255)	7.2.6.21

TYPE 9: Measurands II

Bytes	Description	Spec
1	Value of 9	
1	I = 1 to 9 for number of values	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	
2	Current L_1	7.2.6.8

Bytes	Description	Spec
2	Current L_2	7.2.6.8
2	Current L_3	7.2.6.8
2	Voltage L_{1-E}	7.2.6.8
2	Voltage L_{2-E}	7.2.6.8
2	Voltage L_{3-E}	7.2.6.8
2	Active power P	7.2.6.8
2	Reactive power Q	7.2.6.8
2	Frequency f	7.2.6.8
	Value of I determines number of parameters passed in the message with a range of 1 to 9 but always starts with the L_1 value.	

8.6.7 Definition and Presentation of ASDU's in Control Direction

TYPE 6: Time synchronization

Bytes	Description	Spec
1	Value of 6	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type = GLB (255)	
1	Information number	
7	7-Octet binary time	7.2.6.29

TYPE 7: Initiation of general interrogation

Bytes	Description	Spec
1	Value of 7	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type = GLB (255)	
1	Information number	
1	SCN (scan number 0 to 255)	7.2.6.21

TYPE 20: General command

Bytes	Description	Spec
1	Value of 20	
1	0x81	
1	COT	
1	Common address of ASDU	
1	Function Type	
1	Information number	

Bytes	Description	Spec
1	DCO (1=Off, 2=On)	7.2.6.4
1	RII	7.2.6.19
	RII is not to be processed within the protection equipment, but to be used as a SIN of the return message.	

8.7 Protocol Interoperability Documentation

This companion standard presents sets of parameters and alternatives from which subsets have to be selected to implement particular telecontrol systems. Certain parameter values, such as the number of octets in the COMMON ADDRESS of ASDUs represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information in command and in monitor direction allow the specification of the complete set or subsets, as appropriate for given applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment stemming from different manufacturers it is necessary that all partners agree on the selected parameters.

Note: In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should be marked in the white boxes as follows:

☐	Function or ASDU is not used
☒	Function or ASDU is used as standardized (default)
R	Function or ASDU is used in reverse mode
B	Function or ASDU is used in standard and reverse mode

The possible selection (blank, X, R, B) is specified for each specific clause or parameter.

A black check box indicates that the option cannot be selected in this companion standard.

The pages in this section have been extracted from the 60870-5-103 © IEC:1997, pages 159 to 171.

8.7.1 Physical Layer

8.7.2 Electrical Interface

☒	EIA RS-485
☒	Number of loads 32. For one protection equipment

NOTE - EIA RS-485 standard defines unit loads so that 32 of them can be operated on one line. For detailed information refer to clause 3 of EIA RS-485 standard.

8.7.3 Optical Interface

☐	Glass fiber
☐	Plastic fiber
☐	F-SMA type connector
☐	BFOC/2,5 type connector

8.7.4 Transmission speed

☒	9 600 bit/s
☒	19 200 bit/s

8.7.5 Link Layer

There are no choices for the link layer.

8.7.6 Application Layer

8.7.7 Transmission mode for application data

Mode 1 (least significant octet first), as defined in 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

8.7.8 Common Address of ASDU

☒	One Common Address of ASDU (identical with station address)
☒	More than one Common Address of ASDU

8.7.9 Selection of standard information numbers in monitor direction

8.7.10 System functions in monitor direction

	INF	Semantics
☒	<0>	End of general interrogation
☒	<0>	Time synchronization
☒	<2>	Reset FCB
☒	<3>	Reset CU
☒	<4>	Start/restart
☒	<5>	Power on

8.7.11 Status indications in monitor direction

	INF	Semantics
☒	<16>	Auto-recloser active
☒	<17>	Teleprotection active

	INF	Semantics
☒	<18>	Protection active
☒	<19>	LED reset
☒	<20>	Monitor direction blocked
☒	<21>	Test mode
☒	<22>	Local parameter setting
☒	<23>	Characteristic 1
☒	<24>	Characteristic 2
☒	<25>	Characteristic 3
☒	<26>	Characteristic 4
☒	<27>	Auxiliary input 1
☒	<28>	Auxiliary input 2
☒	<29>	Auxiliary input 3
☒	<30>	Auxiliary input 4

8.7.12 Supervision indications in monitor direction

	INF	Semantics
☒	<32>	Measurand supervision I
☒	<33>	Measurand supervision V
☒	<35>	Phase sequence supervision
☒	<36>	Trip circuit supervision
☒	<37>	I>> back-up operation
☒	<38>	VT fuse failure
☒	<39>	Teleprotection disturbed
☒	<46>	Group warning
☒	<47>	Group alarm

8.7.13 Earth fault indications in monitor direction

	INF	Semantics
☒	<48>	Earth fault L_1
☒	<49>	Earth fault L_2
☒	<50>	Earth fault L_3
☒	<51>	Earth fault forward, that is, line
☒	<52>	Earth fault reverse, that is, busbar

8.7.14 Fault indications in monitor direction

	INF	Semantics
☒	<64>	Start /pick-up L_1
☒	<65>	Start /pick-up L_2
☒	<66>	Start /pick-up L_3
☒	<67>	Start /pick-up N

	INF	Semantics
☒	<68>	General trip
☒	<69>	Trip L_1
☒	<70>	Trip L_2
☒	<71>	Trip L_3
☒	<72>	Trip I>> (back-up operation)
☒	<73>	Fault location X in ohms
☒	<74>	Fault forward/line
☒	<75>	Fault reverse/busbar
☒	<76>	Teleprotection signal transmitted
☒	<77>	Teleprotection signal received
☒	<78>	Zone 1
☒	<79>	Zone 2
☒	<80>	Zone 3
☒	<81>	Zone 4
☒	<82>	Zone 5
☒	<83>	Zone 6
☒	<84>	General start/pick-up
☒	<85>	Breaker failure
☒	<86>	Trip measuring system L_1
☒	<87>	Trip measuring system L_2
☒	<88>	Trip measuring system L_3
☒	<89>	Trip measuring system E
☒	<90>	Trip I>
☒	<91>	Trip I>>
☒	<92>	Trip IN>
☒	<93>	Trip IN>>

8.7.15 Auto-reclosure indications in monitor direction

	INF	Semantics
☒	<128>	CB "on" by AR
☒	<129>	CB "on" by long-time AR
☒	<130>	AR blocked

8.7.16 Measurands in monitor direction

	INF	Semantics
☒	<144>	Measurand I
☒	<145>	Measurands I, V
☒	<146>	Measurands I, V, P, Q
☒	<147>	Measurands I_N, V_{EN}
☒	<148>	Measurands $I_{L1,2,3}$, $V_{L1,2,3}$, P, Q, f

8.7.17 Generic functions in monitor direction

	INF	Semantics
☐	<240>	Read headings of all defined groups
☐	<241>	Read values or attributes of all entries of one group
☐	<243>	Read directory of a single entry
☐	<244>	Read value or attribute of a single entry
☐	<245>	End of general interrogation of generic data
☐	<249>	Write entry with confirmation
☐	<250>	Write entry with execution
☐	<251>	Write entry aborted

8.7.18 Selection of standard information numbers in control direction

8.7.19 System functions in control direction

	INF	Semantics
☒	<0>	Initiation of general interrogation
☒	<0>	Time synchronization

8.7.20 General commands in control direction

	INF	Semantics
☒	<16>	Auto-recloser on/off
☒	<17>	Teleprotection on/off
☒	<18>	Protection on/off
☒	<19>	LED reset
☒	<23>	Activate characteristic 1
☒	<24>	Activate characteristic 2
☒	<25>	Activate characteristic 3
☒	<26>	Activate characteristic 4

8.7.21 Generic functions in control direction

	INF	Semantics
☐	<240>	Read headings of all defined groups
☐	<241>	Read values or attributes of all entries in one group
☐	<243>	Read directory of a single entry
☐	<244>	Read value or attribute of a single entry
☐	<245>	General interrogation of generic data
☐	<248>	Write entry
☐	<249>	Write entry with confirmation
☐	<250>	Write entry with execution
☐	<251>	Write entry abort

8.7.22 Basic application functions

☒	Test mode
☒	Blocking of monitor direction
☐	Disturbance data
☐	Generic services
☒	Private data (if ASDU type is supported by module)

8.7.23 Miscellaneous

Measurands are transmitted with ASDU 3 as well as with ASDU 9. As defined in 7.2.6.8, the maximum MVAL can either be 1,2 or 2,4 times the rated value. No different rating shall be used in ASDU 3 and ASDU 9, that is, for each measurand there is only one choice.

Measurand	Max. MVAL = rated value times		
	1,2	**or**	**2,4**
Current L_1	☒		☒
Current L_2	☒		☒
Current L_3	☒		☒
Voltage L_{1-E}	☒		☒
Voltage L_{2-E}	☒		☒
Voltage L_{3-E}	☒		☒
Active power P	☒		☒
Reactive power Q	☒		☒
Frequency f	☒		☒
Voltage L_1 - L_2	☒		☒

8.8 Form to Define Sector Database

Session Index #:

Sector Index #:

Data Type (1, 2, 3, 4, 5 or 9)	Database Address	Function Code	Point Index

Session Index #:

Sector Index #:

Data Type (1, 2, 3, 4, 5 or 9)	**Database Address**	**Function Code**	**Point Index**

8.9 Form to Define Command List

Enable Code	Database Index	Poll Interval	Session Index	Sector Index	Data Type (0,6,7,20)	Function Code	Point Index	Override Flag	Override Value

Enable Code	Database Index	Poll Interval	Session Index	Sector Index	Data Type (0,6,7,20)	Function Code	Point Index	Override Flag	Override Value

8.10 Frequently Asked Questions

8.10.1 How fast do the "Backplane Data Exchange" commands run?

The "Backplane Data Exchange" commands will execute one at a time during the I/O service interval of the PLC. What this means is that if you had a list of 10 commands at the end of every PLC scan one command would execute. This would mean that it would take 10 PLC scans to execute the 10 commands contained within the "Backplane Data Exchange" section of the configuration file.

8.10.2 What is the maximum number of words I can transfer with a "Backplane Data Exchange" command?

For command types 1 & 2 you may move up to 130 words with each command. Function 3 is somewhat different in that it provides only 64 words of data movement BUT because it is intended to solve very specialized operations its size must be restricted.

8.10.3 Do I need to use "Backplane Data Exchange" function 3?

The only time you should need it is if you are using the DNP, DNP or one of the IEC protocols. If you are using one of these protocols then you can find sample structured text examples included in the manual for these protocols. In all other instances you should not need to use this function.

8.10.4 How much data can I transfer between the PLC and the Module.

You can enter up to 100 commands in the [BACKPLANE DATA EXCHANGE] section of the configuration file. The limit for any single execution of a Function 1 or 2 is 130 words but you may enter multiple commands to transfer more data.

8.10.5 How do I configure the module?

The ProTalk requires a simple text based configuration file to make it operational. For a really quick tutorial on the modules communications with the PLC you should review the [QUICK START GUIDE] or for more in depth information the chapter on "Backplane Data Exchange" should answer most questions.

8.10.6 What kind of data transfer rates can I expect between the PLC and the module?

Data transfer rates between the PLC and the module depend on a number of variables, among them the number of words being transferred per command, the amount of other network traffic at the time data is being transferred, and overall processor scan times.

8.10.7 What software application is required for my Ladder Logic?

The design of the module should be software independent and for many installations minimal or possibly no ladder will be required. The Backplane Data Exchange section offers two samples to help in the few instances where ladder is required.

8.10.8 Is a .MDC available for configuration of the Module?

Yes. The CDROM that ships with the module should have a version for both Concept 2.5 and 2.6 in the ProTalk directory.

8.10.9 Does the module work in a remote rack?

The module is designed to be located in the chassis with the PLC and will not operate in a remote chassis. If your application requires remote placement of the communication device you should investigate the other members of the ProSoft Technology. family such as the 4202-MNET-DFCM (if you require DF1 connectivity for example although many others are available). This module for example would allow you to communicate with DF1 devices and allow you to map the contents of its memory using Modbus TCP/IP.

8.10.10 Can I use the module in a hot backup system?

Support for Hot Backup is not currently implemented in the module. We are currently investigating the addition of this functionality but until this development can be finalized, it may be possible to use one of the 4000 series of ProSoft communication products. Please call our technical support technicians when considering this application.

9 Support, Service & Warranty

In This Chapter

ProSoft Technology, Inc. (ProSoft) is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

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

If the issue is hardware related, we will also need information regarding:

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

9.1 How to Contact Us: Technical Support

Internet	Web Site: www.prosoft-technology.com/support
	E-mail address: support@prosoft-technology.com

Asia Pacific

+603.7724.2080, support.asia@prosoft-technology.com
Languages spoken include: Chinese, English

Europe (location in Toulouse, France)

+33 (0) 5.34.36.87.20, support.EMEA@prosoft-technology.com
Languages spoken include: French, English

North America/Latin America (excluding Brasil) (location in California)

+1.661.716.5100, support@prosoft-technology.com
Languages spoken include: English, Spanish
*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.*

Brasil (location in Sao Paulo)

+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 (collectively, "RMA Policies") apply to any returned Product. These RMA Policies are subject to change by ProSoft without notice. For warranty information, see "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:

a) 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.
b) 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 (page 171). A Technical Support Engineer will request that you perform several tests in an attempt to isolate the problem. If after completing these tests, the Product is found to be the source of the problem, we will issue an RMA.
c) 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 and receipt date. 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 using a shipment method other than that specified by ProSoft or shipped without an RMA number will be returned to the Customer, freight collect. Contact ProSoft Technical Support for further information.
d) A 10% restocking fee applies to all warranty credit returns whereby a Customer has an application change, ordered too many, does not need, and so on.

9.2.2 Procedures for Return of Units Under Warranty:

A Technical Support Engineer must approve the return of Product under ProSoft's Warranty:

a) A replacement module will be shipped and invoiced. A purchase order will be required.
b) Credit for a product under warranty will be issued upon receipt of authorized product by ProSoft at designated location referenced on the Return Material Authorization.

9.2.3 Procedures for Return of Units Out of Warranty:

a) Customer sends unit in for evaluation
b) If no defect is found, Customer will be charged the equivalent of $100 USD, plus freight charges, duties and taxes as applicable. A new purchase order will be required.

c) If unit is repaired, charge to Customer will be 30% of current list price (USD) plus freight charges, duties and taxes as applicable. A new purchase order will be required or authorization to use the purchase order submitted for evaluation fee.

The following is a list of non-repairable units:

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

9.2.4 Purchasing Warranty Extension:

a) ProSoft's standard warranty period is three (3) years from the date of shipment as detailed in "Limited Warranty (page 173)". The Warranty Period may be extended at the time of equipment purchase 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

9.3 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.

9.3.1 What Is Covered By This Warranty

a) *Warranty On New Products*: ProSoft warrants, to the original purchaser, 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 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 39 months. 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.
b) *Warranty On Services*: Materials and labor performed by ProSoft to repair a verified malfunction or defect are warranteed in 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.

9.3.2 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) This Warranty 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, "C" or any variant of "C" programming languages) 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; or (viii) disasters such as fire, flood, earthquake, wind and lightning.

c) 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 guide included with your original product purchase 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.

9.3.3 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 and 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.

9.3.4 Intellectual Property Indemnity

Buyer shall indemnify and hold harmless ProSoft and its employees from and against all liabilities, losses, claims, costs and expenses (including attorney's fees and expenses) related to any claim, investigation, litigation or proceeding (whether or not ProSoft is a party) which arises or is alleged to arise from Buyer's acts or omissions under these Terms or in any way with respect to the Products. Without limiting the foregoing, Buyer (at its own expense) shall indemnify and hold harmless ProSoft and defend or settle any action brought against such Companies to the extent based on a claim that any Product made to Buyer specifications infringed intellectual property rights of another party. ProSoft makes no warranty that the product is or will be delivered free of any person's claiming of patent, trademark, or similar infringement. The Buyer assumes all risks (including the risk of suit) that the product or any use of the product will infringe existing or subsequently issued patents, trademarks, or copyrights.

a) Any documentation included with Product purchased from ProSoft is protected by copyright and may not be duplicated 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.

f) **Additional Restrictions Relating To Software And Other Intellectual Property**

   In addition to compliance 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.

9.3.5 Disclaimer of all Other Warranties

The Warranty set forth in What Is Covered By This Warranty (page 174) 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.

*9.3.6 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 or its Dealer will not be responsible for included, 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.

9.3.7 Time Limit for Bringing Suit

Any action for breach of warranty must be commenced within 39 months following shipment of the Product.

9.3.8 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.

9.3.9 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.

9.3.10 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

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y