ProTalk

PTQ-DNPS

Quantum / Unity Platform

Distributed Network Protocol Interface Module

Application Reference Guide

April 22, 2008

ProSoft
TECHNOLOGY

Please Read This Notice

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

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

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

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

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

1.1 PTQ Installation and Operating Instructions

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

The following or equivalent warnings shall be included:

A Warning - Explosion Hazard - Substitution of components may Impair Suitability for Class I, Division 2;

B Warning - Explosion Hazard - When in Hazardous Locations, Turn off Power before replacing Wiring Modules, and

C Warning - Explosion Hazard - Do not Disconnect Equipment unless Power has been switched Off or the Area is known to be Nonhazardous.

D Caution: The Cell used in this Device may Present a Fire or Chemical Burn Hazard if Mistreated. Do not Disassemble, Heat above 100°C (212°F) or Incinerate.

Important Notice:

⚠

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

Maximum battery load = 200 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
1675 Chester Avenue, Fourth Floor
Bakersfield, CA 93301
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
http://www.prosoft-technology.com

PTQ-DNPS Application Reference Guide
April 22, 2008
PSFT.DNPS.PTQ.UM.08.04.22

Contents

Guide to the PTQ-DNPS Application Reference Guide

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 11)	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 87) Diagnostics and Troubleshooting (page 77)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 91) Functional Overview (page 94) Product Specifications (page 91)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 141)	This section contains Support, Service and Warranty information. Index of chapters.

2 Start Here

In This Chapter

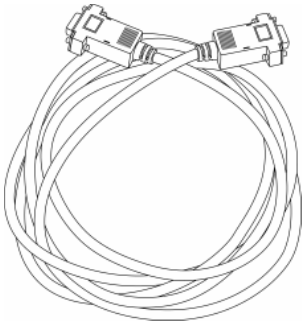
This guide is intended to guide you through the ProTalk module setup process, from removing the module from the box to exchanging data with the processor. In doing this, you will learn how to:

- 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

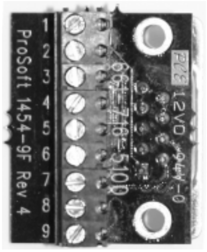
2.1 Hardware and Software Requirements

2.1.1 ProTalk Module Carton Contents

ProTalk Module



Null Modem Serial Cable

 ProSoft 1454-9F Rev 4 1 2 3 4 5 6 7 8 9	
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.

2.1.2 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

2.1.3 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**.

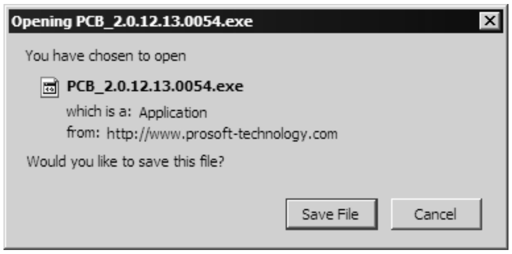
Note: ProTalk modules are compatible with common Quantum / Unity programming applications, including Concept and UnityPro XL. For all other programming applications, please contact technical support.

2.2 Install ProSoft Configuration Builder Software

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

To install ProSoft Configuration Builder from the ProSoft Web Site

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



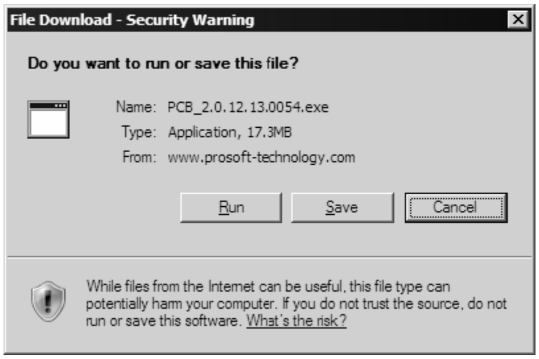
Opening PCB_2.0.12.13.0054.exe

You have chosen to open

PCB_2.0.12.13.0054.exe

which is a: Application
from: http://www.prosoft-technology.com

Would you like to save this file?



File Download - Security Warning

Do you want to run or save this file?

Name: PCB_2.0.12.13.0054.exe
Type: Application, 17.3MB
From: www.prosoft-technology.com

While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?

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

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

To install ProSoft Configuration Builder from the CD-ROM

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

3 Click to open the *Utilities* folder. This folder contains all of the applications and files you will need to set up and configure your module.

4 Double-click the *ProSoft Configuration Builder Setup* program and follow the instructions on your screen to install the software on your PC.

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

3 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 take this into account when monitoring the status of the PTQ module.

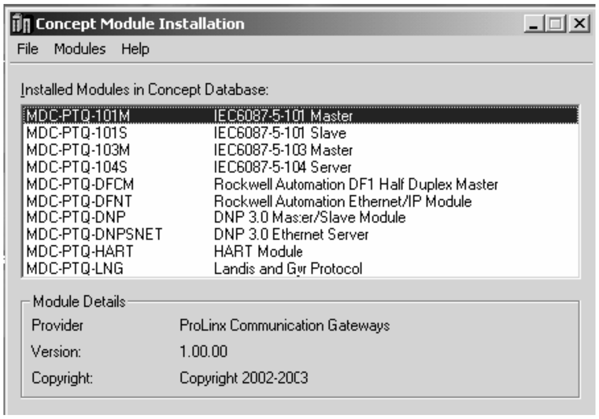
3.1 Information for Concept Version 2.6 Users

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.

3.1.1 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.



Installed Modules in Concept Database:

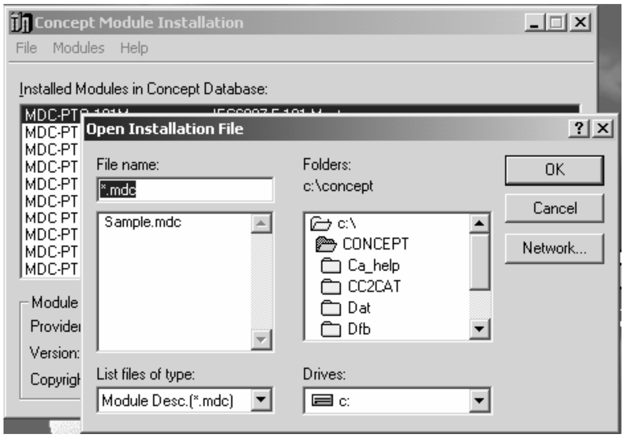
MDC-PTQ-101M	IEC6087-5-101 Master
MDC-PTQ-101S	IEC6087-5-101 Slave
MDC-PTQ-103M	IEC6087-5-103 Master
MDC-PTQ-104S	IEC6087-5-104 Server
MDC-PTQ-DFCM	Rockwell Automation DF1 Half Duplex Master
MDC-PTQ-DFNT	Rockwell Automation Ethernet/IP Module
MDC-PTQ-DNP	DNP 3.0 Master/Slave Module
MDC-PTQ-DNPSNET	DNP 3.0 Ethernet Server
MDC-PTQ-HART	HART Module
MDC-PTQ-LNG	Landis and Gyr Protocol

Module Details

Provider	ProLinx Communication Gateways
Version:	1.00.00
Copyright:	Copyright 2002-2003

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

This action opens the Open Installation File dialog box:


Concept Module Installation
File Modules Help
Installed Modules in Concept Database:
Open Installation File
OK
Cancel
Network...

File name: *.mdc

Sample.mdc

Folders: c:\concept

c:\
CONCEPT
Ca_help
CC2CAT
Dat
Dfb

List files of type: Module Desc.(*.mdc)

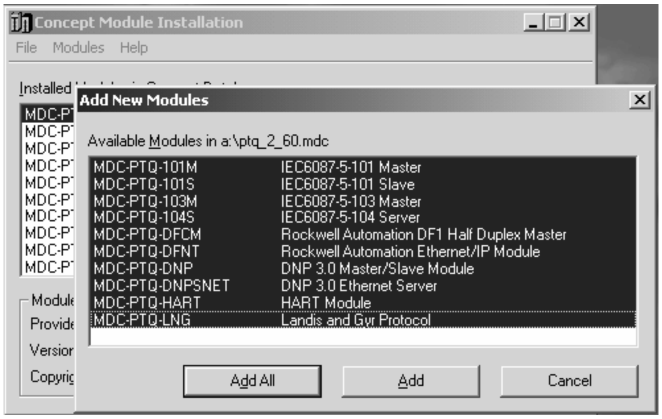
Drives: c:

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.



Available Modules in a:\ptq_2_60.mdc

MDC-PTQ-101M	IEC6087-5-101 Master
MDC-PTQ-101S	IEC6087-5-101 Slave
MDC-PTQ-103M	IEC6087-5-103 Master
MDC-PTQ-104S	IEC6087-5-104 Server
MDC-PTQ-DFCM	Rockwell Automation DF1 Half Duplex Master
MDC-PTQ-DFNT	Rockwell Automation Ethernet/IP Module
MDC-PTQ-DNP	DNP 3.0 Master/Slave Module
MDC-PTQ-DNPSNET	DNP 3.0 Ethernet Server
MDC-PTQ-HART	HART Module
MDC-PTQ-LNG	Landis and Gyr Protocol

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.

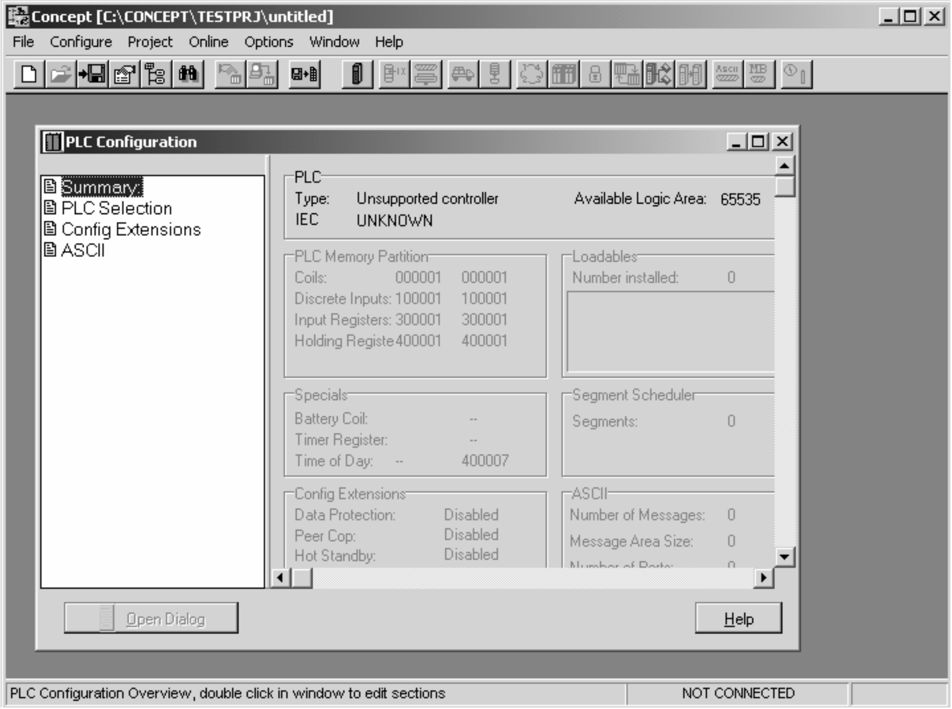
6 When the process is complete, open the File menu and choose Exit to save your changes.

3.2 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.


Concept [C:\CONCEPT\TESTPRJ\untitled]
File Configure Project Online Options Window Help
PLC Configuration
Summary
PLC Selection
Config Extensions
ASCII
Open Dialog
Help

PLC
Type: Unsupported controller   Available Logic Area: 65535
IEC UNKNOWN

PLC Memory Partition
Coils: 000001 000001
Discrete Inputs: 100001 100001
Input Registers: 300001 300001
Holding Registe 400001 400001

Loadables
Number installed: 0

Specials
Battery Coil: --
Timer Register: --
Time of Day: -- 400007

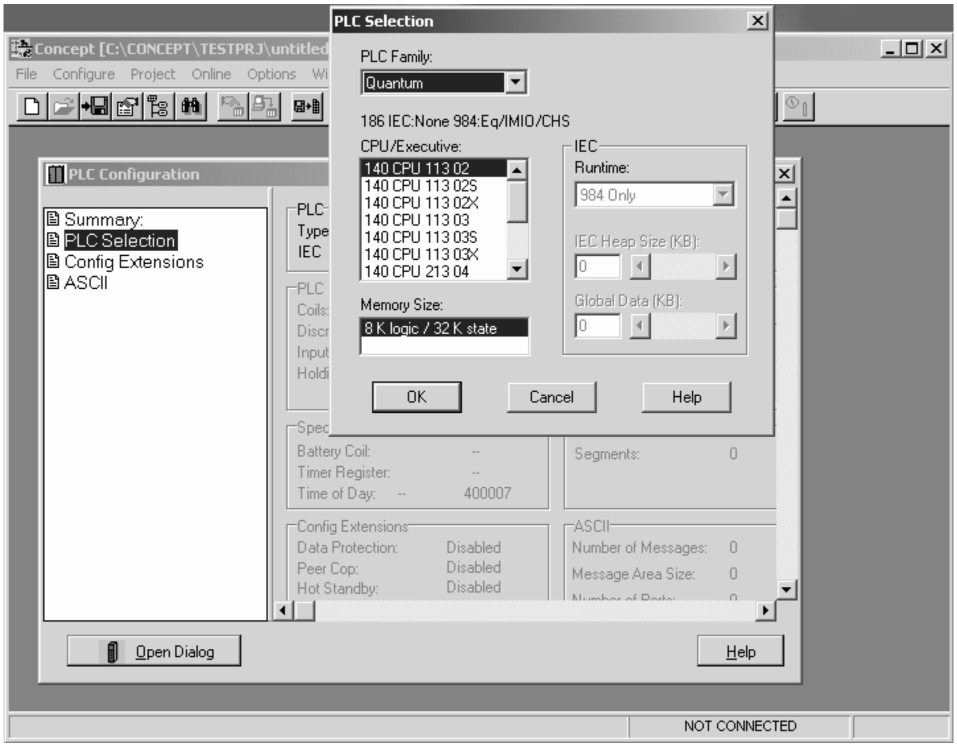
Segment Scheduler
Segments: 0

Config Extensions
Data Protection: Disabled
Peer Cop: Disabled
Hot Standby: Disabled

ASCII
Number of Messages: 0
Message Area Size: 0
Number of Ports: 0

PLC Configuration Overview, double click in window to edit sections   NOT CONNECTED

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.



PLC Selection

PLC Family: Quantum

186 IEC:None 984:Eq/IMIO/CHS

CPU/Executive:

140 CPU 113 02
140 CPU 113 02S
140 CPU 113 02X
140 CPU 113 03
140 CPU 113 03S
140 CPU 113 03X
140 CPU 213 04

Memory Size: 8 K logic / 32 K state

IEC

Runtime: 984 Only

IEC Heap Size (KB): 0

Global Data (KB): 0

Battery Coil: --

Timer Register: --

Time of Day: -- 400007

Segments: 0

Config Extensions

Data Protection: Disabled

Peer Cop: Disabled

Hot Standby: Disabled

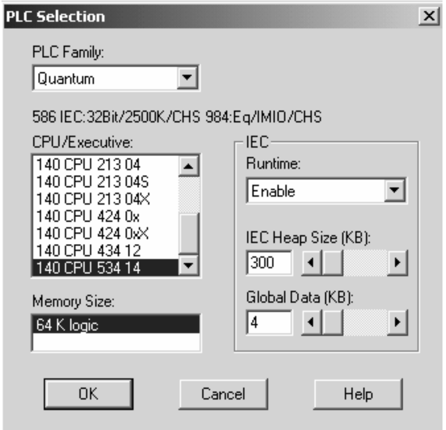
ASCII

Number of Messages: 0

Message Area Size: 0

NOT CONNECTED

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



PLC Selection

PLC Family: Quantum

586 IEC:32Bit/2500K/CHS 984:Eq/IMIO/CHS

CPU/Executive:

140 CPU 213 04
140 CPU 213 04S
140 CPU 213 04X
140 CPU 424 0x
140 CPU 424 0xX
140 CPU 434 12
140 CPU 534 14

Memory Size: 64 K logic

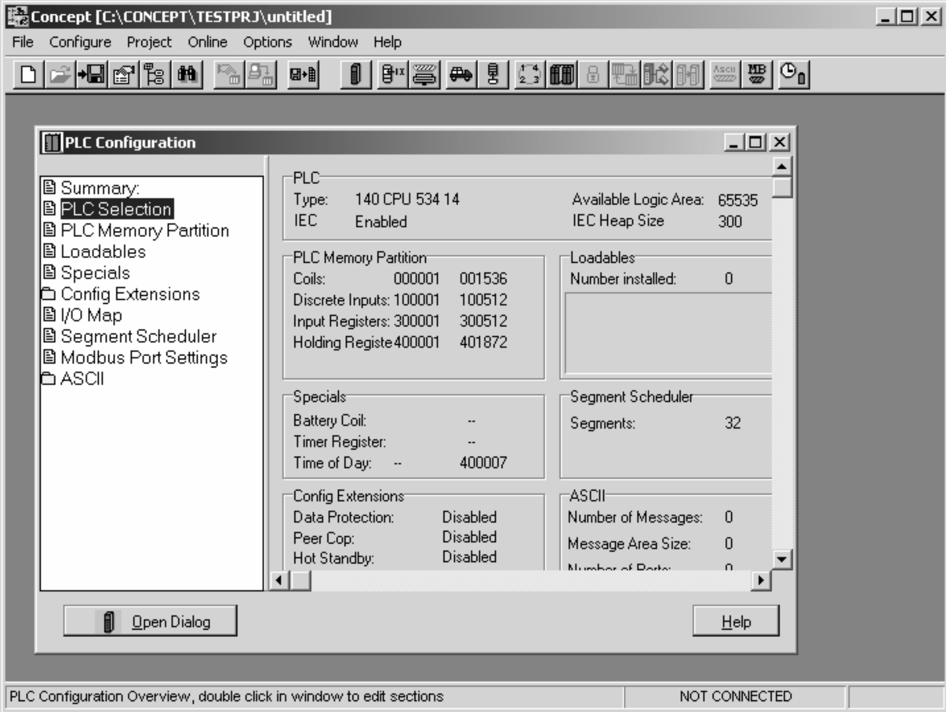
IEC

Runtime: Enable

IEC Heap Size (KB): 300

Global Data (KB): 4

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



PLC
Type: 140 CPU 534 14 Available Logic Area: 65535
IEC Enabled IEC Heap Size 300

PLC Memory Partition
Coils: 000001 001536
Discrete Inputs: 100001 100512
Input Registers: 300001 300512
Holding Registe 400001 401872

Loadables
Number installed: 0

Specials
Battery Coil: --
Timer Register: --
Time of Day: -- 400007

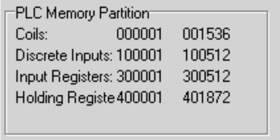
Segment Scheduler
Segments: 32

Config Extensions
Data Protection: Disabled
Peer Cop: Disabled
Hot Standby: Disabled

ASCII
Number of Messages: 0
Message Area Size: 0
Number of Ports: 0

PLC Configuration Overview, double click in window to edit sections NOT CONNECTED

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

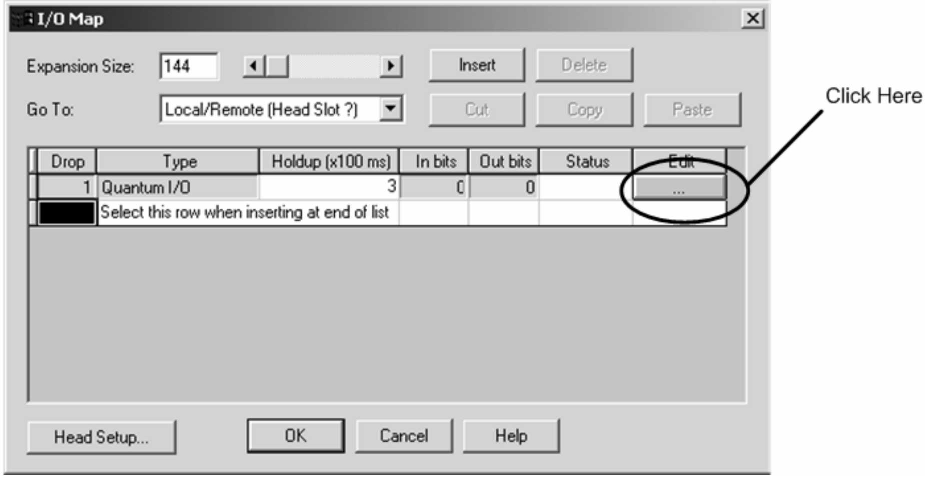


PLC Memory Partition
Coils: 000001 001536
Discrete Inputs: 100001 100512
Input Registers: 300001 300512
Holding Registe 400001 401872

3.3 Add the PTQ Module to the Project

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

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



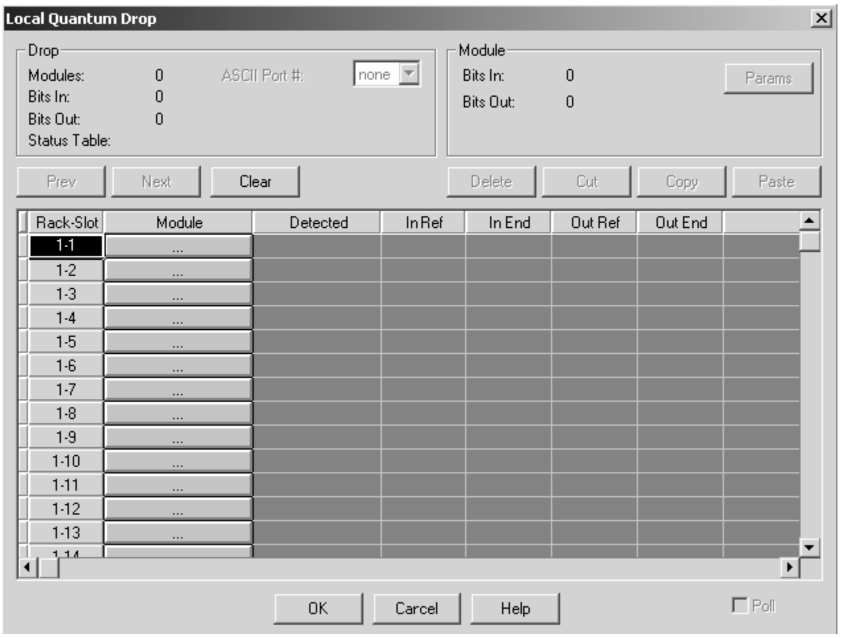
Expansion Size: 144

Go To: Local/Remote (Head Slot ?)

Drop	Type	Holdup (x100 ms)	In bits	Out bits	Status	Edit
1	Quantum I/O	3	0	0		...
	Select this row when inserting at end of list					

Click Here

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



Drop

Modules: 0

Bits In: 0

Bits Out: 0

Status Table:

ASCII Port #: none

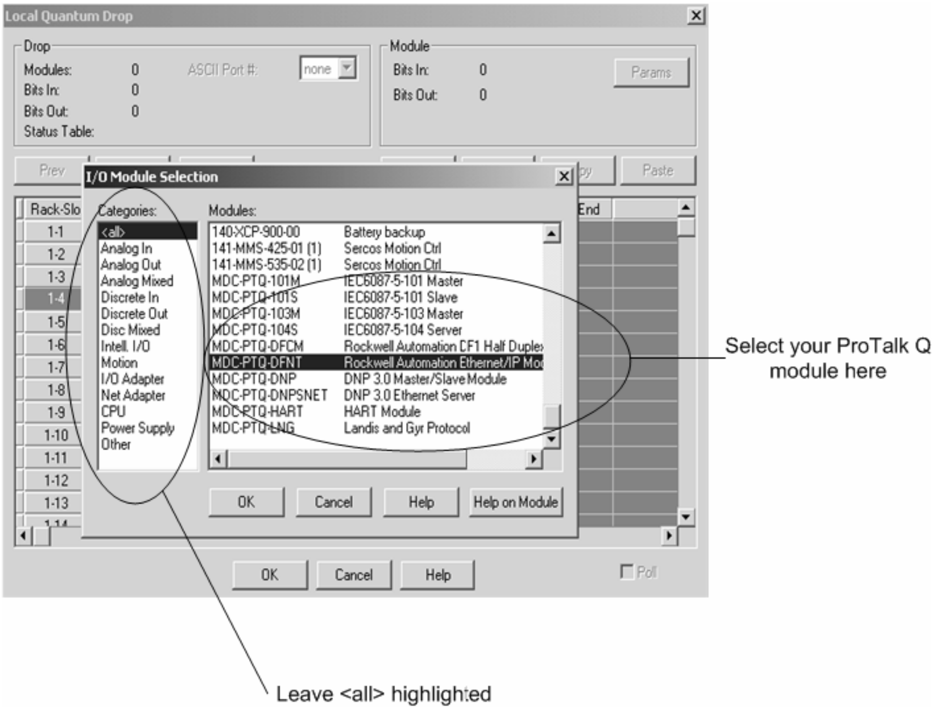
Module

Bits In: 0

Bits Out: 0

Rack-Slot	Module	Detected	In Ref	In End	Out Ref	Out End
1-1	...					
1-2	...					
1-3	...					
1-4	...					
1-5	...					
1-6	...					
1-7	...					
1-8	...					
1-9	...					
1-10	...					
1-11	...					
1-12	...					
1-13	...					
1-14						

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.


Local Quantum Drop

Drop
Modules: 0
Bits In: 0
Bits Out: 0
Status Table:
ASCII Port #: none

Module
Bits In: 0
Bits Out: 0

I/O Module Selection

Categories:
<all>
Analog In
Analog Out
Analog Mixed
Discrete In
Discrete Out
Disc Mixed
Intell. I/O
Motion
I/O Adapter
Net Adapter
CPU
Power Supply
Other

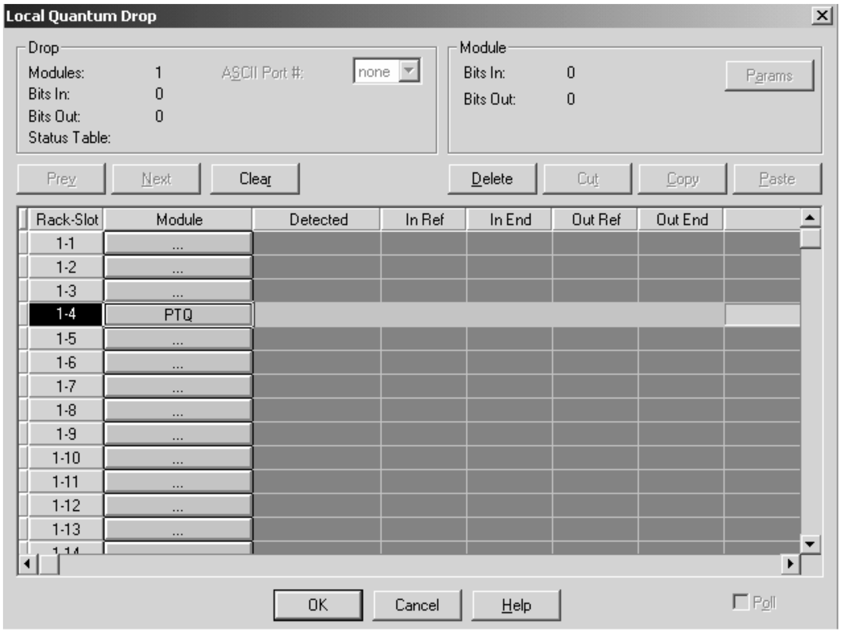
Modules:

140-XCP-900-00	Battery backup
141-MMS-425-01 (1)	Sercos Motion Ctrl
141-MMS-535-02 (1)	Sercos Motion Ctrl
MDC-PTQ-101M	IEC6087-5-101 Master
MDC-PTQ-101S	IEC6087-5-101 Slave
MDC-PTQ-103M	IEC6087-5-103 Master
MDC-PTQ-104S	IEC6087-5-104 Server
MDC-PTQ-DFCM	Rockwell Automation DF1 Half Duplex
MDC-PTQ-DFNT	Rockwell Automation Ethernet/IP Mod
MDC-PTQ-DNP	DNP 3.0 Master/Slave Module
MDC-PTQ-DNPSNET	DNP 3.0 Ethernet Server
MDC-PTQ-HART	HART Module
MDC-PTQ-LNG	Landis and Gyr Protocol

Select your ProTalk Q module here

Leave <all> highlighted

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.


Local Quantum Drop

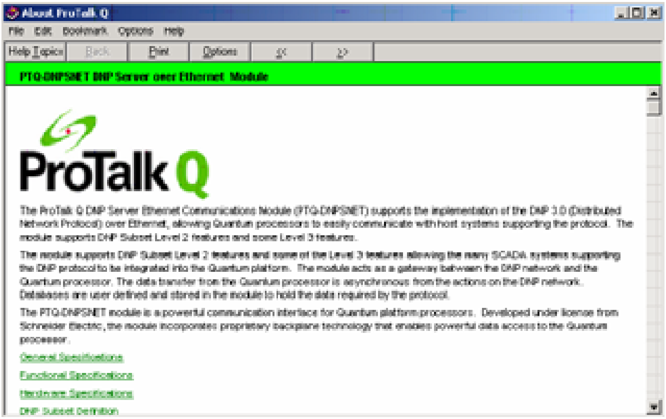
Drop
Modules: 1
Bits In: 0
Bits Out: 0
Status Table:
ASCII Port #: none

Module
Bits In: 0
Bits Out: 0

Rack-Slot	Module	Detected	In Ref	In End	Out Ref	Out End
1-1	...					
1-2	...					
1-3	...					
1-4	PTQ					
1-5	...					
1-6	...					
1-7	...					
1-8	...					
1-9	...					
1-10	...					
1-11	...					
1-12	...					
1-13	...					

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.



PTQ-DNPSNET DNP Server over Ethernet Module

The ProTalk Q DNP Server Ethernet Communications Module (PTQ-DNPSNET) supports the implementation of the DNP 3.0 (Distributed Network Protocol) over Ethernet, allowing Quantum processors to easily communicate with host systems supporting the protocol. The module supports DNP Subset Level 2 features and some Level 3 features.

The module supports DNP Subset Level 2 features and some of the Level 3 features allowing the many SCADA systems supporting the DNP protocol to be integrated into the Quantum platform. The module acts as a gateway between the DNP network and the Quantum processor. The data transfer from the Quantum processor is asynchronous from the actions on the DNP network. Databases are user defined and stored in the module to hold the data required by the protocol.

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

General Specifications

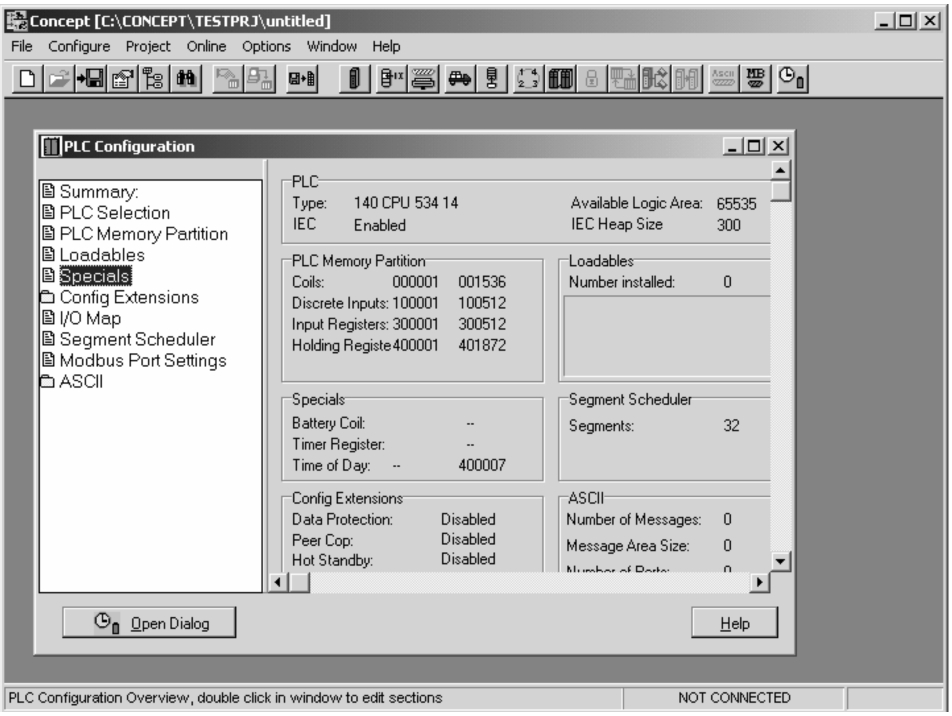
Functional Specifications

Hardware Specifications

DNP Subset Definition

3.4 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.



PLC
Type: 140 CPU 534 14
IEC Enabled
Available Logic Area: 65535
IEC Heap Size 300

PLC Memory Partition
Coils: 000001 001536
Discrete Inputs: 100001 100512
Input Registers: 300001 300512
Holding Registe 400001 401872

Loadables
Number installed: 0

Specials
Battery Coil: --
Timer Register: --
Time of Day: -- 400007

Segment Scheduler
Segments: 32

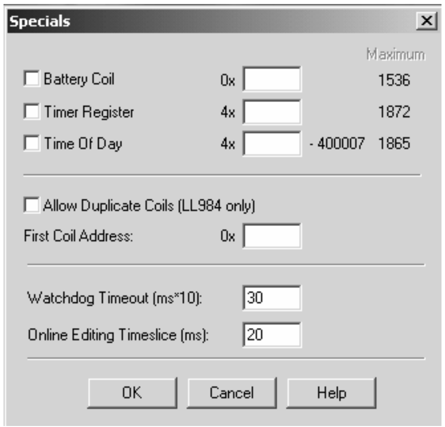
Config Extensions
Data Protection: Disabled
Peer Cop: Disabled
Hot Standby: Disabled

ASCII
Number of Messages: 0
Message Area Size: 0
Number of Ports: 0

PLC Configuration Overview, double click in window to edit sections

NOT CONNECTED

2 This action opens the Specials dialog box.



Specials

Maximum

☐ Battery Coil 0x 1536

☐ Timer Register 4x 1872

☐ Time Of Day 4x - 400007 1865

☐ Allow Duplicate Coils (LL984 only)

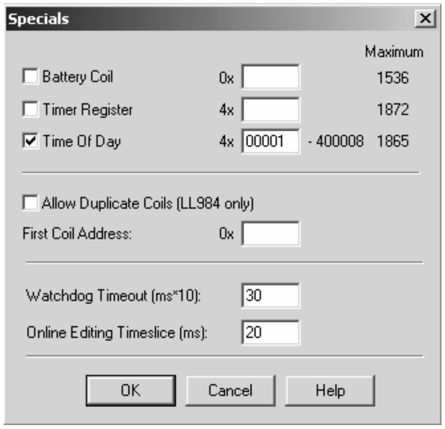
First Coil Address: 0x

Watchdog Timeout (ms*10): 30

Online Editing Timeslice (ms): 20

Selecting the Time of Day

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



Specials

Maximum

☐ Battery Coil 0x 1536

☐ Timer Register 4x 1872

☑ Time Of Day 4x 00001 - 400008 1865

☐ Allow Duplicate Coils (LL984 only)

First Coil Address: 0x

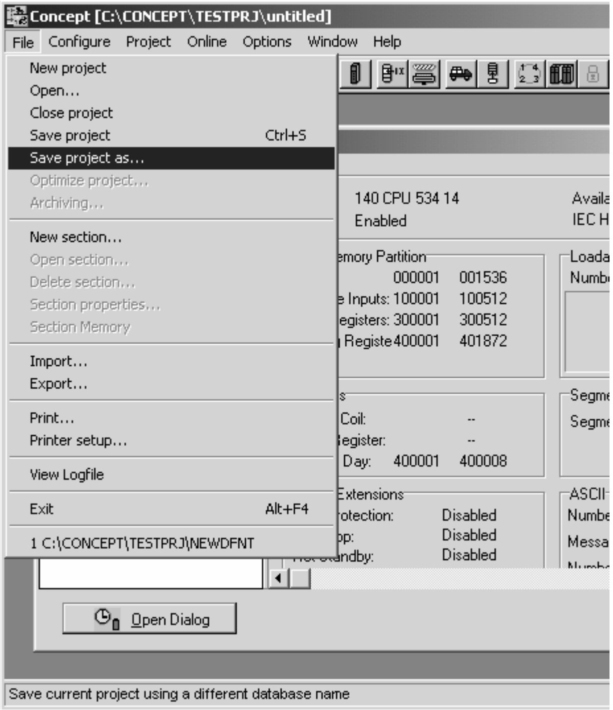
Watchdog Timeout (ms*10): 30

Online Editing Timeslice (ms): 20

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

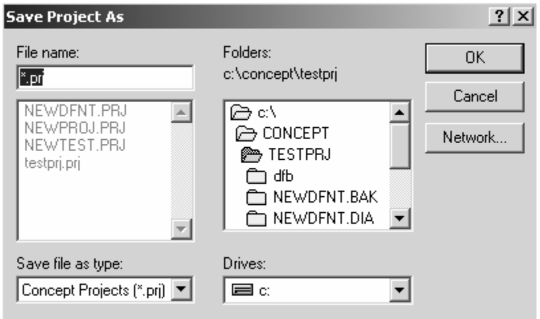
Saving your project

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



Save current project using a different database name

2 This action opens the Save Project as dialog box.



File name: *.prj

NEWDFNT.PRJ
NEWPROJ.PRJ
NEWTEST.PRJ
testprj.prj

Folders: c:\concept\testprj

c:\
CONCEPT
TESTPRJ
dfb
NEWDFNT.BAK
NEWDFNT.DIA

Save file as type: Concept Projects (*.prj)

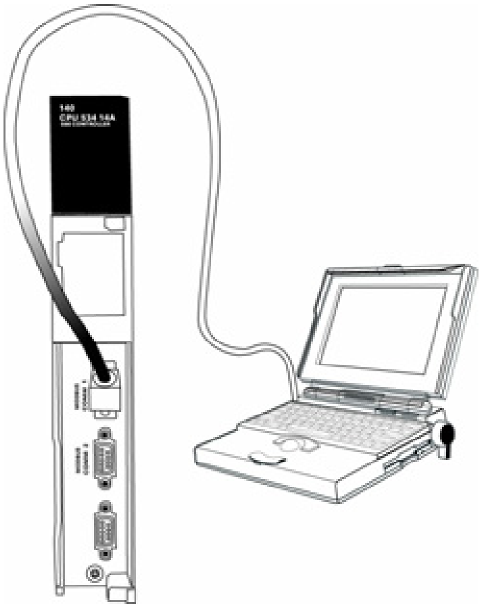
Drives: c:

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

3.5 Download the Project to the Processor

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

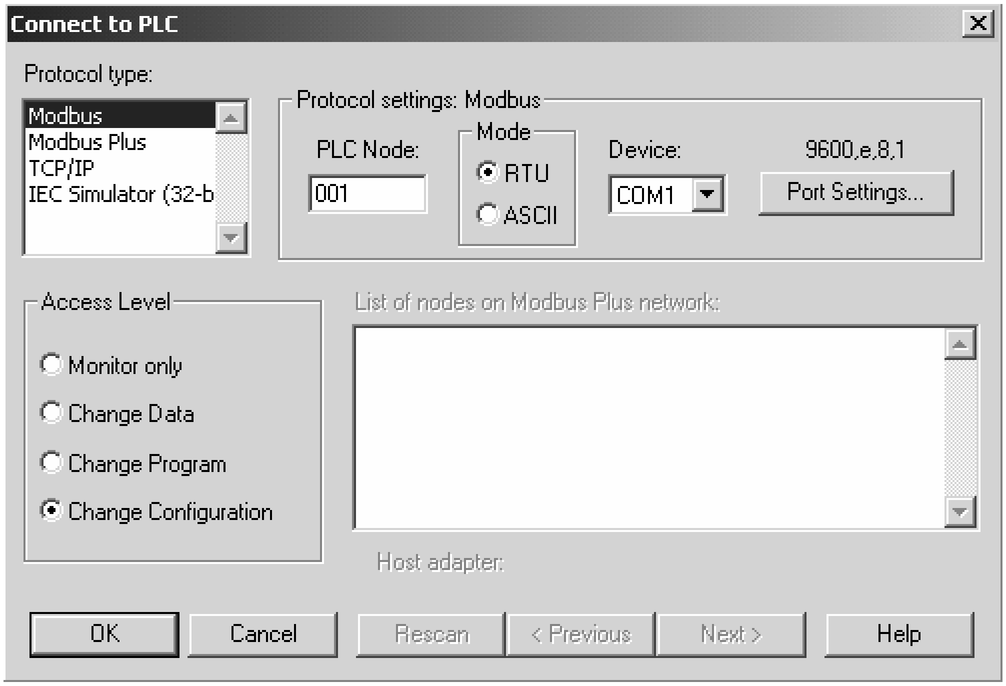
1 Use the null modem cable to connect your PC's serial port to the Quantum processor, as shown in the following illustration.



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

2 Open the PLC menu, and then choose Connect.

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



Protocol type: Modbus, Modbus Plus, TCP/IP, IEC Simulator (32-b

Protocol settings: Modbus

PLC Node: 001

Mode: ● RTU ○ ASCII

Device: COM1

9600,e,8,1

Access Level:
○ Monitor only
○ Change Data
○ Change Program
● Change Configuration

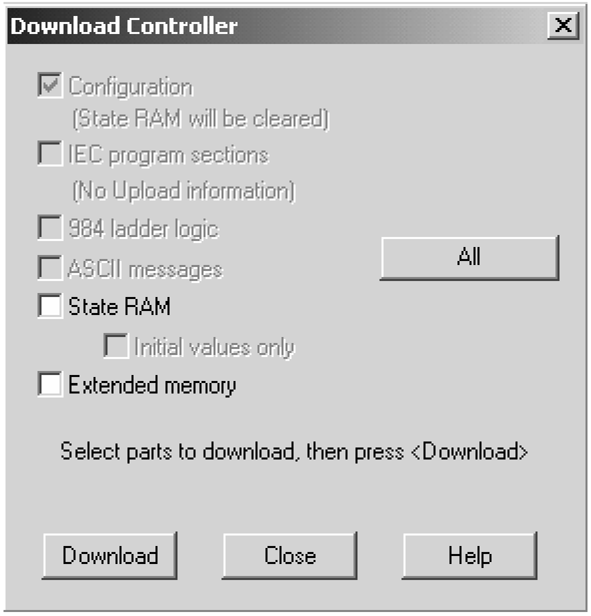
List of nodes on Modbus Plus network:

Host adapter:

4 Leave the default settings as shown and click OK.

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

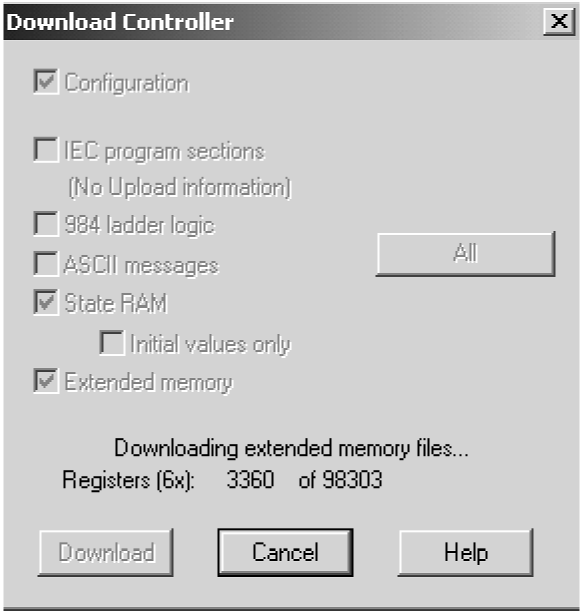
5 In the PLC Configuration window, open the Online menu, and then choose Download. This action opens the Download Controller dialog box.



☑ Configuration
(State RAM will be cleared)
☐ IEC program sections
(No Upload information)
☐ 984 ladder logic
☐ ASCII messages
☐ State RAM
☐ Initial values only
☐ Extended memory

Select parts to download, then press <Download>

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.



☑ Configuration

☐ IEC program sections
(No Upload information)

☐ 984 ladder logic

☐ ASCII messages

☑ State RAM

☐ Initial values only

☑ Extended memory

Downloading extended memory files...

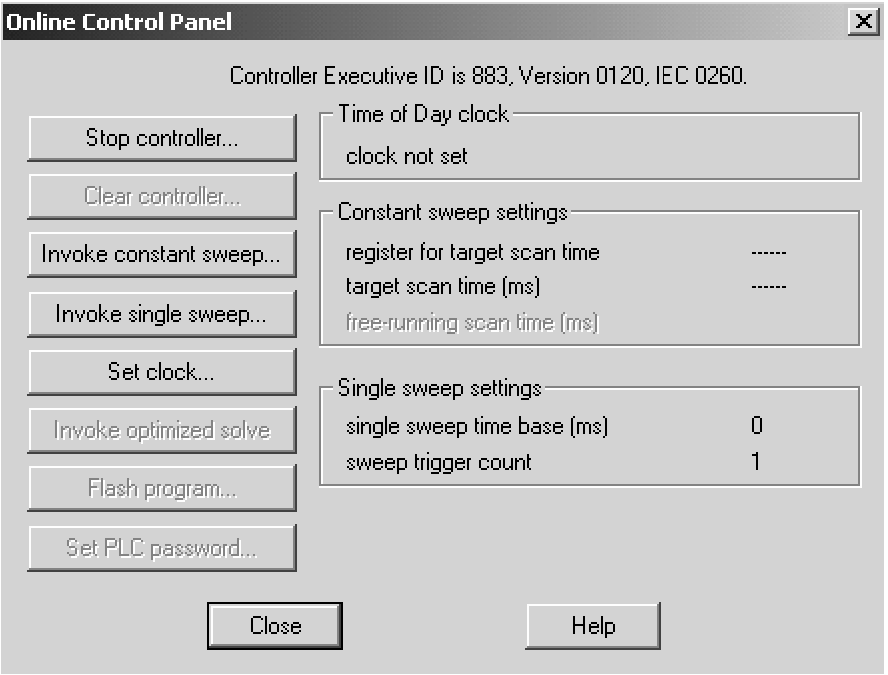
Registers (6x): 3360 of 98303

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

3.6 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.



Online Control Panel

Controller Executive ID is 883, Version 0120, IEC 0260.

Time of Day clock
clock not set

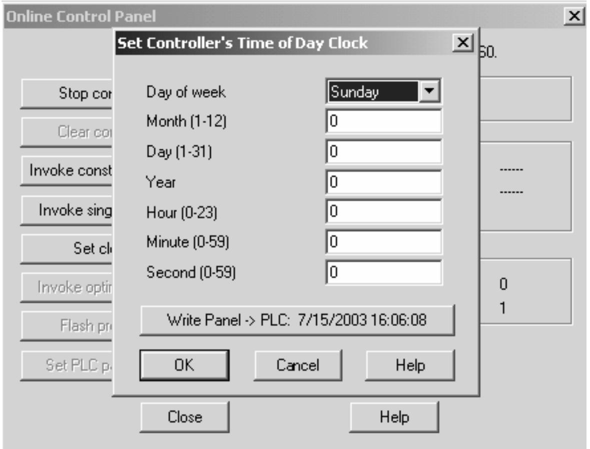
Constant sweep settings

register for target scan time	------
target scan time (ms)	------
free-running scan time (ms)	

Single sweep settings

single sweep time base (ms)	0
sweep trigger count	1

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



Set Controller's Time of Day Clock

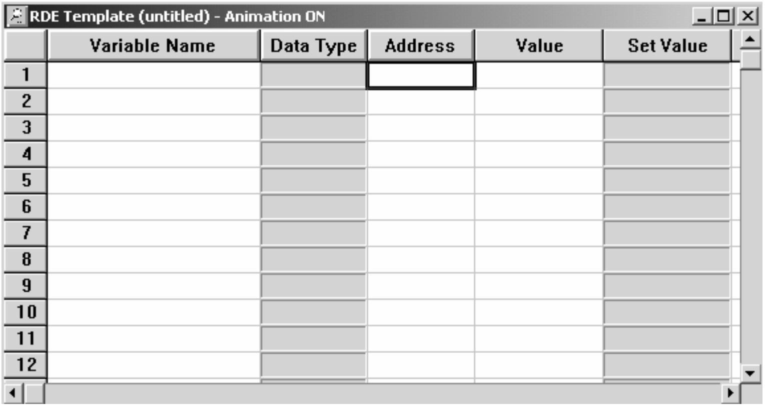
Day of week	Sunday
Month (1-12)	0
Day (1-31)	0
Year	0
Hour (0-23)	0
Minute (0-59)	0
Second (0-59)	0

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.

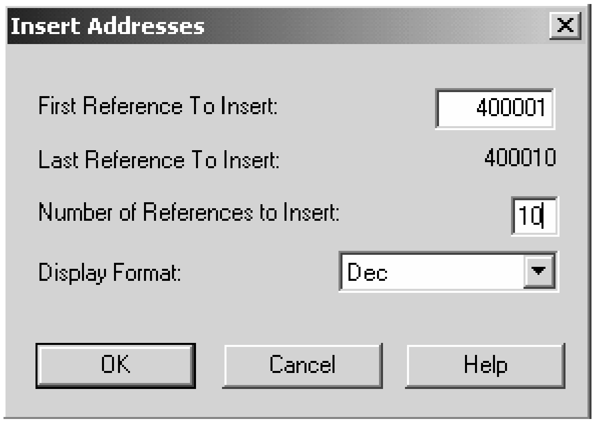


RDE Template (untitled) - Animation ON

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

7 In the PLC Configuration window, open the Templates menu, and then choose Insert addresses. This action opens the Insert addresses dialog box.

8 On the Insert addresses dialog box, enter the values shown in the following illustration, and then click OK.



Insert Addresses

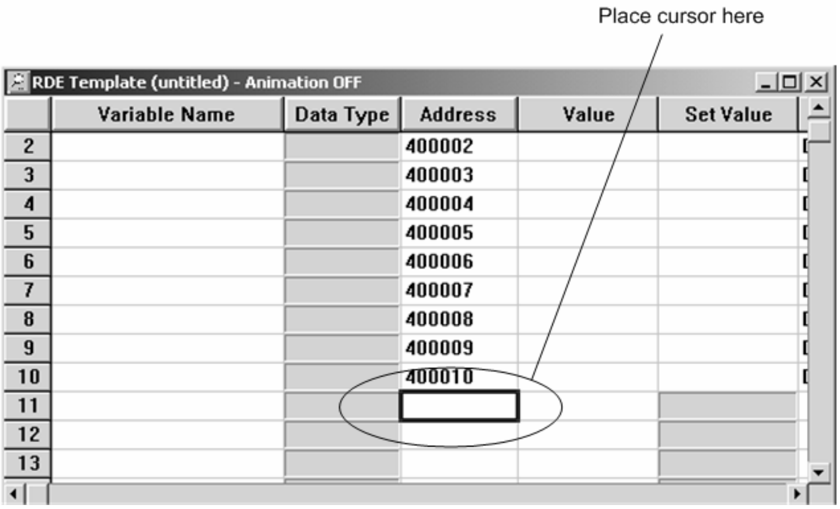
First Reference To Insert: 400001

Last Reference To Insert: 400010

Number of References to Insert: 10

Display Format: Dec

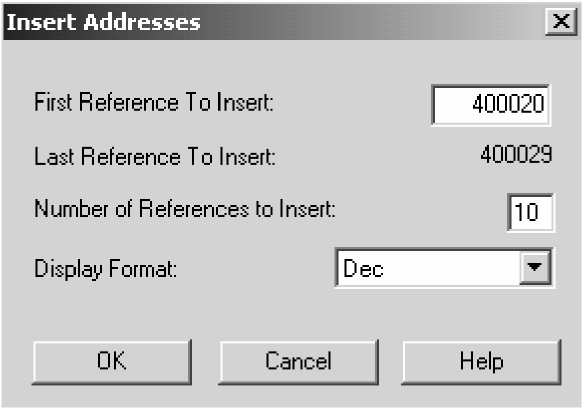
9 Notice that the template populates the address range, as shown in the following illustration. Place your cursor as shown in the first blank address field below the addresses you just entered.


Place cursor here

RDE Template (untitled) - Animation OFF

	Variable Name	Data Type	Address	Value	Set Value
2			400002		
3			400003		
4			400004		
5			400005		
6			400006		
7			400007		
8			400008		
9			400009		
10			400010		
11					
12					
13					

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


Insert Addresses

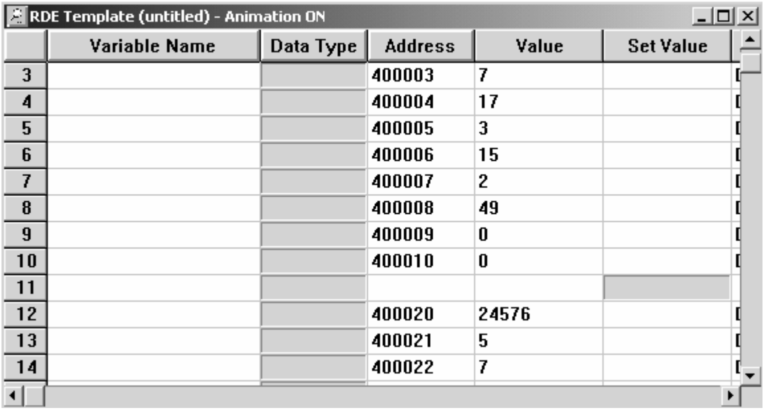
First Reference To Insert: 400020

Last Reference To Insert: 400029

Number of References to Insert: 10

Display Format: Dec

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.



RDE Template (untitled) - Animation ON

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

12 Verify that values shown are cycling, starting from address 400065 on up.

13 In the PLC Configuration window, open the Templates menu, and then choose Save Template as. Name the template ptqclock, and then click OK to save the template.

14 In the PLC Configuration window, open the Online menu, and then choose Disconnect. At the disconnect message, click Yes to confirm your choice.

At this point, you have successfully

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

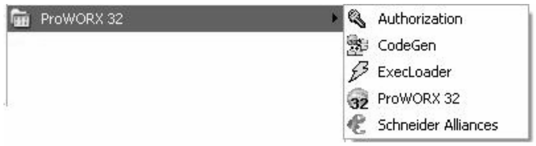
You are now ready to complete the installation and setup of the ProTalk module.

4 Configuring the Processor with ProWORX

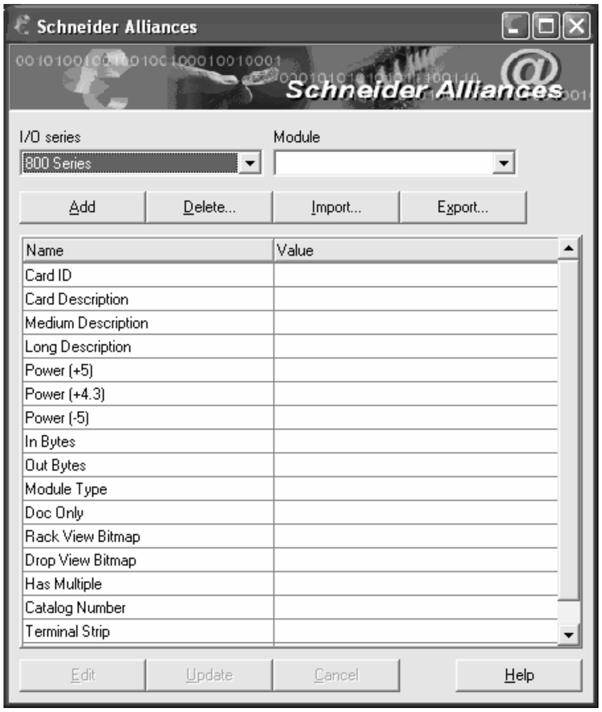
When you use ProWORX 32 software to configure the processor, use the example SaF file provided on the ProTalk Solutions CD-ROM.

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

1 Run the Schneider_alliances.exe application that is installed with the Proworx 32 software:

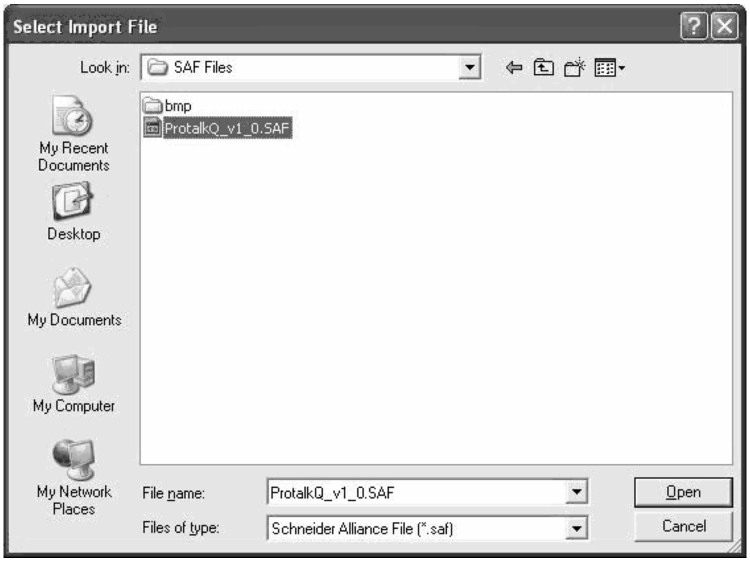


2 Click on Import…



Name	Value
Card ID	
Card Description	
Medium Description	
Long Description	
Power (+5)	
Power (+4.3)	
Power (-5)	
In Bytes	
Out Bytes	
Module Type	
Doc Only	
Rack View Bitmap	
Drop View Bitmap	
Has Multiple	
Catalog Number	
Terminal Strip	

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

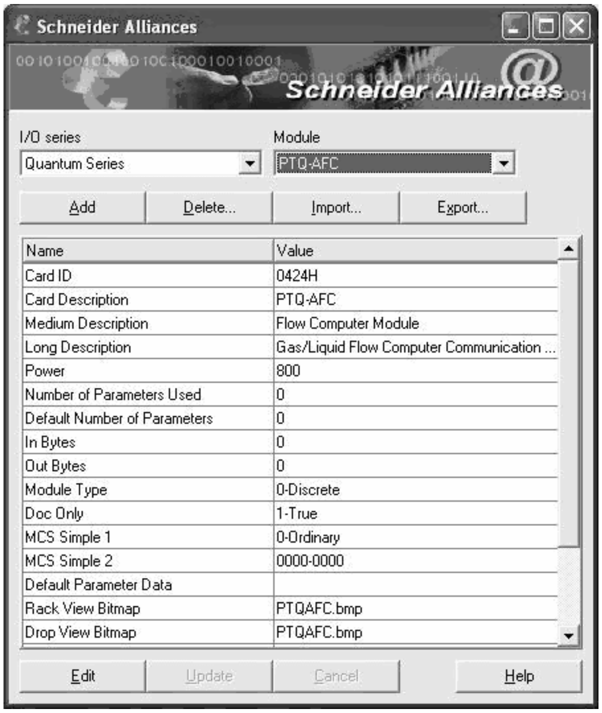


Look in: SAF Files

bmp
ProtalkQ_v1_0.SAF

File name: ProtalkQ_v1_0.SAF
Files of type: Schneider Alliance File (*.saf)

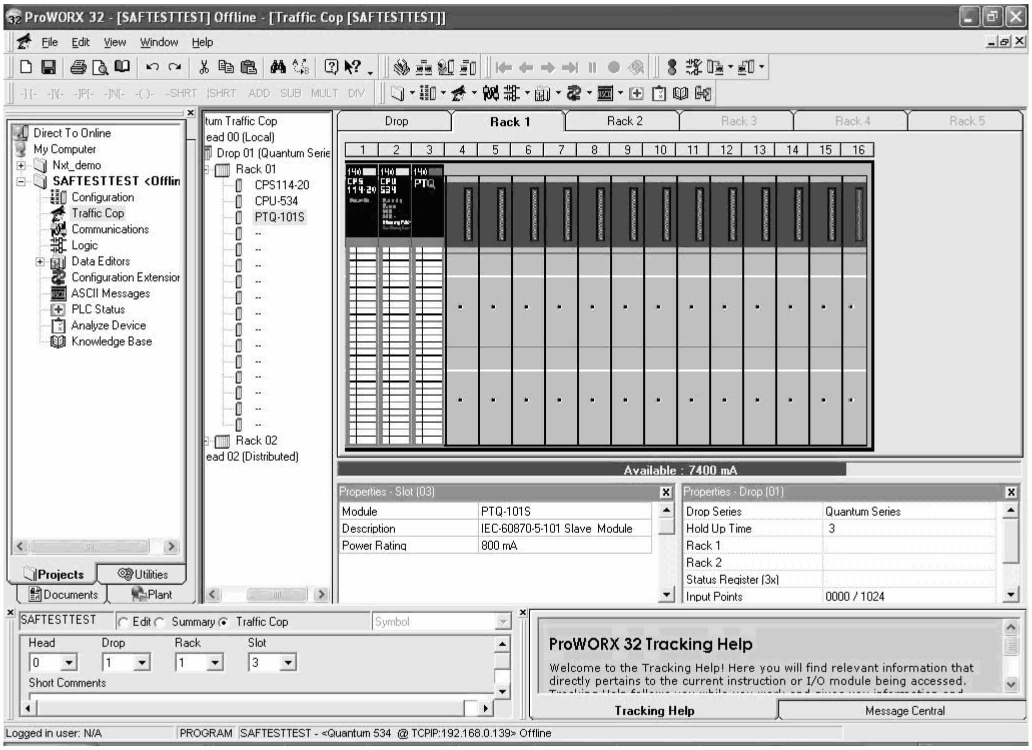
4 After you click on Open you should see the PTQ modules imported (select I/O series as Quantum):



I/O series: Quantum Series
Module: PTQ-AFC

Name	Value
Card ID	0424H
Card Description	PTQ-AFC
Medium Description	Flow Computer Module
Long Description	Gas/Liquid Flow Computer Communication ...
Power	800
Number of Parameters Used	0
Default Number of Parameters	0
In Bytes	0
Out Bytes	0
Module Type	0-Discrete
Doc Only	1-True
MCS Simple 1	0-Ordinary
MCS Simple 2	0000-0000
Default Parameter Data	
Rack View Bitmap	PTQAFC.bmp
Drop View Bitmap	PTQAFC.bmp

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:



Properties - Slot (03)

Module	PTQ-101S
Description	IEC-60870-5-101 Slave Module
Power Rating	800 mA

Properties - Drop (01)

Drop Series	Quantum Series
Hold Up Time	3
Rack 1	
Rack 2	
Status Register (3x)	
Input Points	0000 / 1024

SAFTESTTEST ○ Edit ○ Summary ● Traffic Cop

Head: 0 Drop: 1 Rack: 1 Slot: 3

Short Comments

Welcome to the Tracking Help! Here you will find relevant information that directly pertains to the current instruction or I/O module being accessed.

Logged in user: N/A | PROGRAM | SAFTESTTEST - <Quantum 534 @ TCPIP:192.168.0.139> Offline

5 Configuring the Processor with UnityPro XL

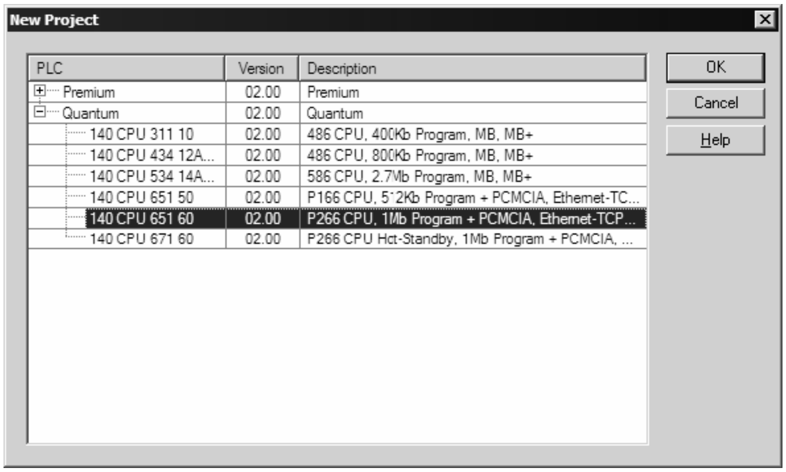
In This Chapter

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.

5.1 Create a New Project

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

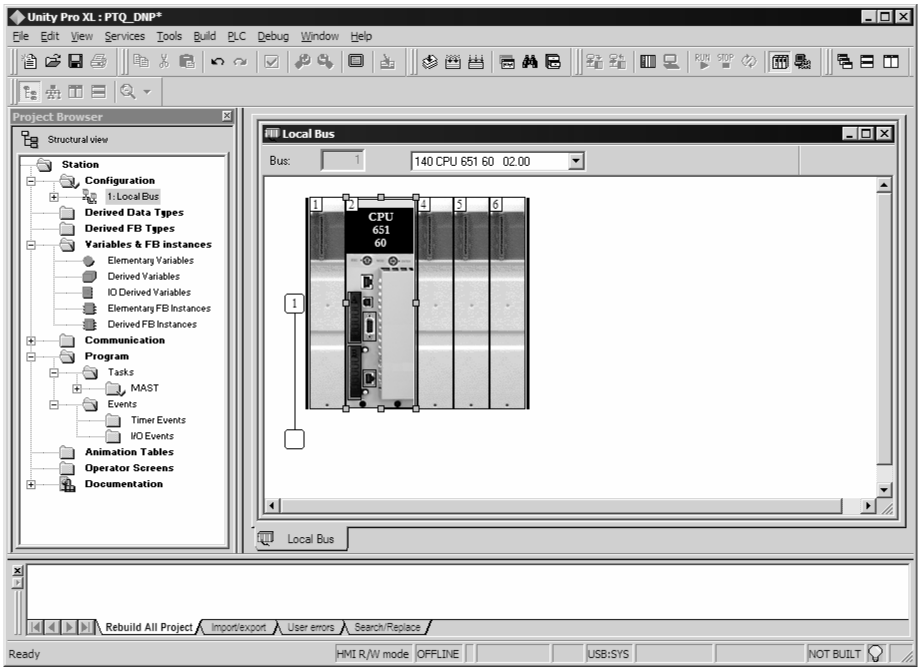
1 In the New Project dialog box, choose the CPU type. In the following illustration, the CPU is 140 CPU 651 60. Choose the processor type that matches your own hardware configuration, if it differs from the example. Click OK to continue.



New Project

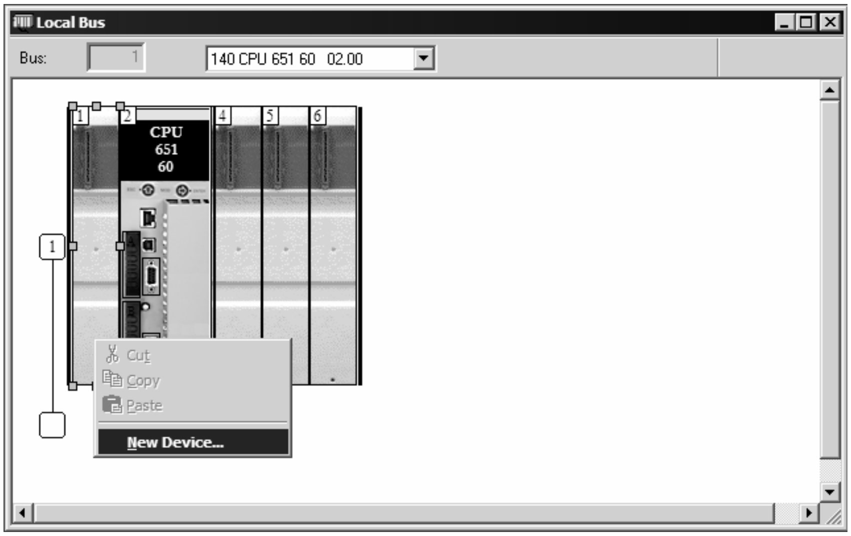
PLC	Version	Description
⊞ Premium	02.00	Premium
⊟ Quantum	02.00	Quantum
140 CPU 311 10	02.00	486 CPU, 400Kb Program, MB, MB+
140 CPU 434 12A...	02.00	486 CPU, 800Kb Program, MB, MB+
140 CPU 534 14A...	02.00	586 CPU, 2.7Mb Program, MB, MB+
140 CPU 651 50	02.00	P166 CPU, 512Kb Program + PCMCIA, Ethernet-TC...
140 CPU 651 60	02.00	P266 CPU, 1Mb Program + PCMCIA, Ethernet-TCP...
140 CPU 671 60	02.00	P266 CPU Hot-Standby, 1Mb Program + PCMCIA, ...

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



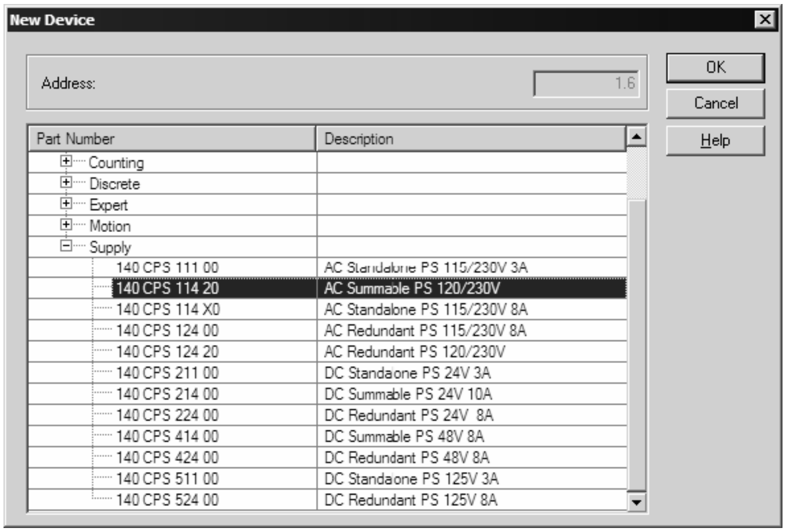
Bus: 1 140 CPU 651 60 02.00

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..



Bus: 1 140 CPU 651 60 02.00

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



Address: 1.6

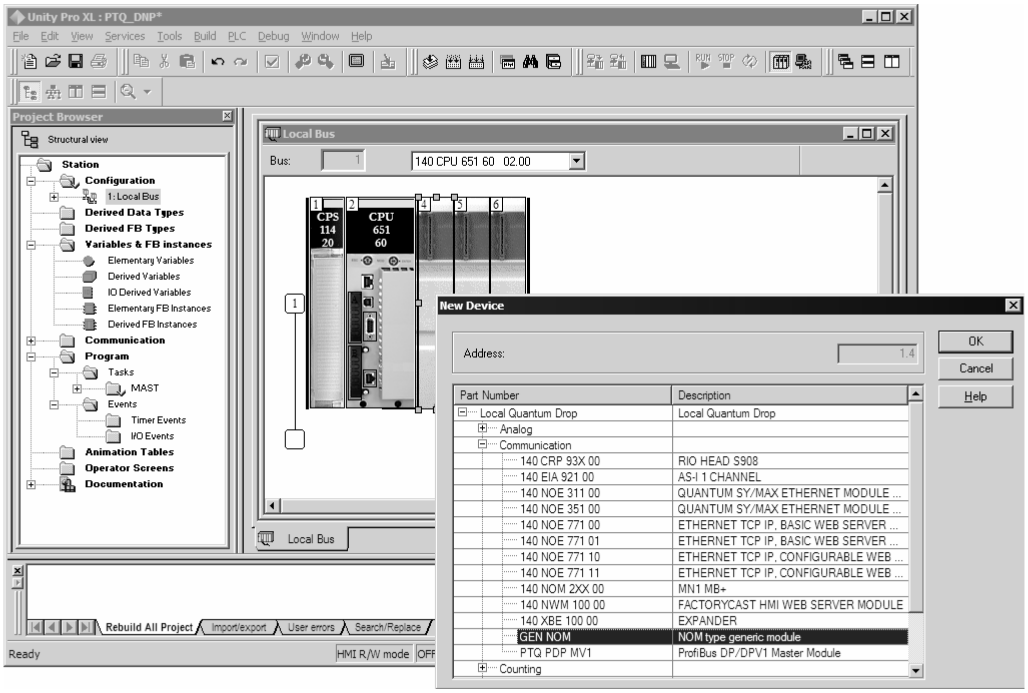
Part Number	Description
⊞ Counting	
⊞ Discrete	
⊞ Expert	
⊞ Motion	
⊟ Supply	
140 CPS 111 00	AC Standalone PS 115/230V 3A
140 CPS 114 20	AC Summable PS 120/230V
140 CPS 114 X0	AC Standalone PS 115/230V 8A
140 CPS 124 00	AC Redundant PS 115/230V 8A
140 CPS 124 20	AC Redundant PS 120/230V
140 CPS 211 00	DC Standaone PS 24V 3A
140 CPS 214 00	DC Summable PS 24V 10A
140 CPS 224 00	DC Redundant PS 24V 8A
140 CPS 414 00	DC Summable PS 48V 8A
140 CPS 424 00	DC Redundant PS 48V 8A
140 CPS 511 00	DC Standaone PS 125V 3A
140 CPS 524 00	DC Redundant PS 125V 8A

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

5.2 Add the PTQ Module to the Project

The next step is to add the PTQ module.

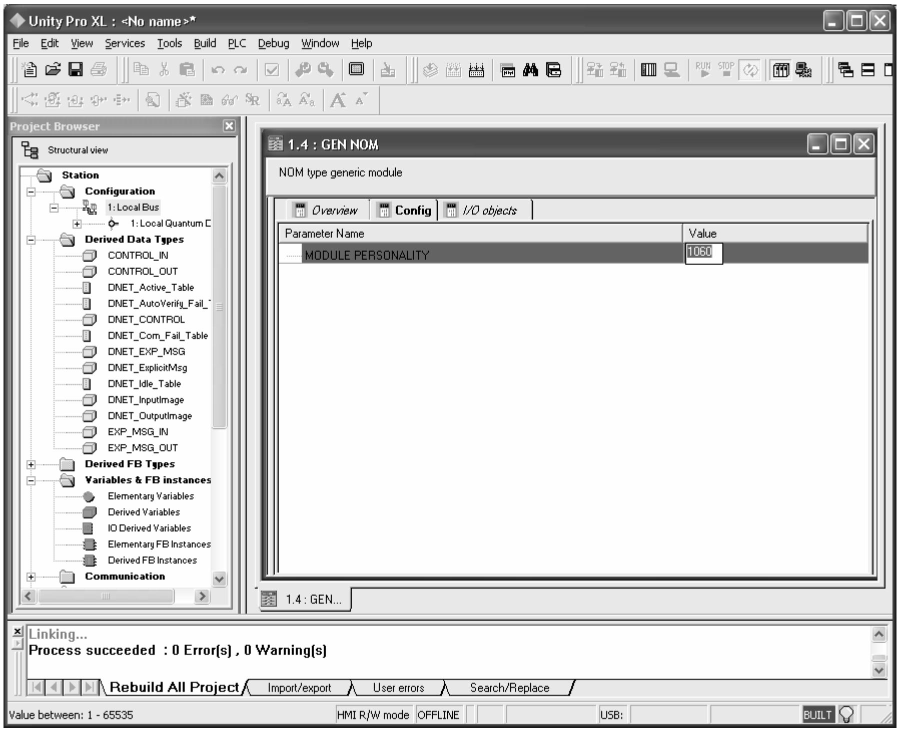
1 Expand the Communication tree, and select GEN NOM. This module type provides extended communication capabilities for the Quantum system, and allows communication between the PLC and the PTQ module without requiring additional programming.



Address: 1.4

Part Number	Description
⊟ Local Quantum Drop	Local Quantum Drop
⊞ Analog	
⊟ Communication	
140 CRP 93X 00	RIO HEAD S908
140 EIA 921 00	AS-I 1 CHANNEL
140 NOE 311 00	QUANTUM SY/MAX ETHERNET MODULE ...
140 NOE 351 00	QUANTUM SY/MAX ETHERNET MODULE ...
140 NOE 771 00	ETHERNET TCP IP, BASIC WEB SERVER ...
140 NOE 771 01	ETHERNET TCP IP, BASIC WEB SERVER ...
140 NOE 771 10	ETHERNET TCP IP, CONFIGURABLE WEB ...
140 NOE 771 11	ETHERNET TCP IP, CONFIGURABLE WEB ...
140 NOM 2XX 00	MN1 MB+
140 NWM 100 00	FACTORYCAST HMI WEB SERVER MODULE
140 XBE 100 00	EXPANDER
GEN NOM	NOM type generic module
PTQ PDP MV1	ProfiBus DP/DPV1 Master Module
⊞ Counting	

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



Parameter Name	Value
MODULE PERSONALITY	1060

Linking...
Process succeeded : 0 Error(s) , 0 Warning(s)

Value between: 1 - 65535

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

4 Save the project.

5.3 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.

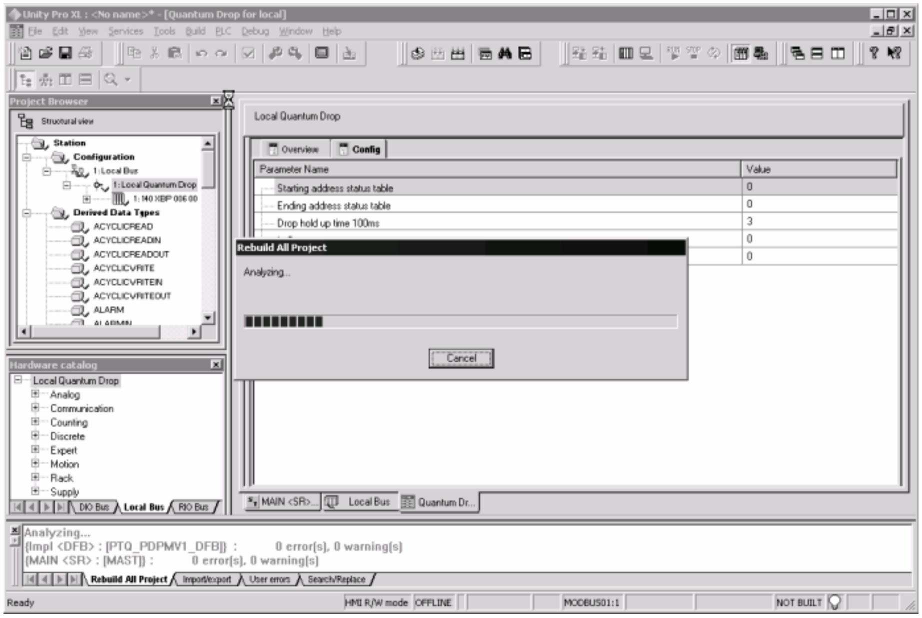
> Note: The following steps show you how to build the project in Unity Pro XL. This is not intended to provide detailed information on using Unity Pro XL, or debugging your programs. Refer to the documentation for your processor and for Unity Pro XL for specialized information.

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.



Parameter Name	Value
Starting address status table	0
Ending address status table	0
Drop hold up time 100ms	3
	0
	0

Rebuild All Project

Analyzing...

Cancel

```
Analyzing...
{Impl <DFB> : [PTQ_PDPMV1_DFB]} :        0 error(s), 0 warning(s)
{MAIN <SR> : [MAST]} :        0 error(s), 0 warning(s)
```

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

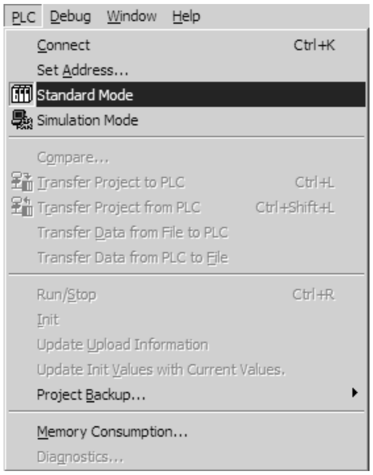
5.4 Connect Your PC to the Processor

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.

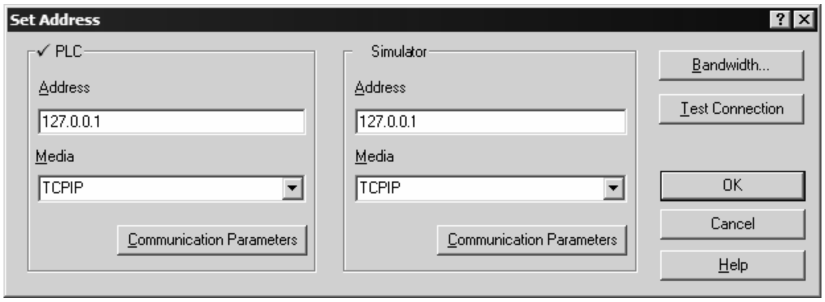
> **Note:** If you have never connected from the PC to your processor before, you must verify that the necessary port drivers are installed and available to UnityPro XL.

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.


PLC
Debug
Window
Help
Connect Ctrl+K
Set Address...
Standard Mode
Simulation Mode
Compare...
Transfer Project to PLC Ctrl+L
Transfer Project from PLC Ctrl+Shift+L
Transfer Data from File to PLC
Transfer Data from PLC to File
Run/Stop Ctrl+R
Init
Update Upload Information
Update Init Values with Current Values.
Project Backup...
Memory Consumption...
Diagnostics...

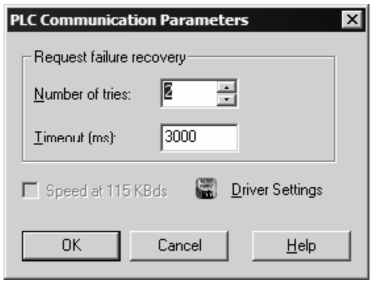
2 Open the PLC menu, and choose Set address... This action opens the Set address dialog box. Open the Media dropdown list and choose the connection type to use (TCPIP or USB).


Set Address

✓ PLC
Address: 127.0.0.1
Media: TCPIP

Simulator
Address: 127.0.0.1
Media: TCPIP

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.



PLC Communication Parameters

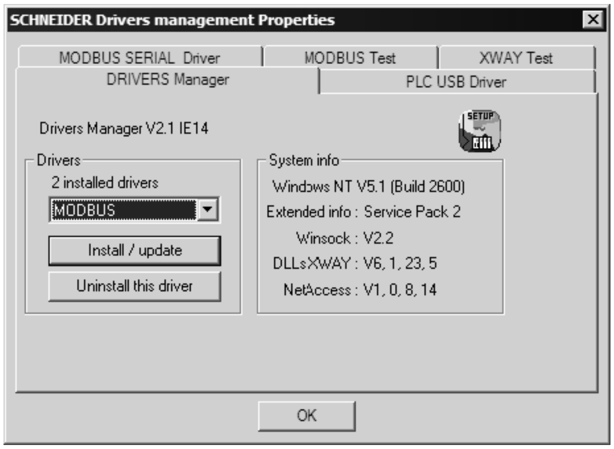
Request failure recovery

Number of tries: 2

Timeout (ms): 3000

□ Speed at 115 KBds

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



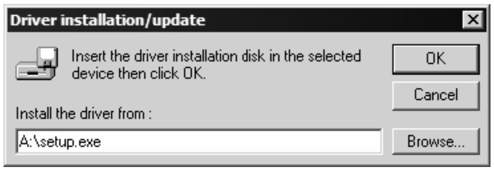
SCHNEIDER Drivers management Properties

Drivers Manager V2.1 IE14

Drivers
2 installed drivers
MODBUS

System info
Windows NT V5.1 (Build 2600)
Extended info : Service Pack 2
Winsock : V2.2
DLLsXWAY : V6, 1, 23, 5
NetAccess : V1, 0, 8, 14

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.



Driver installation/update

Insert the driver installation disk in the selected device then click OK.

Install the driver from : A:\setup.exe

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.

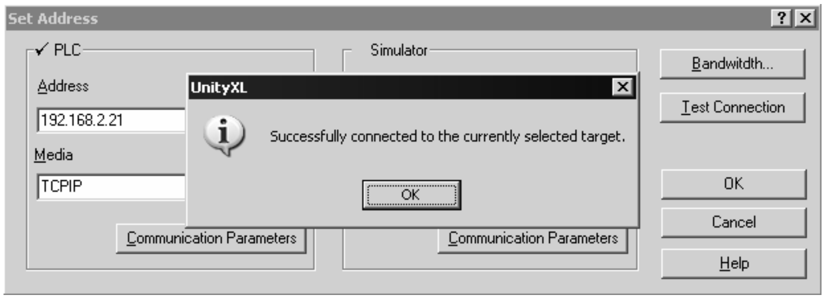
5.4.1 Connecting to the Processor with TCPIP

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.



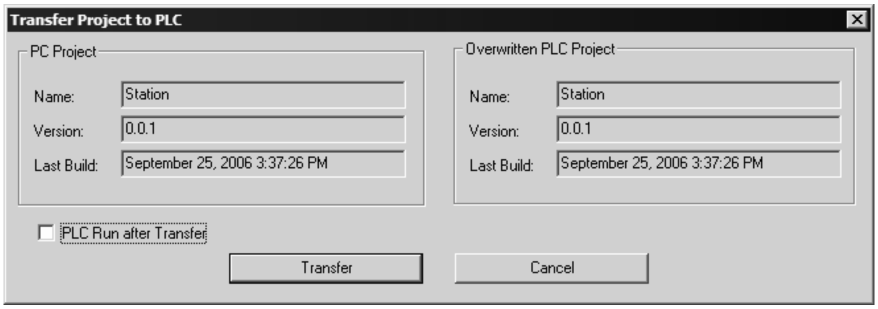
Address: 192.168.2.21
Media: TCPIP

Successfully connected to the currently selected target.

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

5.5 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.



PC Project
Name: Station
Version: 0.0.1
Last Build: September 25, 2006 3:37:26 PM

Overwritten PLC Project
Name: Station
Version: 0.0.1
Last Build: September 25, 2006 3:37:26 PM

☐ PLC Run after Transfer

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.

6 Setting Up the ProTalk Module

In This Chapter

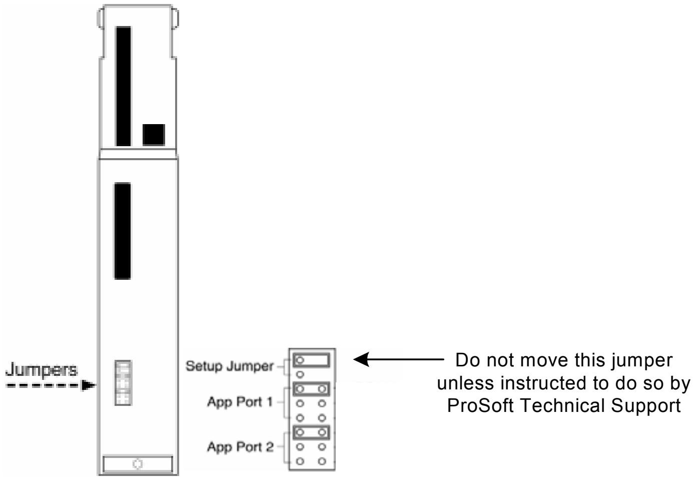
After you complete the following procedures, the ProTalk module will actively be transferring data bi-directionally with the processor.

6.1 Install the ProTalk Module in the Quantum Rack

6.1.1 Verify Jumper Settings

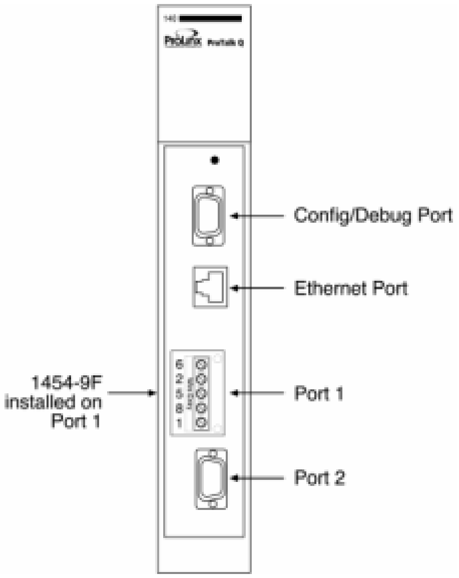
ProTalk modules are configured for RS-232 serial communications by default. To use RS-422 or RS-485, you must change the jumpers.

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


Jumpers
Setup Jumper
App Port 1
App Port 2
Do not move this jumper unless instructed to do so by ProSoft Technical Support

6.1.2 Inserting the 1454-9F connector

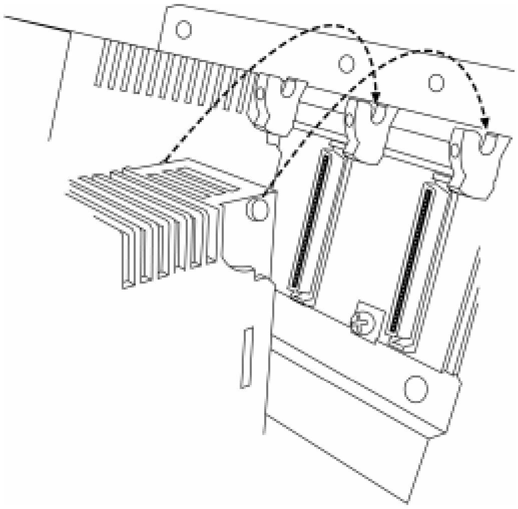
Insert the 1454-9F connector as shown. Wiring locations are shown in the table:



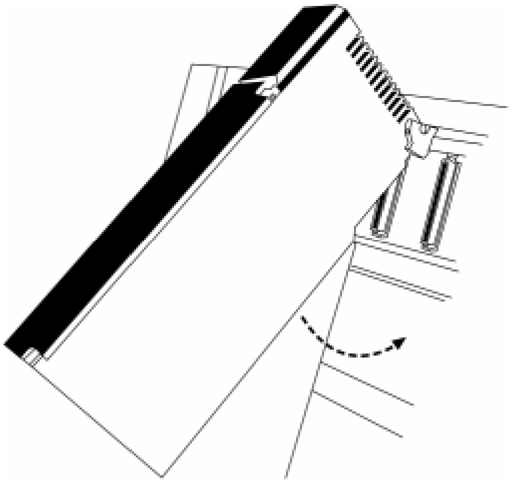
6.1.3 Install the ProTalk Module in the Quantum Rack

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.

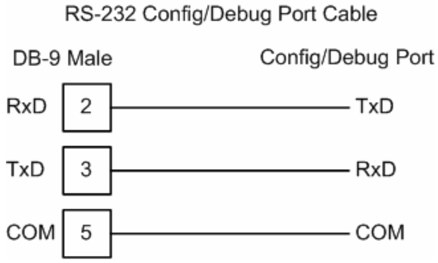
6.2 Cable Connections

The application ports on the PTQ-DNPS 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.

6.2.1 RS-232 Configuration/Debug Port

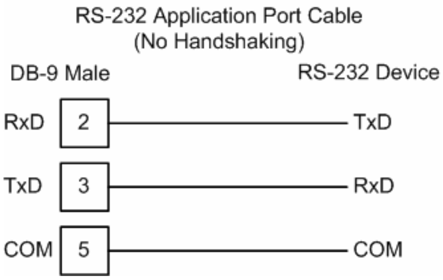
This port is physically a DB-9 connection. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following diagram:



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

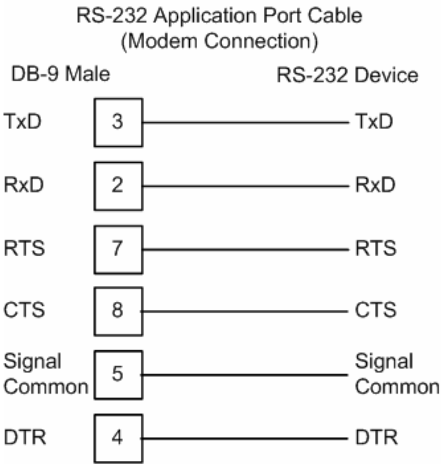
6.2.2 RS-232

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

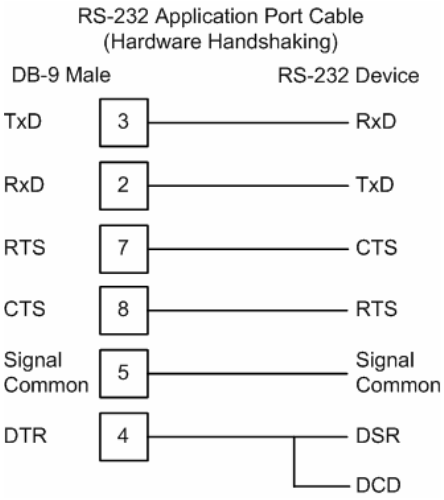
This type of connection is required between the module and a modem or other communication device.



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

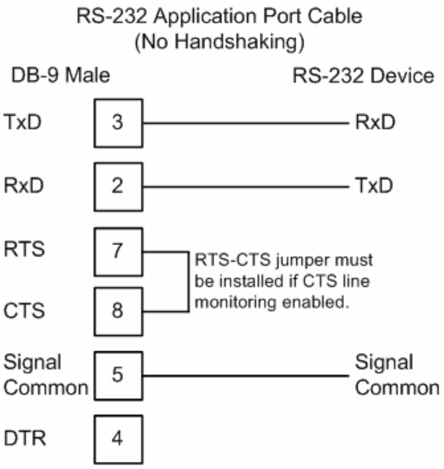
RS-232: Null Modem Connection (Hardware Handshaking)

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)

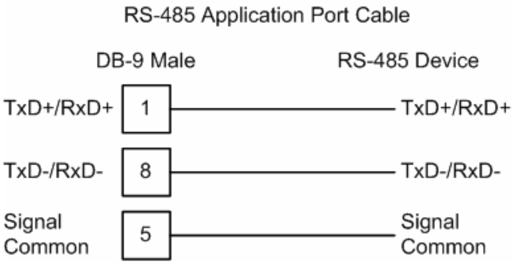
This type of connection can be used to connect the module to a computer or field device communication port.


RS-232 Application Port Cable
(No Handshaking)
DB-9 Male
RS-232 Device
TxD
3
RxD
RxD
2
TxD
RTS
7
CTS
8
RTS-CTS jumper must be installed if CTS line monitoring enabled.
Signal Common
5
Signal Common
DTR
4

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.

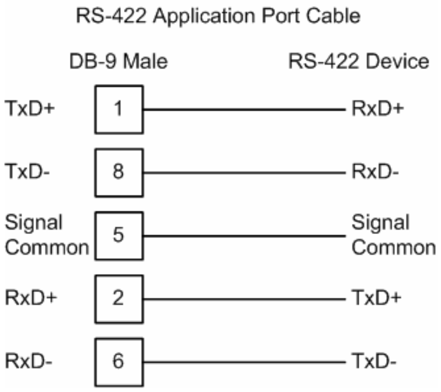
6.2.3 RS-485

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.

6.2.4 RS-422

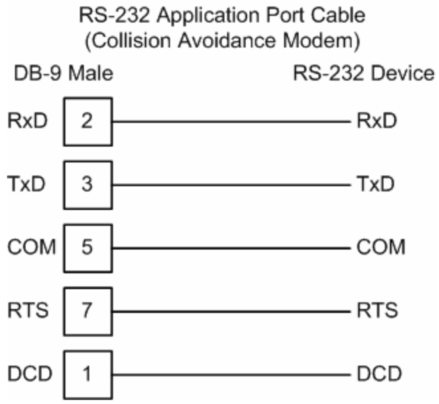


RS-485 and RS-422 Tip

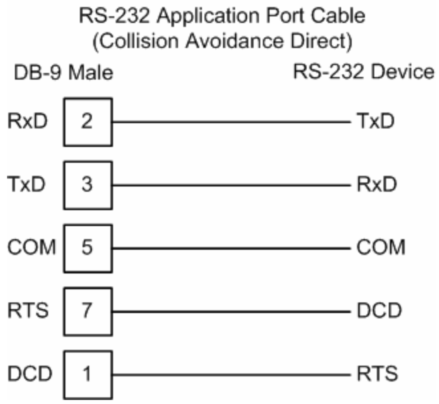
If communication in the RS-422/RS-485 mode does not work at first, despite all attempts, try switching termination polarities. Some manufacturers interpret +/- and A/B polarities differently.

6.3 Collision Avoidance (DNP modules only)

The RTS line is controlled by the RTS on and off parameters set for the port. If the CTS line is used (usually only required for half-duplex modems and not defined for use in the DNPS specification), the RTS and CTS lines must either be connected together or connected to the modem. The following illustration shows the cable required when connecting the port to a modem.



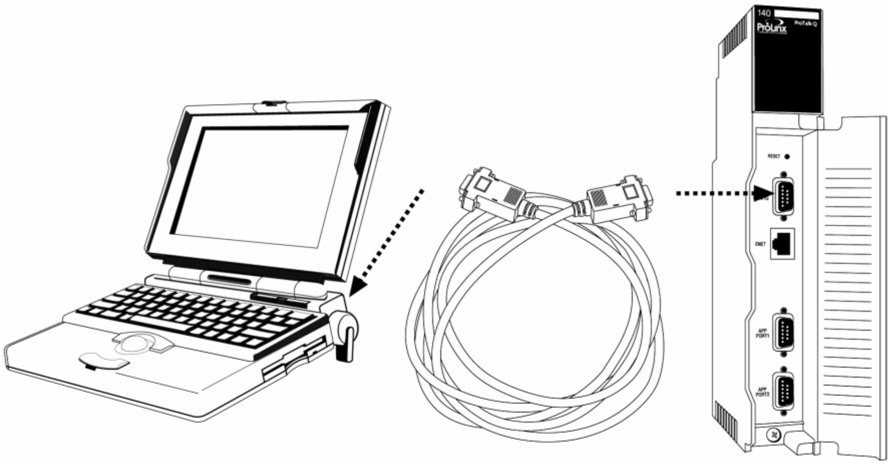
If collision avoidance is used in a point-to-point connection on the RS-232 interface, the following cable should be used.



6.4 Connect the PC to the ProTalk Configuration/Debug Port

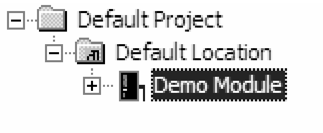
Make sure you have exited the Quantum programming software before performing these steps. This action will avoid serial port conflict.

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

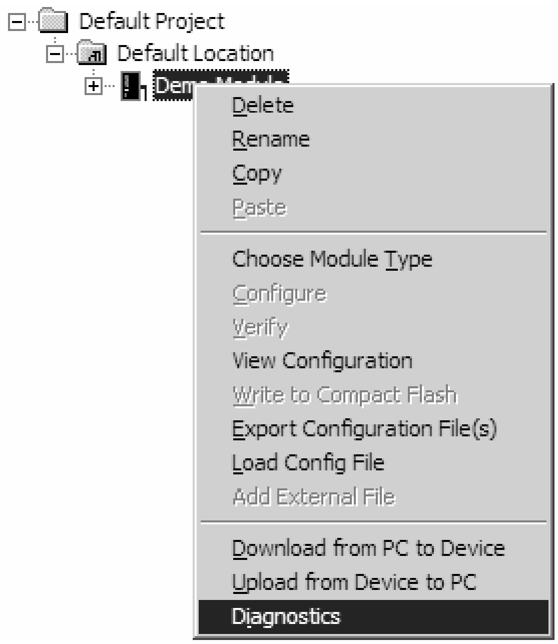


To connect to the module's Configuration/Debug serial port:

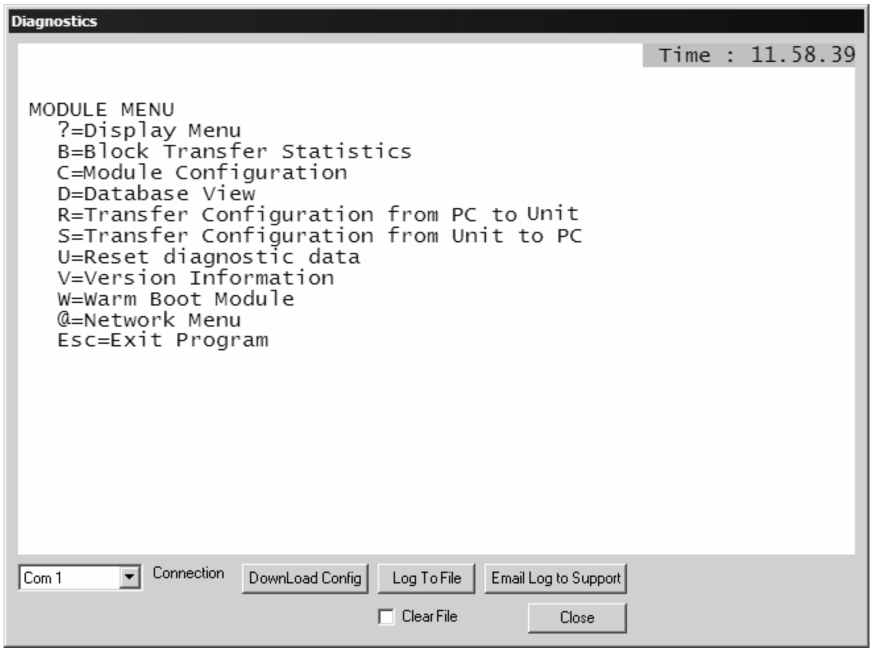
1 Start PCB program with the application file to be tested. Right click over the module icon.



2 On the shortcut menu, choose Diagnostics.



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



```
                                                  Time : 11.58.39

MODULE MENU
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  R=Transfer Configuration from PC to Unit
  S=Transfer Configuration from Unit to PC
  U=Reset diagnostic data
  V=Version Information
  W=Warm Boot Module
  @=Network Menu
  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.

7 Modifying the Configuration File

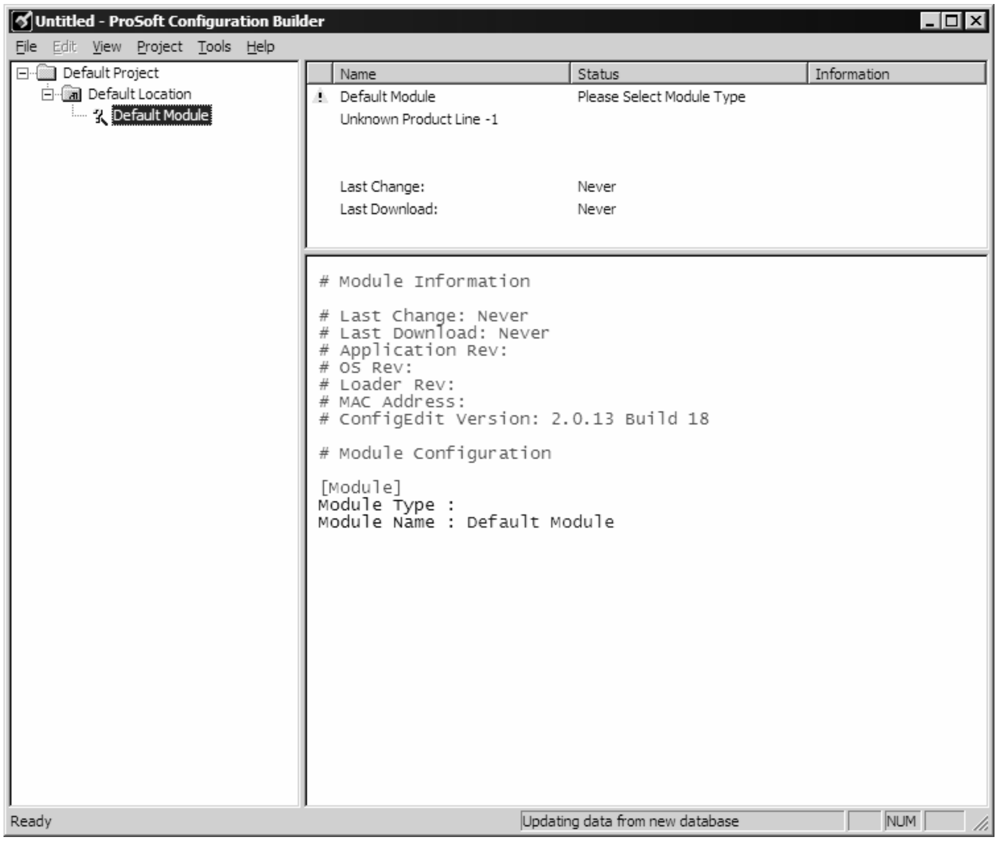
In This Chapter

7.1 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.

7.1.1 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.



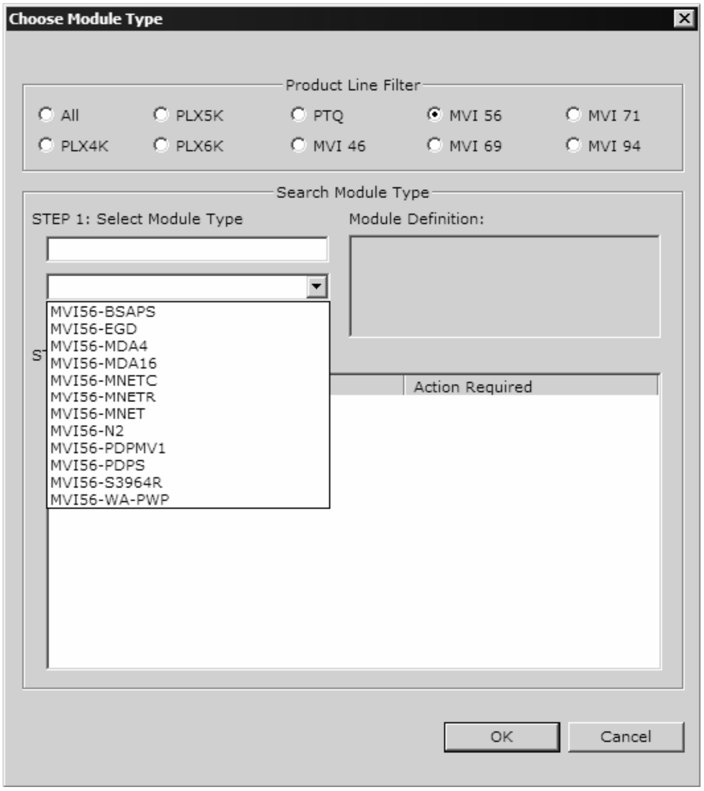
Name	Status	Information
Default Module	Please Select Module Type	
Unknown Product Line -1		
Last Change:	Never	
Last Download:	Never	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 2.0.13 Build 18

# Module Configuration

[Module]
Module Type :
Module Name : Default Module
```

Your first task is to add the PTQ-DNPS 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-DNPS, and then click OK to save your settings and return to the ProSoft Configuration Builder window.

The next task is to set the module parameters.

Adding a Project

To add a project to an existing project file:

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

Adding a Module

To add a module to your project:

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

or

1 Open the Project menu and choose Location

2 On the Location menu, choose Add Module.

To add a module to a different location:

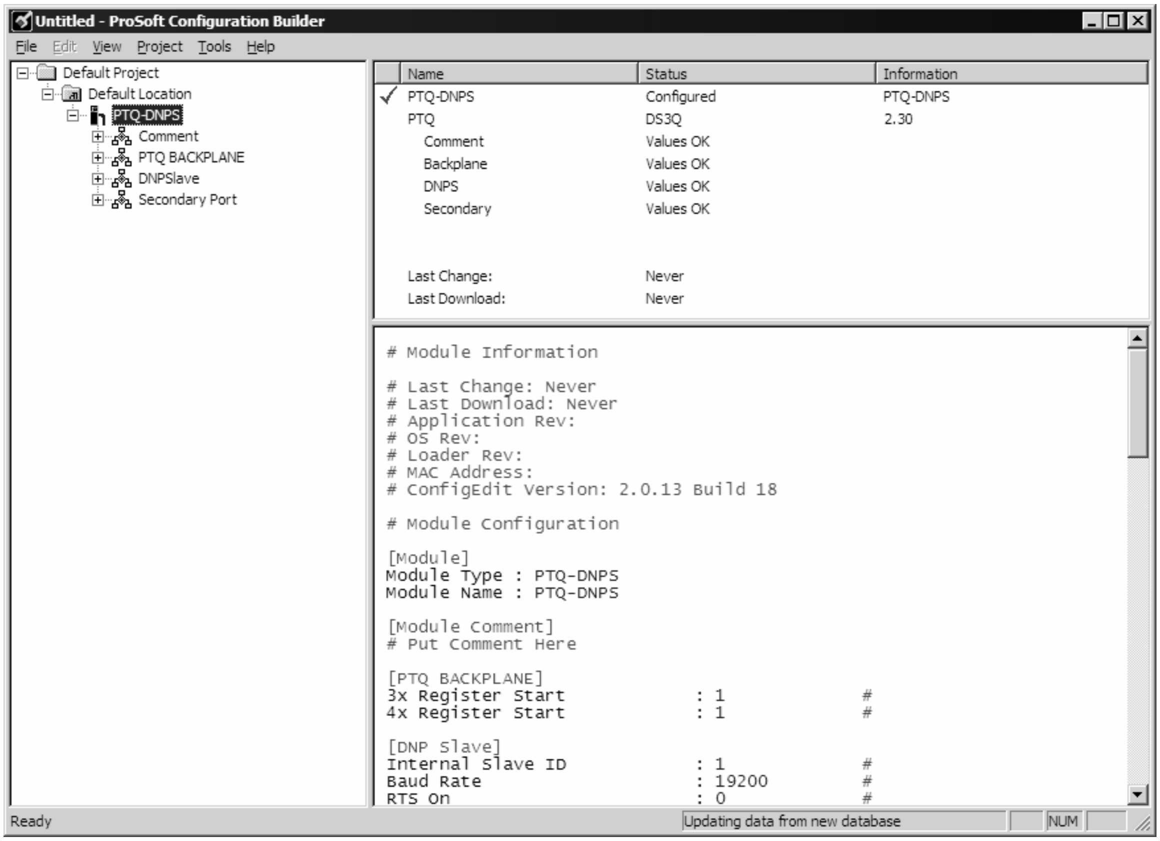
1 Right-click the Location folder and choose Add Module. A new module icon appears.

or
Select the Location icon.

2 From the Project menu, select Location, then select Add Module.

7.1.2 Set Module Parameters

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



Name	Status	Information
✓ PTQ-DNPS	Configured	PTQ-DNPS
PTQ	DS3Q	2.30
Comment	Values OK	
Backplane	Values OK	
DNPS	Values OK	
Secondary	Values OK	
Last Change:	Never	
Last Download:	Never	

```
# Module Information

# Last Change: Never
# Last Download: Never
# Application Rev:
# OS Rev:
# Loader Rev:
# MAC Address:
# ConfigEdit Version: 2.0.13 Build 18

# Module Configuration

[Module]
Module Type : PTQ-DNPS
Module Name : PTQ-DNPS

[Module Comment]
# Put Comment Here

[PTQ BACKPLANE]
3x Register Start            : 1           #
4x Register Start            : 1           #

[DNP Slave]
Internal Slave ID            : 1           #
Baud Rate                    : 19200       #
RTS On                       : 0           #
```

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

To rename an object:

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

Module Entries

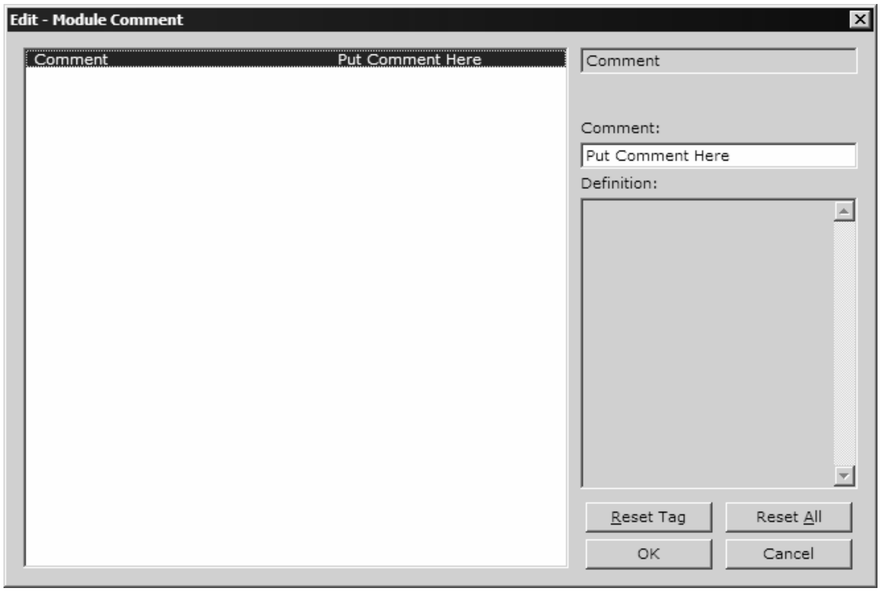
To configure module parameters

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

Comment Entries

To add comments to your configuration file:

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


Edit - Module Comment

Comment	Put Comment Here

Comment

Comment: Put Comment Here

Definition:

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

Printing a Configuration File

To print a configuration file:

1. Select the Module icon, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose View Configuration. This action opens the View Configuration window.
3. On the View Configuration window, open the File menu, and choose Print. This action opens the Print dialog box.

4 On the Print dialog box, choose the printer to use from the dropdown list, select printing options, and then click OK.

7.2 [Module]

This section defines the module name for the Module.

7.2.1 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.

7.3 [PTQ BACKPLANE]

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

The following example shows a sample [PTQ Backplane] section:

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

7.3.1 3x Register Start

1 to n

This parameter sets the first register in the processor where the data transferred from the module to the processor will be placed.

7.3.2 4x Register Start

1 to n

This parameter sets the first register in the processor where the data transferred from the processor to the module is present.

7.4 [DNP Slave]

This section provides information required to configure a slave application with the module. Most entries contained within this section are self explanatory.

Note: A limitation of the DNP slave driver is that all points defined in the module slave database must fit within one Class 0 poll. The maximum packet size for a Class 0 poll is 2048 bytes. A DNP Message Size Calculator is available on the ProSoft Technology web site. This calculator will help you ensure that the packet size fits within this requirement.

The following example shows a sample [DNP Slave] section:

```
# This section defines the configuration for the Module.
# port. This port will receive requests from a remote DNP master unit.
#
[DNP Slave]
Internal Slave ID      : 32     #0-65534 slave identification code for this
# unit

# DNP slave communication port configuration
Baud Rate              : 19200  #Baud rate for port 110-115200
RTS On                 : 0      #0-65535 milliseconds before message
RTS Off                : 0      #0-65535 milliseconds after message
Min Response Delay     : 0      #0-65535 milliseconds before response sent from
                                # slave

# DNP slave modem configuration
Modem                  : No     #Use a dial-up modem on this port (Yes or
                                # No)
Connect Timeout        : 20000  #0-65535 milliseconds before connect timeout
First Character Delay  : 1000   #0-65535 milliseconds before 1st char after
connect
Redial Delay Time      : 100    #0-65535 1/10 seconds min before redial
                                # attempt
Redial Random Delay    : 150    #0-65535 1/10 seconds random before redial
                                # attempt
Idle Timeout           : 200    #0-65535 1/10 seconds inactive timeout
Phone Number           : ATDT18001234567

# Collision Avoidance parameters
Collision Avoidance    : No     #Use Collision Avoidance (Yes or No)
CD Idle Time           : 10     #0-32000 mSec min idle time before transmit
CD Random Time         : 15     #0-32000 mSec random idle time before
                                # transmit
CD Time Before Receive : 5      #0-65535 milliseconds before receive

#Default Class Settings
BI Class               :  2     #Default class for binary input events
                                # (0=disable, else 1-3)
AI Class               :  3     #Default class for analog input events
                                # (0=disable, else 1-3)
Float Class            :  0     #Default class for float input events
                                # (0=disable, else 1-3)

# DNP specific parameters
AI Deadband            : 1      #0-32767 analog deadband value for events
Float Deadband         : 1000.0 #Single float deadband
Select/Operate Arm Time: 2000   #1-65535 milliseconds arm timeout for select/op
                                # outputs
Write Time Interval    : 60     #0-1440 minutes for time sync from master

Data Link Confirm Mode : Never  #DL confirm mode (N=Never, S=Sometimes,
                                # A=Always)
Data Link Confirm Tout : 1000   #1-65535 milliseconds DL confirm timeout
Data Link Max Retry    : 2      #0-255 maximum DL confirm retry count
App Layer Confirm Tout : 2000   #1-65535 milliseconds App Layer confirm timeout

Unsolicited Response   : No     #Generate Unsolicited responses (Yes or No)
Class 1 Unsol Resp Min : 10     #1-255 min number of events before send
Class 2 Unsol Resp Min : 10     #1-255 min number of events before send
Class 3 Unsol Resp Min : 10     #1-255 min number of events before send
Unsol Resp Delay       : 2000   #0-65535 milliseconds before events sent
UResp Master Address   : 1      #DNP address of master to send UResp data
UResp Retry Count      : 0      #0-255 Number of retries before switching
                                # ports

BI with flag            : Yes    #return BI data with flag data
BI Events without time  : Yes    #return BI events without time/date
BO without flag         : Yes    #return BO data without flag data (packed)
Counter with flag       : Yes    #return counters with flag byte
Frozen counter with flag: Yes    #return frozen counters with flag byte
AI with flag            : Yes    #return AI with flag byte
AI Events with time     : Yes    #timestamp AI Event data default (Yes or
                                 # No)

Time Sync Before Events: No     #timesync module before events gen (Yes or
                                # No)
Initialize DNP Database: Yes    #Initialize the DNP Slave output database
                                # areas (Y/N)
```

7.4.1 Internal Slave ID

0 to 65534

This is the DNP address for the module. All messages with this address from the master will be processed by the module.

7.4.2 Baud Rate

Baud rate value

DNP Port Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 384 (38400) , 576 (57600), 115 (115200)

7.4.3 RTS On

0 to 65535 milliseconds

This value represents the number of 1 ms increments to be inserted between asserting the RTS modem line and the actual transmission of the data.

7.4.4 RTS Off

0 to 65535 milliseconds

This value represents the number of 1 ms increments to be inserted after the last character of data is transmitted before the RTS modem line is dropped.

7.4.5 Min Response Delay

0 to 65535 milliseconds

Minimum time between receiving a request and transmitting a response. Allows master time to disable transmitter on an RS-485 network.

7.4.6 Modem

Y or N

This parameter defines if a dial-up modem is used on the secondary DNP slave port. A modem cannot be used if the port is configured as a master.

7.4.7 Connect Timeout

0 to 65535

Defines the number of milliseconds to wait for the CD signal to be set high. The CD signal indicates a connection is made using a dial-up modem.

7.4.8 First Character Delay

0 to 65535

Defines the number of milliseconds to wait before sending the first message after the connection is first made. This delay only applies to the first packet sent to the modem.

7.4.9 Redial Delay Time

0 to 32000

Defines the minimum number of milliseconds to wait before a redial attempt is made by the slave.

7.4.10 Redial Random Delay

0 to 32000

Defines a random millisecond time range to be added to the redial delay time before the modem is accessed.

7.4.11 Idle Timeout

0 to 65535

Defines the number of milliseconds the modem is inactive before it will disconnect.

7.4.12 Phone Number

ASCII String Data

This field contain a null-terminated, ASCII character string used by the dial-up modem. The string must contain all characters required by the modem. An example string is ATDT1800222333. Maximum length is 34 bytes including the terminating 0.

7.4.13 Collision Avoidance

Y or N

This parameter defines if collision avoidance will be utilized on the primary DNP slave port.

7.4.14 CD Idle Time

0 to 32000

Defines the minimum number of milliseconds to wait before transmitting a message after the CD signal is recognized as low.

7.4.15 CD Random Time

0 to 32000

Defines the range of random time to be added to the CD Idle Time before a message will be transmitted from the slave.

7.4.16 CD Time Before Receive

0 to 65535

Defines the number of milliseconds to wait before receiving characters after the CD signal is recognized as high.

7.4.17 BI Class

0 to 3

This parameter specifies the default class to be utilized for all the binary input points in the DNP database that are not defined in the override list section.

7.4.18 AI Class

0 to 3

This parameter specifies the default class to be utilized for all the analog input points in the DNP database that are not defined in the override list section.

7.4.19 Float Class

0 to 3

This parameter specifies the default class to be utilized for all the floating-point input points in the DNP database that are not defined in the override list section.

7.4.20 AI Deadband

0 to 32767

This parameter specifies the default deadband value assigned to all points not defined in the override list for the analog input point type in the DNP database.

7.4.21 Float Deadband

0 to maximum float value

This parameter specifies the default deadband value assigned to all points not defined in the override list for the floating-point input point type in the DNP database.

7.4.22 Select/Operate Arm Time

1 to 65535 milliseconds

Time period after select command received in which operate command will be performed. Once the select command is received, the operate command will only be honored if it arrives within this period of time.

7.4.23 Write Time Interval

0 to 1440 minutes

Time interval to set the need time IIN bit (0=never), which will cause the master to write the time. Stored in milliseconds in the module memory.

7.4.24 Data Link Confirm Mode

Coded Value (N=Never, S=Sometimes, A=Always)

IED can request acknowledgement from master station when sending data. The codes are as follows: 0=Never, 1=Sometimes, 2=Always

7.4.25 Data Link Confirm Tout

1 to 65535 milliseconds

Time period to wait for Master Data Link confirmation of last frame sent. This time is in milliseconds. This parameter is only used if the frame is sent with confirmation requested.

7.4.26 Data Link Max Retry

0 to 255 retries

Maximum number of retries at the Data Link level to obtain a confirmation. If this value is set to 0, retries are disabled at the data link level of the protocol. This parameter is only used if the frame is sent with confirmation requested.

7.4.27 App Layer Confirm Tout

1 to 65535 milliseconds

Event data contained in the last response may be sent again if not confirmed within the millisecond time period set. If application layer confirms are used with data link confirms, ensure that the application layer confirm timeout is set long enough.

7.4.28 Unsolicited Response

Y or N

Set if the slave unit will send unsolicited response messages. If set to N, the slave will not send unsolicited responses. If set to Y, the slave will send unsolicited responses.

7.4.29 Class 1 Unsol Resp Min

1 to 255 events

Minimum number of events in Class 1 required before an unsolicited response will be generated.

7.4.30 Class 2 Unsol Resp Min

1 to 255 events

Minimum number of events in Class 2 required before an unsolicited response will be generated.

7.4.31 Class 3 Unsol Resp Min

1 to 255 events

Minimum number of events in Class 3 required before an unsolicited response will be generated.

7.4.32 Unsol Resp Delay

0 to 65535 milliseconds

Maximum number of 1 millisecond intervals to wait after an event occurs before sending an unsolicited response message. If set to 0, only use minimum number of events.

7.4.33 Uresp Master Address

0 to 65534

DNP destination address where unsolicited response messages are sent.

7.4.34 Uresp Retry Count

0 to 255 retries

Determines the number of unsolicited message retries sent on primary DNP port before changing to secondary port. If the value is 0, port switching will be disabled.

7.4.35 BI with Flag

Y or N

This parameter determines which variation will be returned for object 1 when the master requests variation 0. If the parameter is set to N, variation 1 will be returned. If the parameter is set to Y, variation 2 will be returned.

7.4.36 BI Events without time

Y or N

This parameter determines which variation will be returned for object 2 when the master requests variation 0. If the parameter is set to N, variation 2 will be returned. If the parameter is set to Y, variation 1 will be returned.

7.4.37 BO without flag

Y or N

This parameter determines which variation will be returned for object 10 when the master requests variation 0. If the parameter is set to N, variation 2 will be returned. If the parameter is set to Y, variation 1 will be returned.

7.4.38 Counter with flag

Y or N

This parameter determines which variation will be returned for object 20 when the master requests variation 0. If the parameter is set to N, variation 5 will be returned. If the parameter is set to Y, variation 1 will be returned.

7.4.39 Frozen counter with flag

Y or N

This parameter determines which variation will be returned for object 21 when the master requests variation 0. If the parameter is set to N, variation 9 will be returned. If the parameter is set to Y, variation 1 will be returned.

7.4.40 AI with flag

Y or N

This parameter determines which variation will be returned for object 30 when the master requests variation 0. If the parameter is set to N, variation 4 will be returned. If the parameter is set to Y, variation 2 will be returned.

7.4.41 AI Events with time

Y or N

This parameter determines if the analog input events generated by the module will include the date and time of the event. If the parameter is set to N, the default is set to no time data. If the parameter is set to Y, the default object will include the time of the event.

7.4.42 Time Sync Before Events

Y or N

This parameter determines if events are to be generated by the module before the time synchronization from the master unit. If the parameter is set to N, no events will be generated until the module's time has been synchronized. If the parameter is set to Y, events will always be generated.

7.4.43 Initialize DNP Database

Y or N

This parameter determines if the module will request data from the processor to initialize the DNP database output data areas. If this option is utilized, ladder logic is required to send the requested block from the processor to the module.

7.5 [DNP Slave Database]

The [DNP SLAVE DATABASE] section contains the information to size the database to be used to interface with the remote master. The values entered in this section define the blocks to be transferred between the module and the processor over the backplane and the data to be interfaced with the remote DNP master device. The parameters defined in this section are shown in the configuration form.

```
[DNP Slave Database]
Binary Inputs          : 512    #0-512 point count to hold BI data
Analog Inputs          :  50    #0-512 points of analog input data
Float Inputs           :   5    #0-128 points of floating-point format data
Counters               :  20    #0-128 points of counter data
Binary Outputs         : 160    #0-512 point count to hold BO data
Analog Outputs         :  28    #0-512 points of analog output data
Float Outputs          :   4    #0-128 points of floating-point format data
```

7.5.1 Binary Inputs

0 to 512 points

Number of digital input points to configure in the DNP slave device. Each point will be stored as a single word in the module memory.

7.5.2 Analog Inputs

0 to 512 points

Number of analog input points to configure in the DNP slave device. Each point will occupy a two word area in the module memory.

7.5.3 Float Inputs

0 to 128 points

Number of floating-point input points to configure in the DNP slave device. Each point will occupy a three word area in the module memory.

7.5.4 Counters

0 to 128 points

Number of counter points to configure in the DNP slave device. Each point will occupy a three word area in the module memory. This number corresponds to the number of frozen counters. The application maps the counters to the frozen counters directly.

7.5.5 Binary Outputs

0 to 512 points

Number of digital output points to configure in the DNP slave device. Each point will be stored as a single word in the module memory.

7.5.6 Analog Outputs

0 to 512 points

Number of analog output points to configure in the DNP slave device. Each point will occupy a two word area in the module memory.

7.5.7 Float Outputs

0 to 128 points

Number of floating-point output points to configure in the DNP slave device. Each point will occupy a three word area in the module memory.

7.6 [DNP Slave Binary Inputs]

This area is to override the class (2) binary input database points.

```
#
# Point#   Class
Start
#    0       1
#    1       2
#    2       3
#    3       0    #Events will never be generated for this point
End
```

7.6.1 Point

This is the information object address of the point.

7.6.2 Class

Class 1 - Highest priority

Class 2 - Middle priority

Class 3 - Lowest priority

0 - Disable.

7.7 [DNP Slave Analog Inputs]

This area is to override the class (3) and deadband for the integer analog input database. The point # is the offset from the start of the analog input database.

```
#
# Point#   Class   Deadband
Start
#    6       1          2000  #points 0-5=class 1, deadband = 1000
#    7       1          2000
#    8       2          1000
End
```

[Section]/Item	[redacted]	Description
[DNP Slave Analog Inputs]	[redacted]	DNP database analog input override values

7.7.1 Point

This is the information object address of the point.

7.7.2 Class

Class 1 - Highest priority

Class 2 - Middle priority

Class 3 - Lowest priority

0 - Disable.

7.7.3 Deadband

A range of values within which the module will avoid generating events.

7.8 [DNP Slave Float Inputs]

This area is to override the class (3) and debased for the single float database. The point # is not the address in the analog database, but is the offset from the start of the single floating-point database.

```
#
# Point#   Class   Deadband
Start
    0       1          100.
    1       2         12.34
    3       0         13.45  #Events will never be generated for this point
    4       2        3000.0  #points 5 to 11=class 1, deadband = 1000.00
End
```

7.8.1 Point

This is the information object address of the point.

7.8.2 Class

Class 1 - Highest priority

Class 2 - Middle priority

Class 3 - Lowest priority

0 - Disable.

7.8.3 Deadband

A range of values within which the module will avoid generating events.

7.9 [Secondary Port]

```
[Secondary Port]
Type                   : S      #' '=Disabled, S=Backup Slave
Baud Rate              : 19200  #Baud rate for port 110-115200
RTS On                 : 10     #0-65535 milliseconds before message
RTS Off                : 0      #0-65535 milliseconds after message
Min Response Delay     : 0      #0-65535 milliseconds before response sent from
slave

# Collision Avoidance parameters
Collision Avoidance    : No     #Use Collision Avoidance (N=No, Y=Yes)
CD Idle Time           : 10     #0-32000 mSec min idle time before transmit
CD Random Time         : 20     #0-32000 mSec random idle time before transmit
CD Time Before Receive : 6      #0-65535 milliseconds before receive
```

[Section]/Item	**Value**	**Range**	**Description**
[Secondary Port]			Definitions for secondary port on module
Type:		S or blank	This parameter defines the functionality of the secondary port on the module. S = back-up DNP slave port to the primary port. Any other value will disable the port.
Baud Rate:		Baud rate value	Secondary DNP Port Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 384 (38400) , 576 (57600), 115 (115200)
RTS On:		0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted between asserting the RTS modem line and the actual transmission of the data.
RTS Off:		0 to 65535 milliseconds	This value represents the number of 1 ms increments to be inserted after the last character of data is transmitted before the RTS modem line is dropped.
Min Response Delay:		0 to 65535 milliseconds	Minimum time between receiving a request and transmitting a response. Allows master time to disable transmitter on an RS-485 network.
Collision Avoidance:		Y or N	This parameter defines if collision avoidance will be utilized on the primary DNP slave port.
CD Idle Time:		0 to 32000	Defines the minimum number of milliseconds to wait before transmitting a message after the CD signal is recognized as low.
CD Random Time:		0 to 32000	Defines the range of random time to be added to the CD Idle Time before a message will be transmitted from the slave.
CD Time Before Receive:		0 to 65535	Defines the number of milliseconds to wait before receiving characters after the CD signal is recognized as high.

7.9.1 Type

S or blank

This parameter defines the functionality of the secondary port on the module.

S = back-up DNP slave port to the primary port.

Any other value will disable the port.

7.9.2 Baud Rate

Baud rate value

DNP Port Baud Rate: 300, 600, 1200, 2400, 4800, 9600, 19200, 384 (38400) , 576 (57600), 115 (115200)

7.9.3 RTS On

0 to 65535 milliseconds

This value represents the number of 1 ms increments to be inserted between asserting the RTS modem line and the actual transmission of the data.

7.9.4 RTS Off

0 to 65535 milliseconds

This value represents the number of 1 ms increments to be inserted after the last character of data is transmitted before the RTS modem line is dropped.

7.9.5 Min Response Delay

0 to 65535 milliseconds

Minimum time between receiving a request and transmitting a response. Allows master time to disable transmitter on an RS-485 network.

7.9.6 Collision Avoidance

Y or N

This parameter defines if collision avoidance will be utilized on the primary DNP slave port.

7.9.7 CD Idle Time

0 to 32000

Defines the minimum number of milliseconds to wait before transmitting a message after the CD signal is recognized as low.

7.9.8 CD Random Time

0 to 32000

Defines the range of random time to be added to the CD Idle Time before a message will be transmitted from the slave.

7.9.9 CD Time Before Receive

0 to 65535

Defines the number of milliseconds to wait before receiving characters after the CD signal is recognized as high.

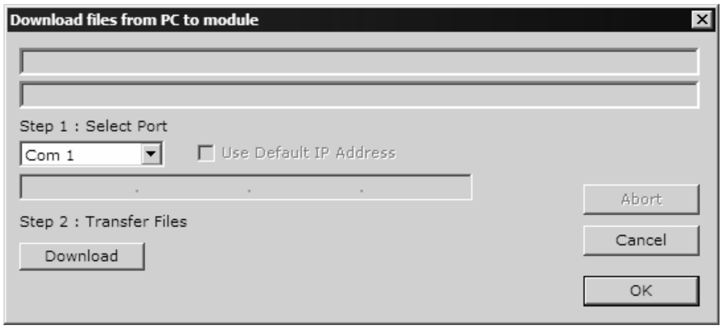
7.10 Download the Project to the Module

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-DNPS module.

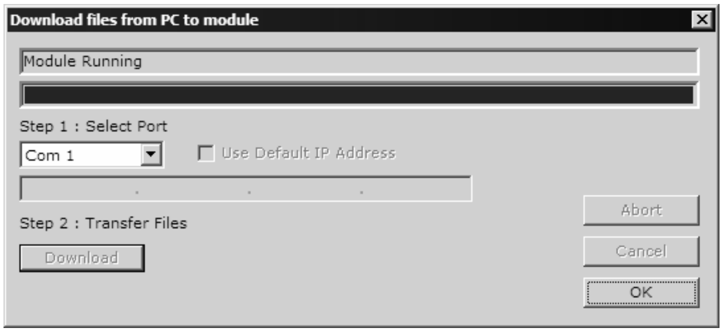
2 Open the **Project menu,** and then choose **Module / Download**. The program will scan your PC for a valid com port (this may take a few seconds). When PCB has found a valid com port, the following dialog box will open.



Com 1

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

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



Module Running

Com 1

7.11 Verification and Troubleshooting

You can now verify that the module is configured properly by viewing parameters that you specified in the configuration file. This is done using the module's Main Menu. If you are not already at the Main menu, press **[Shift][/].**

Use the database menu to verify that data appears in registers that you've mapped. Refer to Diagnostics and Troubleshooting (page 77) for information on accessing information on the operation of the module.

Use the DNP Statistics screen to view the communication status. Refer to Viewing DNP Communication Status for more information.

8 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.

8.1 Reading Status Data from the Module

The PTQ-DNPS module returns a Status Data block that can be used to determine the module's operating status. This data is located in the module's database status database and error status list. This data is transferred to the Quantum / Unity processor read blocks with an identification code of 100. For a complete listing of the status data object, refer to the Installing and Configuring the Module section.

8.1.1 Required Hardware

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

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

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

8.1.2 The Configuration/Debug Menu

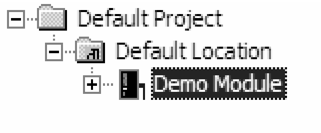
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

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

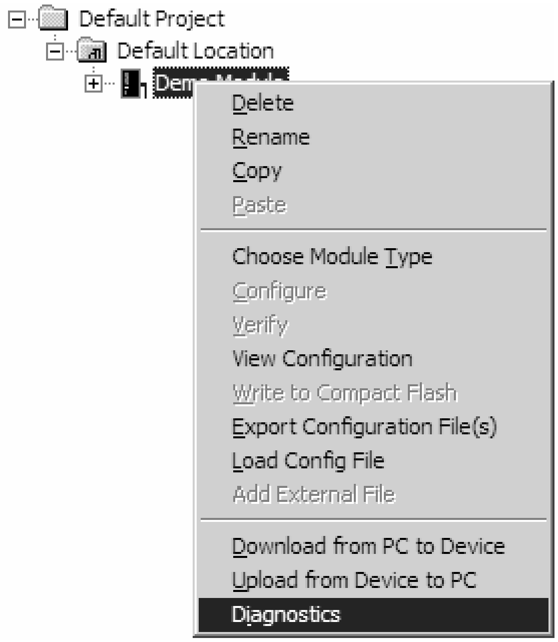
Using the Diagnostic Window in ProSoft Configuration Builder

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

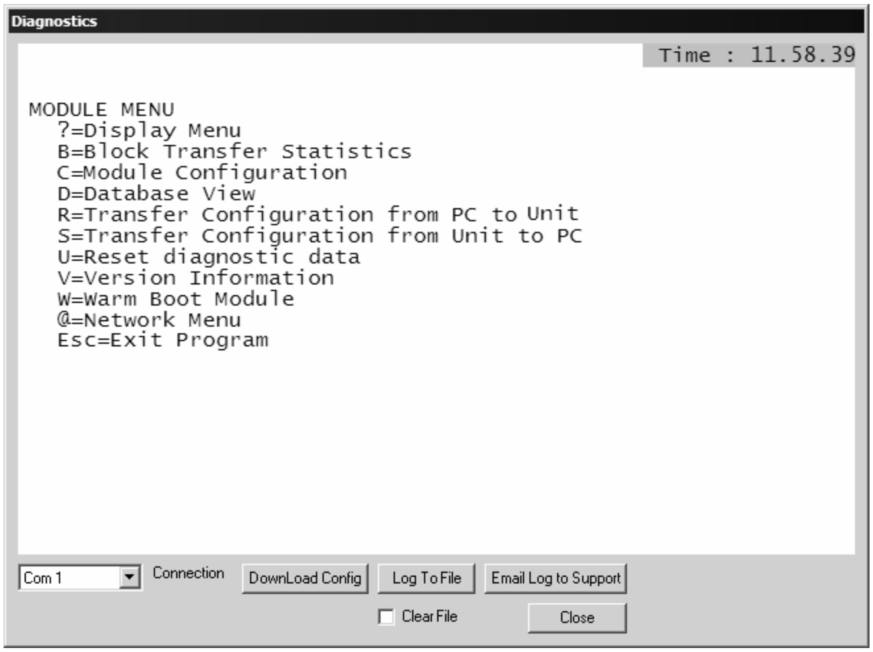
1 Start PCB program with the application file to be tested. Right click over the module icon.



2 On the shortcut menu, choose Diagnostics.



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



```
                                                  Time : 11.58.39

MODULE MENU
  ?=Display Menu
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  R=Transfer Configuration from PC to Unit
  S=Transfer Configuration from Unit to PC
  U=Reset diagnostic data
  V=Version Information
  W=Warm Boot Module
  @=Network Menu
  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:

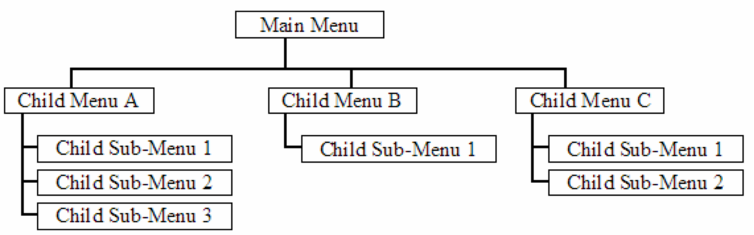
1 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.
2 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

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

Navigation

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

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



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

Keystrokes

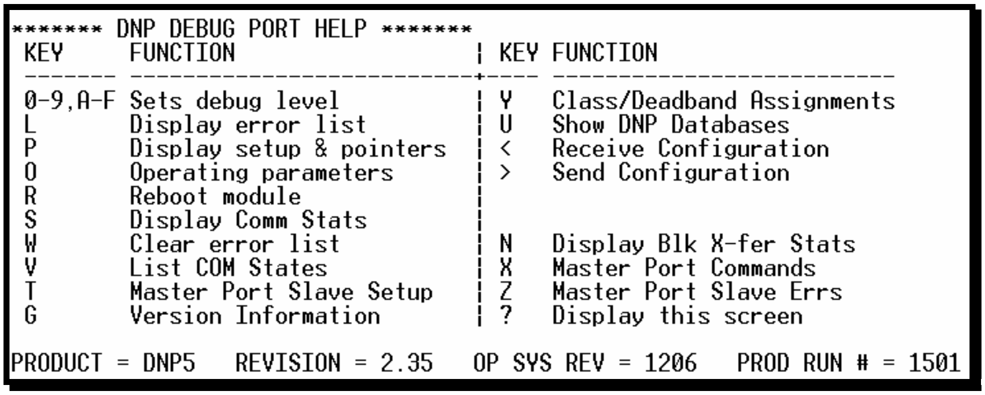
The keyboard commands on these menus are almost always non-case sensitive. You can enter most commands in lower case or capital letters.

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

Also, take care to distinguish capital letter **[I]** from lower case letter **[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.

8.1.3 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:



```
******* DNP DEBUG PORT HELP *******
 KEY     FUNCTION                   ¦ KEY FUNCTION
 ------- ---------------------------+---- --------------------------
 0-9,A-F Sets debug level           ¦ Y   Class/Deadband Assignments
 L       Display error list         ¦ U   Show DNP Databases
 P       Display setup & pointers   ¦ <   Receive Configuration
 O       Operating parameters       ¦ >   Send Configuration
 R       Reboot module              ¦
 S       Display Comm Stats         ¦
 W       Clear error list           ¦ N   Display Blk X-fer Stats
 V       List COM States            ¦ X   Master Port Commands
 T       Master Port Slave Setup    ¦ Z   Master Port Slave Errs
 G       Version Information        ¦ ?   Display this screen

PRODUCT = DNP5   REVISION = 2.35   OP SYS REV = 1206   PROD RUN # = 1501
```

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.

Setting the Debug Level

You can increase or decrease the level of debug messages sent from the module to the Debug Menu. The following table shows the type of debugging information for each key **[0]** to **[9]**, **[A]** to **[F]**

Key	None	DNP Statistics	Data Link Layer Messages	DPA Level Messages
0	X			
1		X		
2			X	
3		X	X	
4				
5		X		
6			X	
7		X	X	
8				X
9		X		X
A			X	X
B		X	X	X
C				X
D		X		X
E			X	X
F		X	X	X

Viewing the Error List

Press **[L]** to display the last 60 errors for the DNP slave port. Refer to the error list section of the user manual to interpret each error recorded by the module.

If there are no errors present for the module, the message "NO ERRORS FOR SYSTEM!" is displayed.

Viewing DNP Set Up & Pointers

Press **[P]** to display the memory allocation and the database setup parameters.

Viewing Operating Parameters

Press **[O]** to view the DNP Protocol setup information (Operating Parameters) for 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 **[R]** from the Main Menu to warm boot (restart) the module. This command will cause the program to exit and reload, refreshing configuration parameters that must be set on program initialization. Only use this command if you must force the module to re-boot.

Viewing Comm Stats

Press **[S]** to view the communication status for the DNP port.

Clearing the Error List

Press **[W]** to clear the error list. Use this command after viewing the error list (page 81) to delete the current list of errors and start a new list.

Viewing COM States

Press **[V]** to view the current state of the DNP application port and the port configuration information.

Viewing Version Information

Press **[G]** 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.

Opening the Class Assignment Menu

Press **[Y]** to view the class and deadband override values for the binary, analog, float and double input DNP database.

Opening the DNP Database View Menu

Press **[U]** to open the DNP Database View Menu. This menu allows you to view all data associated with the DNP Server driver. For more information about the commands on this menu, refer to DNP Database View Menu (page 84).

Receiving the Configuration File

Press **[<] (Shift Comma)** 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 **[>] (Shift Period)** 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 Block Transfer Statistics

Press **[N]** 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.

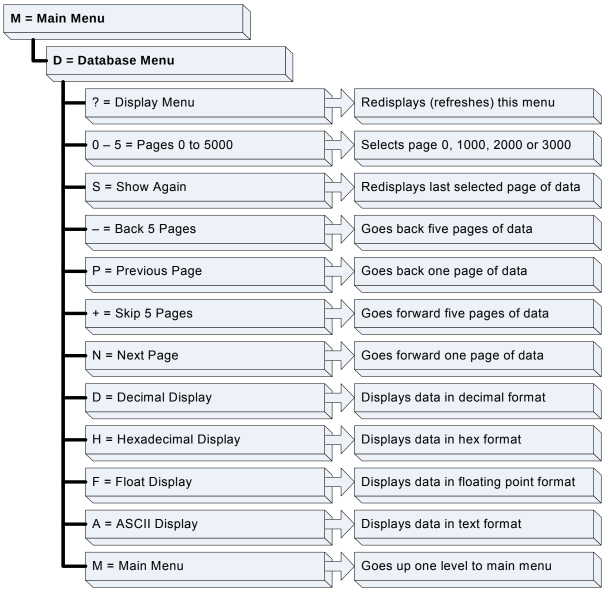
Tip: Repeat this command at one-second intervals to determine the number of blocks transferred each second.

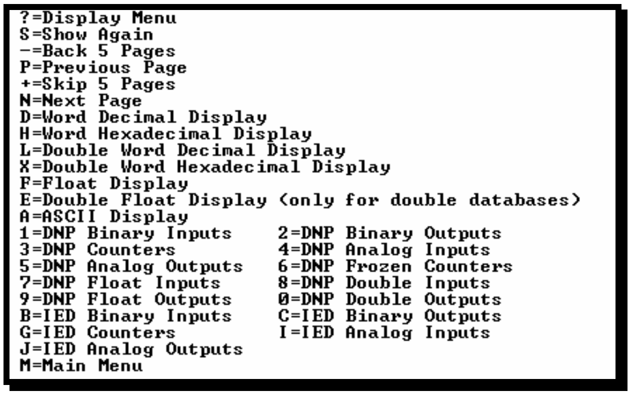
Viewing DNP Server Data

Press **[I]** to view all data associated with the DNP server driver.

8.1.4 DNP Database View Menu

Use this menu command to view the current contents of the selected database. Press **[?]** to view a list of commands available on this menu.

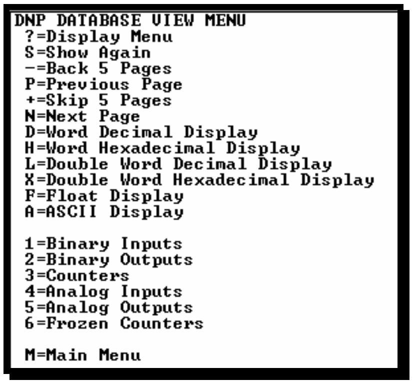

M = Main Menu
D = Database Menu
? = Display Menu
Redisplays (refreshes) this menu
0 – 5 = Pages 0 to 5000
Selects page 0, 1000, 2000 or 3000
S = Show Again
Redisplays last selected page of data
– = Back 5 Pages
Goes back five pages of data
P = Previous Page
Goes back one page of data
+ = Skip 5 Pages
Goes forward five pages of data
N = Next Page
Goes forward one page of data
D = Decimal Display
Displays data in decimal format
H = Hexadecimal Display
Displays data in hex format
F = Float Display
Displays data in floating point format
A = ASCII Display
Displays data in text format
M = Main Menu
Goes up one level to main menu



```
?=Display Menu
S=Show Again
-=Back 5 Pages
P=Previous Page
+=Skip 5 Pages
N=Next Page
D=Word Decimal Display
H=Word Hexadecimal Display
L=Double Word Decimal Display
X=Double Word Hexadecimal Display
F=Float Display
E=Double Float Display (only for double databases)
A=ASCII Display
1=DNP Binary Inputs      2=DNP Binary Outputs
3=DNP Counters           4=DNP Analog Inputs
5=DNP Analog Outputs     6=DNP Frozen Counters
7=DNP Float Inputs       8=DNP Double Inputs
9=DNP Float Outputs      0=DNP Double Outputs
B=IED Binary Inputs      C=IED Binary Outputs
G=IED Counters           I=IED Analog Inputs
J=IED Analog Outputs
M=Main Menu
```

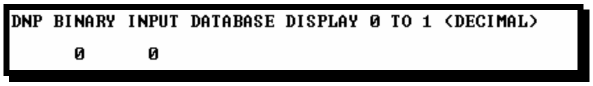
Viewing Data Type Databases

Press **[D]** from the DNP menu, then hold down the **[Shift]** key and press the ***/*** key.



```
DNP DATABASE VIEW MENU
 ?=Display Menu
 S=Show Again
 -=Back 5 Pages
 P=Previous Page
 +=Skip 5 Pages
 N=Next Page
 D=Word Decimal Display
 H=Word Hexadecimal Display
 L=Double Word Decimal Display
 X=Double Word Hexadecimal Display
 F=Float Display
 A=ASCII Display

 1=Binary Inputs
 2=Binary Outputs
 3=Counters
 4=Analog Inputs
 5=Analog Outputs
 6=Frozen Counters

 M=Main Menu
```

Use the number keys 1 to 6 to select the display of the data type you wish to view. For example, if the '1' key is pressed, the following is displayed:



```
DNP BINARY INPUT DATABASE DISPLAY 0 TO 1 (DECIMAL)
      0      0
```

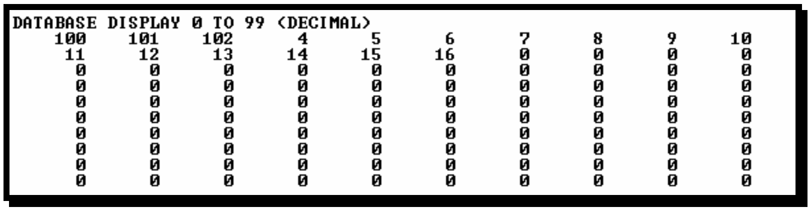
Viewing Register Pages

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



```
DATABASE DISPLAY 0 TO 99 (DECIMAL)
     100     101     102       4       5       6       7       8       9      10
      11      12      13      14      15      16       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
       0       0       0       0       0       0       0       0       0       0
```

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 back to the previous 500 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.

Skipping 500 Registers of Data

Hold down **[Shift]** and press **[=]** to skip forward to the next 500 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.

Viewing Data in Double Word Decimal Format

Press **[L]** to display the data on the current page in Double Word Decimal format. This is useful for regions of the database that contain Double Word Decimal data.

Viewing Data in Double Word Decimal Format

Press **[X]** to display the data on the current page in Double Word Hexadecimal format. This is useful for regions of the database that contain Double Word Hexadecimal data.

Viewing DNP Binary Inputs

Press **[1]** to view a list of DNP Binary Inputs.

Viewing DNP Binary Outputs

Press **[2]** to view a list of DNP Binary Outputs.

Viewing DNP Counters

Press **[3]** to view a list of DNP Counters.

Viewing DNP Analog Inputs

Press **[4]** to view a list of DNP Analog Inputs.

Viewing DNP Analog Outputs

Press **[5]** to view a list of DNP Analog Outputs.

Viewing DNP Frozen Counters

Press **[6]** to view a list of DNP Frozen Counters.

Viewing DNP Float Inputs

Press **[7]** to view a list of DNP Float Inputs.

Viewing DNP Float Outputs

Press **[9]** to view a list of DNP Float Outputs.

Returning to the Main Menu

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

8.2 LED Status Indicators

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

ProSoft Module	Color	Status	Indication
CFG	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.
P1	Green	On	Data is being transferred between the module and the DNPS network on its DNPS Primary slave port.
		Off	No data is being transferred on the port.
P2	Green	On	Data is being transferred between the module and the DNPS network on its Port 3. This may either be configured as a DNPS master or backup slave port.
		Off	No data is being transferred on the port.
APP	Amber	Off	The PTQ-DNPS is working normally.
		On	The PTQ-DNPS module program has recognized a communication error on one of its DNPS ports.

ProSoft Module	Color	Status	Indication
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

If the APP, BP ACT and OK LEDs blink at a rate of every one-second, this indicates a serious problem with the module. Call ProSoft Technology support to arrange for repairs.

8.2.1 Clearing a Fault Condition

Typically, if the OK LED on the front of the module turns red for more than ten seconds, a hardware problem has been detected in the module, or the program has exited.

To clear the condition, follow these steps:

1. Turn off power to the rack
2. Remove the card from the rack
3. Verify that all jumpers are set correctly
4. If the module requires a Compact Flash card, verify that the card is installed correctly
5. Re-insert the card in the rack and turn the power back on
6. Verify the configuration data being transferred to the module from the Quantum / Unity processor.

If the module's OK LED does not turn green, verify that the module is inserted completely into the rack. If this does not cure the problem, contact ProSoft Technology Support.

8.2.2 Troubleshooting

Use the following troubleshooting steps if you encounter problems when the module is powered up. If these steps do not resolve your problem, please contact ProSoft Technology Technical Support.

Processor Errors

Problem Description	Steps to take
Processor Fault	Verify that the module is plugged into the slot that has been configured for the module. Verify that the slot in the rack configuration has been set up correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. Verify that all modules in the rack are configured in the ladder logic.

Module Errors

Problem Description	Steps to take
BP ACT LED remains off or blinks slowly	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in Run mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write block data transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is configured in the processor.
OK LED remains red	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert the card in the rack, and then restore power to the rack.

9 Reference

In This Chapter

9.1 Product Specifications

The ProTalk DNPS 3.0 Slave Communication Module (PTQ-DNPS) is a Quantum backplane compatible module that allows Quantum processors to interface easily with DNP 3.0 compatible devices and hosts. Devices commonly supporting the protocol include relays, breakers, sub-station Communication Modules and other serial devices most commonly associated with power monitoring.

9.1.1 Features and Benefits

The module supports DNP Subset Level 2 features and some of the Level 3 features allowing the many SCADA and field devices supporting the DNP protocol to be integrated into the Quantum platform. The module acts as an input/output module between the DNP network and the Modicon backplane. The data transfer from the Quantum processor is asynchronous from the actions on the DNP network. Databases are user defined and stored in the module to hold the data required by the protocol.

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

9.1.2 General Specifications

- Single Slot - Quantum backplane compatible

- The module is recognized as an Options module and has access to PLC memory for data transfer
- Configuration data is stored in non-volatile memory in the ProTalk module
- Configuration software for Microsoft Windows XP, 2000 and NT is included with the 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.
    - UnityPro XL
    - Concept
    - ProWORX
- Quantum data types supported: 3x, 4x
- High speed data transfer across backplane provides quick data update times.
- Sample ladder file available.

9.1.3 Hardware Specifications

Specification	Value
Backplane Current Load	800 mA @ 5 V
Operating Temperature	0 to 60°C (32 to 140°F)
Storage Temperature	-40 to 85°C (-40 to 185°F)
Relative Humidity	5% to 95% (non-condensing)
Vibration	Sine vibration 4-100 Hz in each of the 3 orthogonal axes
Shock	30G, 11 mSec. in each of the 3 orthogonal axes
LED Indicators	Module Status Backplane Transfer Status Serial Port Activity LED Serial Activity and Error LED Status
Configuration Serial Port (PRT1)	DB-9M PC Compatible RS-232 only No hardware handshaking
Application Serial Ports	(PRT2, PRT3) DB-9M PC Compatible RS-232/422/485 jumper selectable RS-422/485 screw termination included RS-232 handshaking configurable 500V Optical isolation from backplane

9.1.4 Functional Specifications

The PTQ-DNPS module supports the DNP 3.0 protocol with a minimum of Level 2 functionality. DNP protocol Subset Definitions for the Slave drivers are available in the module's User Manual.

The module has two DNP protocol ports that can be user configured to operate in a Slave/Slave redundant port configuration.

The module has 4000 words of user defined internal register space that are accessible to the protocol driver and to the Quantum processor memory.

Redundant Slave Port Operation

When configured in the Slave/Slave port configuration, the module's slave ports operate in a primary and secondary fashion. In this mode, a single host polls the module via redundant physical layer connections. Several methods are supported to automatically switch between the primary and secondary slave ports.

DNP 3.0 Slave Protocol Specifications

The DNP Slave port(s) accepts DNP commands to control and monitor data stored in the module's DNP Slave databases.

- Report-by-Exception data is logged to the module's database
- Supports unsolicited messaging
- Each DNP point type is user configurable by point
- Total point counts must be configured so that Class 0 responses do not exceed 2048 bytes in size
- Class assignments are completely user-definable on a Type and point basis (BI, AI, FI, DI point types)
- The analog inputs are class and deadband configurable on a point basis for all formats (integer, float, double float)
- Supports clock synchronization from a master or from the Quantum
- Support for four octet-strings are supported (object type 110) in the slave driver to return version and other module information
- Up to 400 events are stored for Floats, Binary In, Analog In and Double Inputs
- In addition to the module generated events, AI and BI events can be generated in the Quantum and transferred to the module (useful with external timestamping hardware)
- Configurable event buffer transmission threshold based on count and/or time since last event transmission
- Collision avoidance algorithm per DNP organization for redundant port switching (redundant slave mode)
- Special modem AT command string and timing support for dialing out on redundant port (redundant slave mode)

9.2 Functional Overview

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

9.2.1 General Concepts

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

Module Power Up

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

1 Initialize hardware components

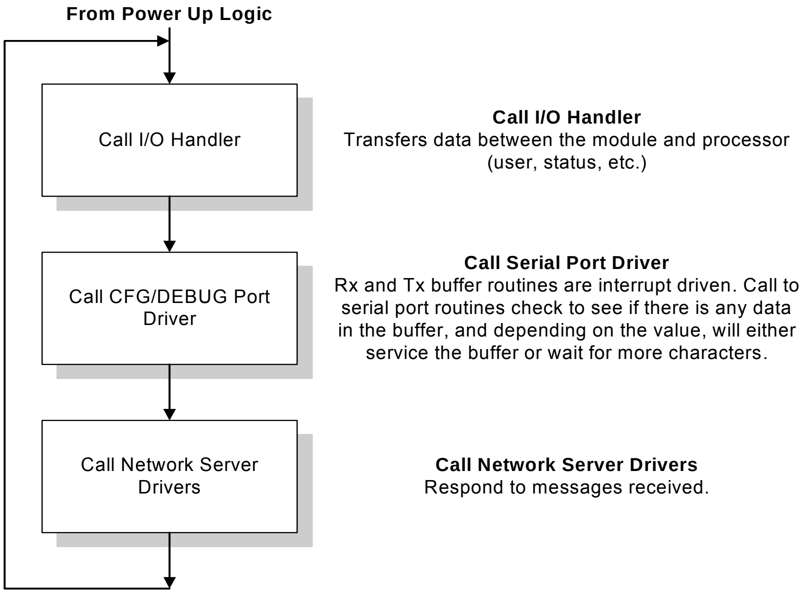
- Initialize Quantum / Unity backplane driver
- Test and clear all RAM
- Initialize the serial communication ports

2 Read configuration file from Random Access Memory
3 Enable Slave Driver

After the module has received the configuration, the module will begin communicating with other nodes on the network, depending on the configuration.

Main Logic Loop

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


From Power Up Logic
Call I/O Handler
Call CFG/DEBUG Port Driver
Call Network Server Drivers
Call I/O Handler
Transfers data between the module and processor (user, status, etc.)
Call Serial Port Driver
Rx and Tx buffer routines are interrupt driven. Call to serial port routines check to see if there is any data in the buffer, and depending on the value, will either service the buffer or wait for more characters.
Call Network Server Drivers
Respond to messages received.

Backplane Data Transfer

The current version of the PTQ-DNPS backplane driver (version 2.0) uses a Large I/O model, which differs from previous versions of the backplane driver in that it transfers all of the data in the input and output databases between the module and the processor on every scan.

The [Backplane Configuration] section of the configuration file defines the starting register for input and output. The typical configuration is 5000 words read (3x/ % IW) and 5000 words write (4x / % MW).

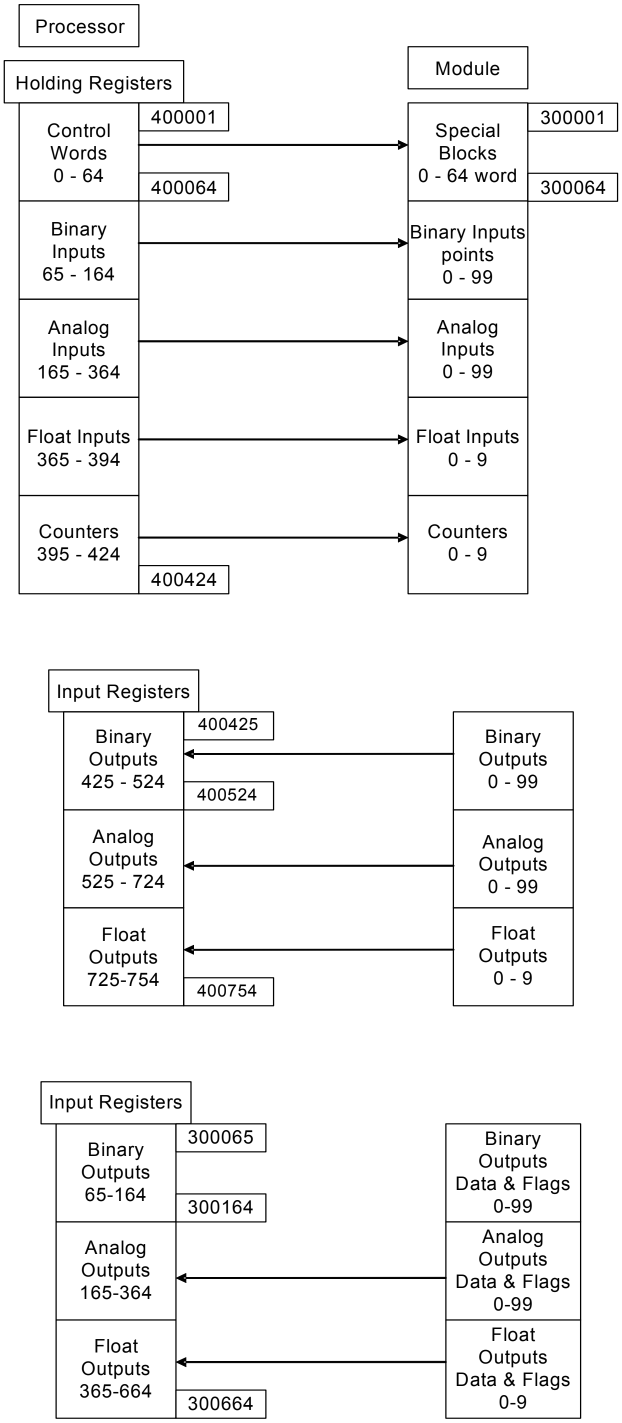
The following parameters control how much database content will be transferred between the Processor and the Module:

- Binary Inputs
- Analog Inputs
- Float Inputs
- Counters
- Binary Outputs
- Analog Outputs
- Float Outputs

Example 1 (Sample Default Values)

```
Binary Inputs       : 100  #0-512 point count to hold BI data
Analog Inputs       : 100  #0-512 points of analog input data
Float Inputs        :  10  #0-128 points of floating-point format data
Counters            :  10  #0-128 points of counter data
Binary Outputs      : 100  #0-512 point count to hold BO data
Analog Outputs      : 100  #0-512 points of analog output data
Float Outputs       :  10  #0-128 points of floating-point format data
```

The following illustration shows the data transfer between processor and Module using the configuration values in Example 1:

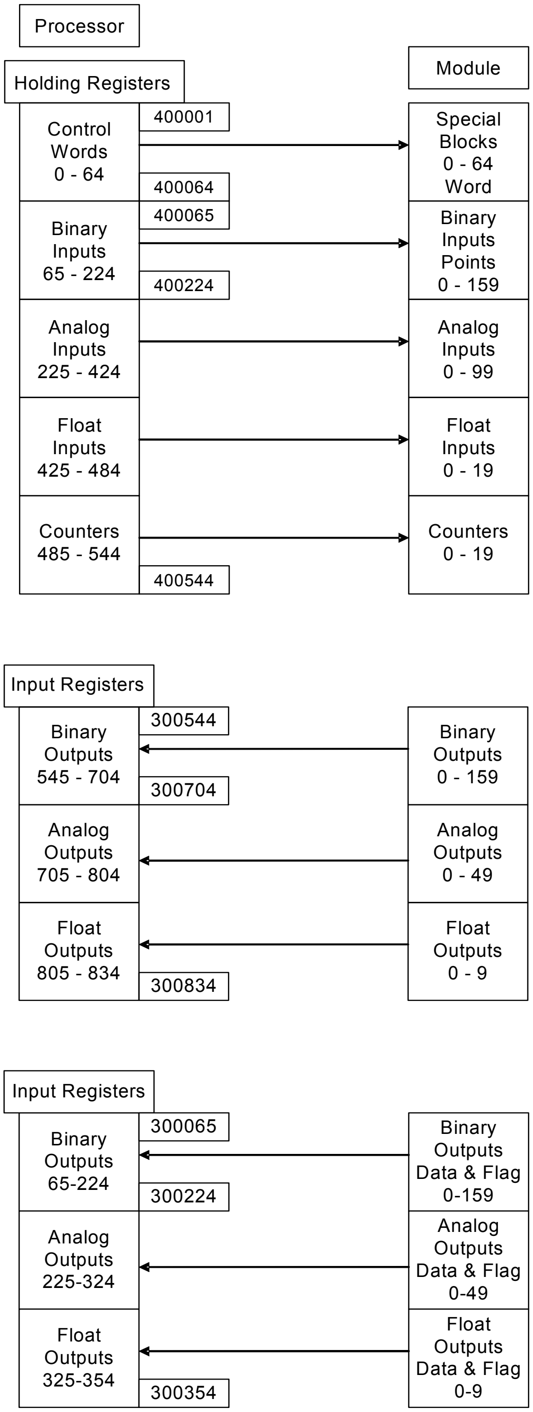


Example 2

```
Binary Inputs      : 160   #0-512 point count to hold BI data
```

```
   Analog Inputs      : 100   #0-512 points of analog input data
   Float Inputs       :  20   #0-128 points of floating-point format data
   Counters           :  20   #0-128 points of counter data
   Binary Outputs     : 160   #0-512 point count to hold BO data
   Analog Outputs     :  50   #0-512 points of analog output data
   Float Outputs      :  10   #0-128 points of floating-point format data
```

The following illustration shows the data transfer between processor and Module using the configuration values in Example 2:



9.2.2 Input and Output Data Blocks

Status Block 9250

Block 9250 identification code requests the module's status data. The module supports a buffer queue of 99 events per data type. The application can verify the status of the queue (free space in the queue) through the module's status data (Block 9250). When the queue is full, the module will delete the older event in the queue if a new event is received.

There are two ways to request status data:

- Read the data in the database, starting at word 4000. Refer to Error Status Table for detailed information.
- Request the entire status block with a block 9250 block request.

The module responds to a valid block 9250 request with a block containing the requested status data. The status data area for the module starts at address 4000 in the database.

Block Format for Read

The format of this block is shown in the following table:

Word Offset	**Variable Name**	**Description**
0	Sequence number	This field contains a new value each time the block is handled.
1	Status request block	This word will contain the value of 9250 when the operation is complete.
2	Current DNP Slave Port status	This value represents the current value of the error code for the port. This value will only be valid if the port is configured as a slave. The possible values are described in the application documentation.
3	DNP Slave Port last transmitted error code	This value represents the last error code transmitted to the master by this slave port.
4	DNP Slave Port total number of message frames received by slave	This value represents the total number of message frames that have matched this slaves address on this port. This count includes message frames which the slave may or may not be able to parse and respond.
5	DNP Slave Port total number of response message frames sent from slave	This value represents the number of good (non-error) responses that the slave has sent to the master on this port. The presumption is that if the slave is responding, the message was good. Note: This is a frame count.
6	DNP Slave Port total number of message frames seen by slave	This value represents the total number of message frames received by the slave, regardless of the slave address.
7	DNP Slave synchronization error count (Physical Layer Error)	This value counts the number of times a sync error occurs. The error occurs when extra bytes are received before the start bytes (0x05 and 0x64) are received.
8	DNP Slave overrun error count (Physical Layer Error)	This value counts the number of times the overrun error occurs. This error occurs when the mainline Data Link Layer routine cannot read the data received on the communication port before it is overwritten.

Word Offset	Variable Name	Description
9	DNP Slave length error count (Physical Layer Error)	This value counts the number of times an invalid length byte is received. If the length of the message does not match the length value in the message, this error occurs.
10	DNP Slave bad CRC error (Data Link Layer Error)	This value counts the number of times a bad CRC value is received in a message.
11	DNP Slave user data overflow error (Transport Layer Error)	This value counts the number of times the application layer receives a message fragment buffer which is too small.
12	DNP Slave sequence error (Transport Layer Error)	This value counts the number of times the sequence numbers of multi-frame request fragments do not increment correctly.
13	DNP Slave address error (Transport Layer Error)	This value counts the number of times the source addresses contained in a multi-frame request fragments do not match.
14	DNP Slave Binary Input Event count	This value contains the total number of binary input events which have occurred.
15	DNP Slave Binary Input Event count in buffer	This value represents the number of binary input events which are waiting to be sent to the master.
16	DNP Slave Analog Input Event count	This value contains the total number of analog input events which have occurred.
17	DNP Slave Analog Input Event count in buffer	This value represents the number of analog input events which are waiting to be sent to the master.
18	DNP Slave bad function code error (Application Layer Error)	This value counts the number of times a bad function code for a selected object/variation is received by the slave device.
19	DNP Slave object unknown error (Application Layer Error)	This value counts the number of times a request for an unsupported object is received by the slave device.
20	DNP Slave out of range error (Application Layer Error)	This value counts the number of times a parameter in the qualifier, range or data field is not valid or out of range.
21	DNP Slave message overflow error (Application Layer Error)	This value counts the number of times an application response message from the slave is too long to transmit.
22	DNP Slave multi-frame message from DNP Master error (Application Layer Error)	This value counts the number of times the slave receives a multi-frame message from the master. The application does not support multi-frame master messages.
23	Total blocks transferred	Total BTR/BTW or side-connect interface transfers attempted by the module.
24	Successful blocks transferred	This value represents the total number of transfer operations between the PLC and module that are successful.
25	Total errors in block transfer	Total number of transfers that resulted in an error condition.
26	Total BTR or write errors	Total number of BTR or write transfers that resulted in an error.

Word Offset	Variable Name	Description
27	Total BTW or read errors	Total number of BTW or read transfers that resulted in an error.
28	Block number error	Number of BTW requests that resulted in an incorrect BTW identification code.
29	Continuous block error counter	Count of sequential data transfer errors. When this value exceeds that specified for the data transfer operation, the error flag below will be set.
30	Block transfer error flag	This flag indicates that data is not being successfully transferred between the PLC and the module. This flag corresponds to the Device Trouble IIN bit.
31	Configuration Type	This is a coded field that defines the configuration of the module. The codes are as follows: 0=Single Slave Configuration or 1=Dual Slave Configuration with one slave active while other is backup.
32 to 33	Product Name (ASCII)	These two words contain the product name of the module in ASCII format.
34 to 35	Revision (ASCII)	These two words contain the product revision level of the firmware in ASCII format.
36 to 37	Operating System Revision (ASCII)	These two words contain the module's internal operating system revision level in ASCII format.
38 to 39	Production Run Number (ASCII)	These two words contain the production 'batch' number for the particular chip in the module in ASCII format.
40	Scan Counter LSB	Program scan counter
41	Scan Counter MSB	
42	Free Memory LSB	Free memory in module
43	Free Memory MSB	
44	DNP Slave Port Transmit State	Value of the DNP Slave state machine for transmit.
45	DNP Float Event Count	Total number of events generated for analog floating-point input data points.
46	DNP Double Event Count	Total number of events generated for analog double, floating-point input data points.
47	Processor State	This status register will contain a value of 1 if the processor is in run mode and 0 if it is not.
48	I/O parameters set	This status register contains a value of 0 if the I/O sizes have been read from the processor and 1 if not.
49	Hot-Standby status word	This status register contains the hot-standby status word.

Status Block 9251

Block 9251 identification code requests the module's error list data. The module supports a buffer queue of 99 events per data type.

The module responds to a valid block 9251 request with a block containing the requested error list data.

Block Format for Write

The format of this block is shown in the following table:

Word Offset	Variable Name	Description
0	Sequence number	This field contains a new value each time the block is handled.
1	Status Request Control Word	This word will contain the value of 9251 when the operation is complete.
2	Error_List[0]	First value in error list
3	Error_List[1]	Second value in error list
-	-	-
61	Error_List[59]	Last value in error list

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Sequence Number	This field contains a new value each time the block is handled.
1	Block ID	This field contains the value of 9250 identifying the block type to the module.
2 to 63	Not Used	Not Used

Block 9970 - Set Quantum / Unity Processor Time using Module Time

This block transfers the module's time to the Quantum / Unity processor. Ladder logic must be used to set the processor's clock using the data received. The format of the block sent from the Quantum / Unity processor has the following format:

Word Offset in Block	Data Field(s)	Description
0	Sequence Number	This field contains a new value each time the block is handled.
1	Block ID	This field contains the value of 9970 identifying the block type to the module.

The module responds to the request with a read block 9970 with the following format:

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block Write ID	This is the next block requested by the module.
2	Year	This field contains the four-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	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
6	Minute	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 Field(s)	Description
7	Seconds	This field contains the second 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	Remote Time Synchronization	This field informs the PLC if the date and time passed has been synchronized with a remote DNPS master device on the module's slave port.

Block 9971 - Set Module's Time using the Quantum / Unity Processor Time

This block sets the clock in the module to match the clock in the Quantum / Unity processor. If the Quantum / Unity processor sends a block 9971, the module will set its time using the data contained the block. The format of the block is shown in the following table:

Block Format from Processor

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the block identification code of 9971 for the block.
2	Year	This field contains the four-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	Hour	This field contains the hour value for the new time. Valid entry for this field is in the range of 0 to 23.
6	Minute	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 second 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.

Block Format from Module

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the block identification code of 9971 for the block.

Block 9958 - Send Binary Input Events

Block 9958 identification code is used by the PLC to send a set of binary input events to the module.

Block Format from Processor

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the value of 9958 identifying the event block to the module.
2	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 10.
3	Sequence Counter	This field holds the sequence counter for each 9958 block transfer. This synchronizes and confirms receipt of the block by the module.
4	DNP Binary Input Data point	This is the data point in the DNP binary input database represented by the event.
5	Event Value	This word contains the new value for the point and the event. Only the LSB byte portion of the word is valid for the DNP protocol.
6	Month/Day	Formatted: bits 0 to 4 = Day, bits 8 to 11 = Month. All other bits are ignored.
7	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
8	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.
9	Year	This is the four digit year for the event.
10 to 15		Six words of data for Event #2.
16 to 21		Six words of data for Event #3.
22 to 27		Six words of data for Event #4.
28 to 33		Six words of data for Event #5.
34 to 39		Six words of data for Event #6.
40 to 45		Six words of data for Event #7.
46 to 51		Six words of data for Event #8.
52 to 57		Six words of data for Event #9.
58 to 63		Six words of data for Event #10.

To insure the receipt of this block of information, the module returns a block 9958 with the sequence counter set to the value of the last successful block 9958 received.

Block Format from Module

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the value of 9958 identifying the event block to the module.
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9958 received.

Block 9959 - Send Analog Input Events

Block 9959 identification code is used by the PLC to send a set of analog input events to the module.

Block Format from Processor

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	This field contains the value of 9959 identifying the event block to the module.
2	Event Count	This field contains the number of events contained in the block. Valid values for this field are 1 to 10.
3	Sequence Counter	This field holds the sequence counter for each 9959 block transfer. This synchronizes and confirms receipt of the block by the module.
4	DNP Analog Input Data point	This is the data point in the DNP analog input database represented by the event.
5	Analog Input Value	This is the new analog input value represented in the event.
6	Month/Day	Formatted: bits 0 to 4 = Day, bits 8 to 11 = Month. All other bits are ignored.
7	Hour/Minute	Formatted: bits 0 to 5 = Minutes, bits 8 to 12 = Hour. All other bits are ignored.
8	Sec/Millisecond	Formatted: bits 0 to 9 = Milliseconds, bits 10 to 15 = Seconds.
9	Year	Four digit year value for event.
10 to 15		Six words of data for Event #2.
16 to 21		Six words of data for Event #3.
22 to 27		Six words of data for Event #4.
28 to 33		Six words of data for Event #5.
34 to 39		Six words of data for Event #6.
40 to 45		Six words of data for Event #7.
46 to 51		Six words of data for Event #8.
52 to 57		Six words of data for Event #9.
58 to 63		Six words of data for Event #10.

To ensure the receipt of this block of information, the module returns a BTR block 9959 with the sequence counter set to the value of the last successful block 9959 received.

Block Format from Module

Word Offset in Block	Data Field(s)	Description
0	Sequence Counter	This field contains a new value each time the block is handled.
1	Block ID	Block identification code for request from PLC by the module.

Word Offset in Block	Data Field(s)	Description
2	Event Count	This field contains the number of events processed by the module.
3	Sequence Counter	This field contains the sequence counter of the last successful block 9959 received.

9.2.3 PTQ-DNPS Application Design

This documentation describes the PTQ-DNPS module configuration and setup as it applies to application design. The design of the entire system must be complete before you attempt to implement this module with a DNP network. This includes:

- definition of all the data types and point counts required for each type,
- all communication parameters required for the network including media type and
- the use of advanced features such as unsolicited messaging.

These must be defined for all master and slave devices on the network. Additionally, the DNP Device Profiles and DNP Subset Definition documents for each device must be reviewed to make sure all the devices will interact on the network as expected. Failure to fully understand these important documents for all devices on the network will usually lead to many problems when implementing the design.

It is important to fully understand the DNP specification as outlined in the Basic Four Documents. These are available to users of the DNP users group. It is recommended that all users of the module have access to these important documents as they define the DNP data types, functions and variations. It will be very difficult to implement the module without an understanding of the protocol and the rules that are defined in the specification. Additionally, potential users should review the DNP Subset and Conformance Test documents and the document that discusses DNP protocol support. These documents provide auxiliary information on the protocol. All of these documents are available to members of the DNP User Group at http://www.dnp.org (http://www.dnp.org). Please check this site for other important information regarding the DNP protocol.

Design

In order to implement a solution using the module, the Quantum processor must be set up using predefined user data structures. This program will interact with the module by sending and receiving data and issuing special control commands.

An internal database in the Quantum processor contains the data to be used by the module and the configuration information is stored in the text file, DNPSPTQ.CFG, stored on the module's non volatile memory. Before you generate the program or layout the data files, you must first design your system. Time spent doing system design at the outset of the project will greatly enhance the success and ease of development of the project.

Designing the system

System design defines the data requirements of the system, communication parameters, and module functionality. The application developer should refer to the person responsible for the DNP master and slave device configurations to verify that the functionality and data types required for the whole system are consistent. Review the DNP Device Profile and DNP Subset documentation for a definition of the level of DNP support offered by the module.

The following topics describe each element of system design.

Data Requirements

This phase of design defines what data elements are to be interfaced in the Quantum processor with the DNP master. The module provides the following data types:

- Digital Input
- Digital Output
- Counter
- Floating Point
- Analog Input
- Analog Output

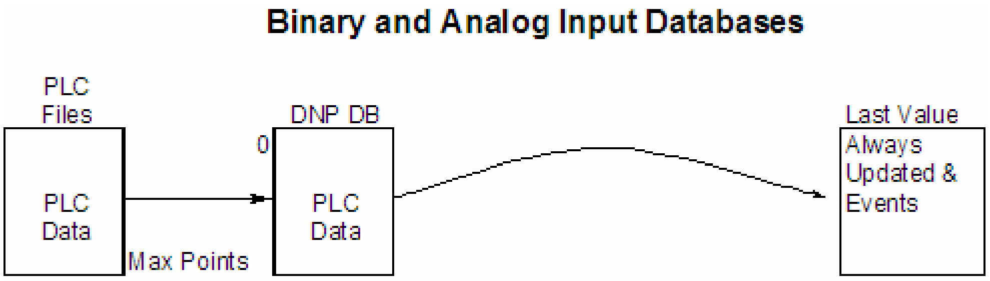
All communications between the DNP master and the PLC is through these data types. Therefore, all data to be used by the system must be contained and configured in one of these data types.

The following illustration shows the databases maintained by the module for the DNP data.

DATA AREA	
DNP DATA	BINARY INPUTS
	ANALOG INPUTS
	FLOAT INPUTS
	COUNTER DATA
	BINARY OUTPUTS
	ANALOG OUTPUTS

The module is responsible for maintaining the databases using data acquired from the PLC and DNP master attached network port.

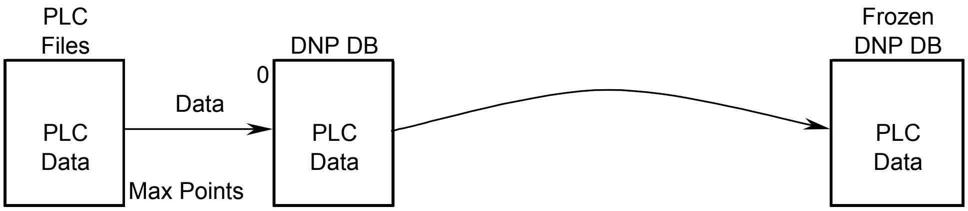
The following illustration shows the interaction of the binary and analog input points with the databases.



All data for these data types is derived from the processor and is passed to the module over the backplane. The module will constantly monitor for changes in this data and generate event messages when point values change. For binary input points, events will be generated on any state change. For analog input points, events will be generated for points that have a current value outside of the user-set deadband based on the last value used for an event.

The following illustration shows the interaction of the counter points with the databases.

Counter Databases

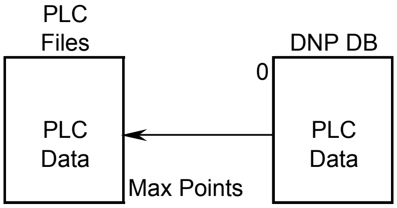


This data is constantly sourced from the processor and placed in the module's internal database. This information is available to the remote master for monitoring. When the module receives a freeze command from the master unit, it will copy the current counter values into the frozen counter database area. The remote master can then monitor this information. If the module receives a counter freeze with reset command, the current counter values will be passed to the frozen counter database and only the module's values will be set to 0.

Note: This data is not sent to the controller, and the zero data be overwritten by the counter data contained in the controller. Therefore, the freeze with reset should not be used with this module. The results will not be as expected. There is no way to guarantee that counts will not be lost during the reset step in the module and controller. As a result, this feature was not implemented in the module.

The following illustration shows the interaction of the binary and analog output points with the databases.

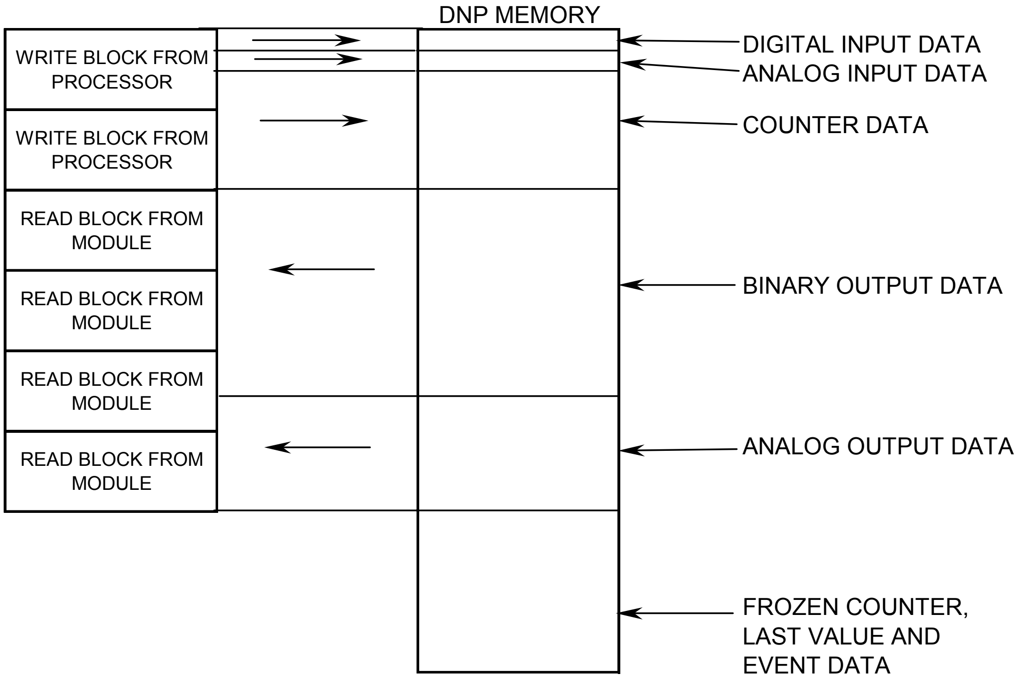
Binary and Analog Output Databases



Output data is sourced from the controlling master station and passed to the processor over backplane from the module. These data are used in the ladder logic to control operations and I/O in the processor.

Data Transfer Interface

The following figure displays the direction of movement of the DNP database data between the module and the processor.

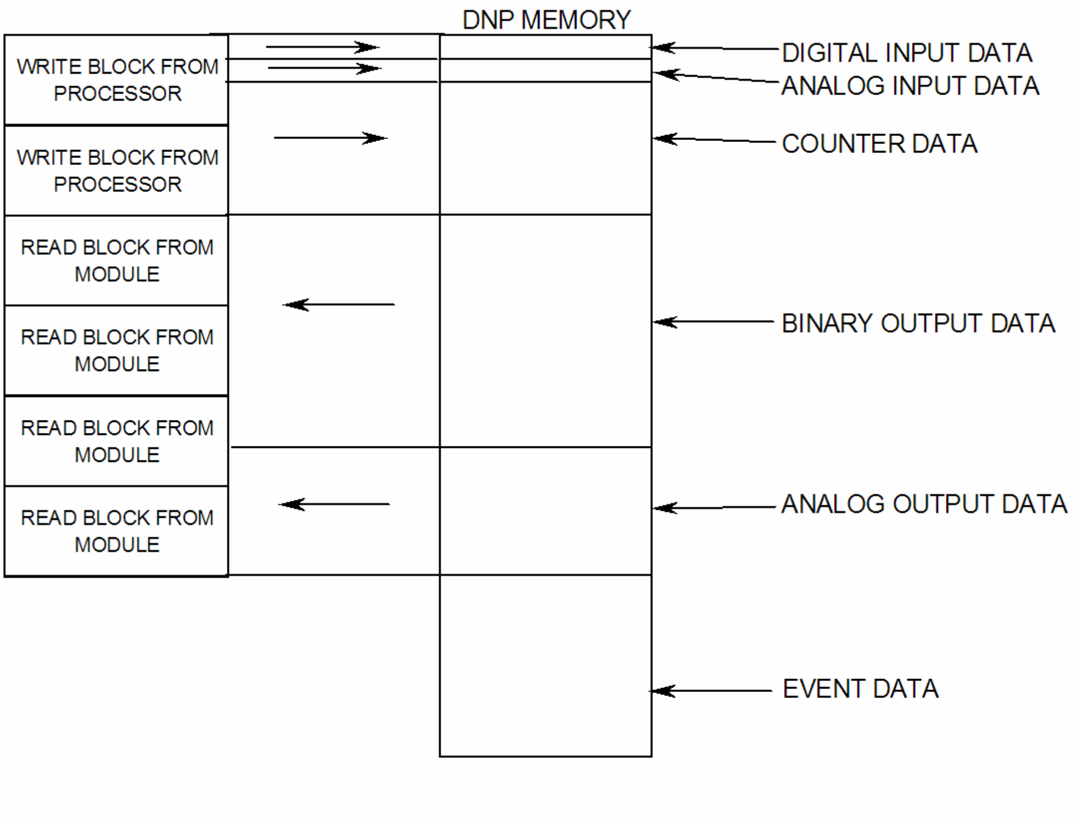


It is important to understand the relationship of the block identifications and the data in the module.

The Reference chapter contains forms to aid in designing your system. They can be used to document the relationship between the point assignments, block identification numbers and the PLC file and offset values and to define the program configuration. Use these forms during your design phase.

Data Transfer Interface

The following figure displays the direction of movement of the DNP database data between the module and the processor.



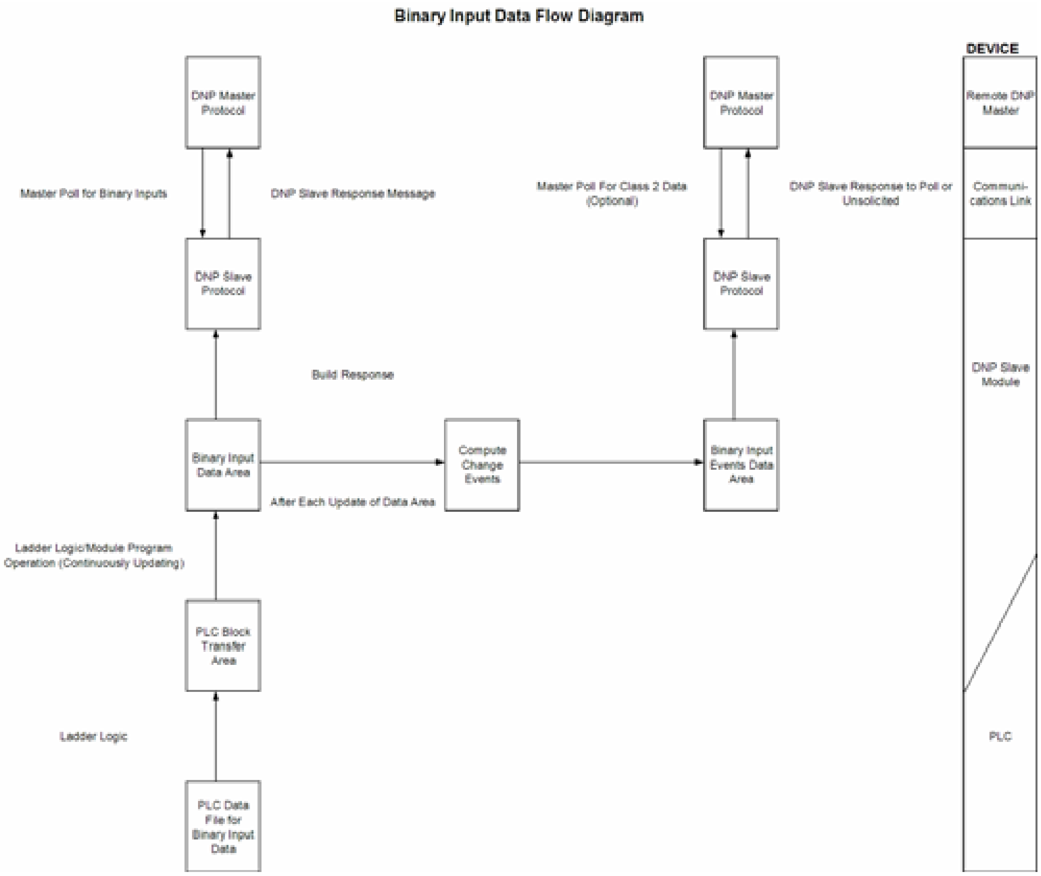
It is important to understand the relationship of the block identifications and the data in the module.

The Reference chapter contains forms to aid in designing your system. They can be used to document the relationship between the point assignments, block identification numbers and the PLC file and offset values and to define the program configuration. Use these forms during your design phase.

DNP Digital Input Data

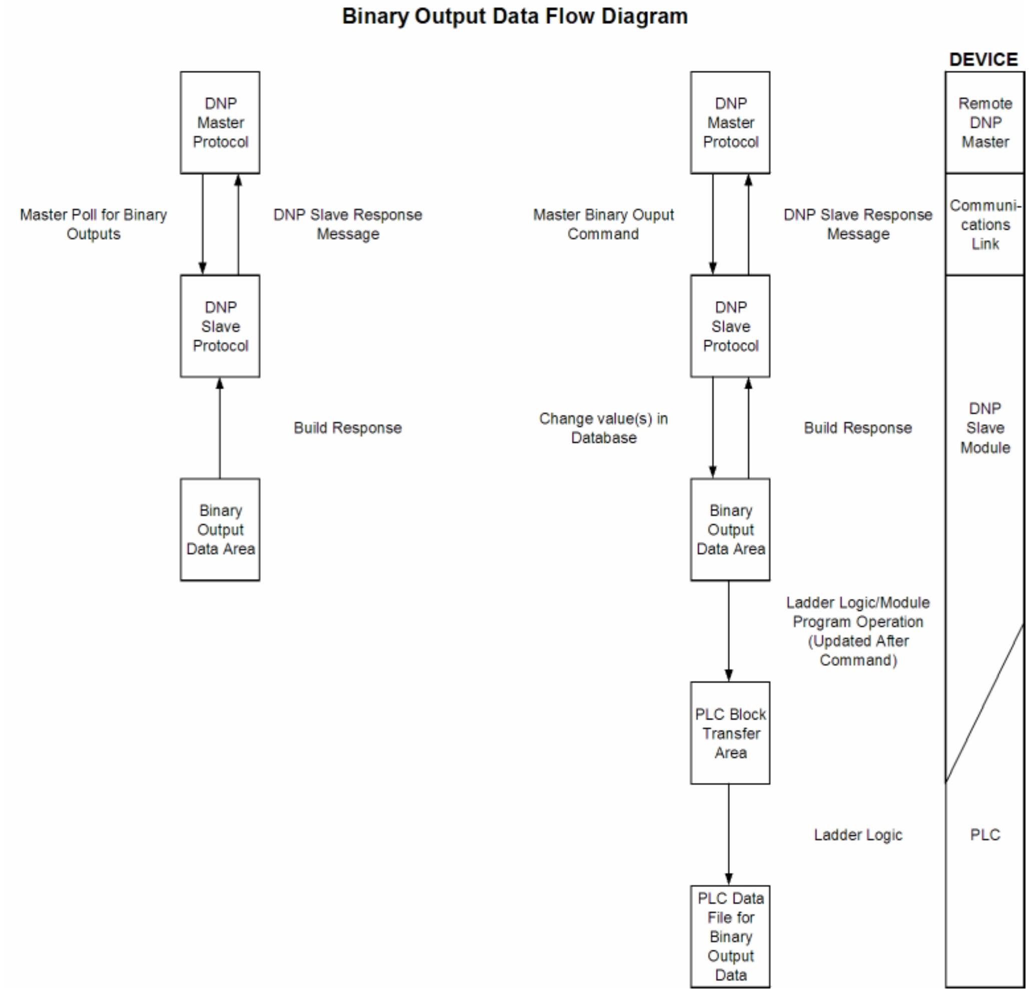
This data type stores the binary value of 1 or 0. The size of this data area is determined from the configuration parameter Binary Inputs (number of words, each containing 1 binary input point). These data are transferred to the module from the PLC using the read operation. Therefore, these data are read-only for the module and the DNP master unit communicating with the module. When the module receives a new block of this data from the PLC, it compares the new values to those currently in the database. If there is a change in any of the data, the module will generate an event message for the points that change.

The remote DNP master unit can read the current status data and the event data from the module. Event messages generated by the module can be retrieved using a poll for Class 2 data, as all digital input events are considered a Class 2 data type. If unsolicited message generation is enabled in the application, the events will automatically be sent by the module to the DNP master unit when the maximum event count for Class 2 data is reached or when the timeout for unsolicited messages is exceeded. A data flow diagram for the digital input data is shown in the following figure.


Binary Input Data Flow Diagram
DEVICE
DNP Master Protocol
DNP Master Protocol
Remote DNP Master
Master Poll for Binary Inputs
DNP Slave Response Message
Master Poll For Class 2 Data (Optional)
DNP Slave Response to Poll or Unsolicited
Communi-cations Link
DNP Slave Protocol
DNP Slave Protocol
Build Response
DNP Slave Module
Binary Input Data Area
Compute Change Events
Binary Input Events Data Area
After Each Update of Data Area
Ladder Logic/Module Program Operation (Continuously Updating)
PLC Block Transfer Area
Ladder Logic
PLC
PLC Data File for Binary Input Data

DNP Digital Output Data

This data type stores digital control and command state data received from the DNP master unit with a value of 1 or 0. The size of this data area is determined from the configuration parameter Binary Outputs (defines number of words, each containing 1 binary output point). These data are transferred from the module to the PLC using the write operation. Therefore, these data are read-only for the PLC, as the PLC cannot directly alter these values in module. It is the responsibility of the DNP master unit to maintain this data. For example, if the DNP master sets a digital point on, it will remain on until the master resets the point. A data flow diagram for the digital output data is shown in the following figure.

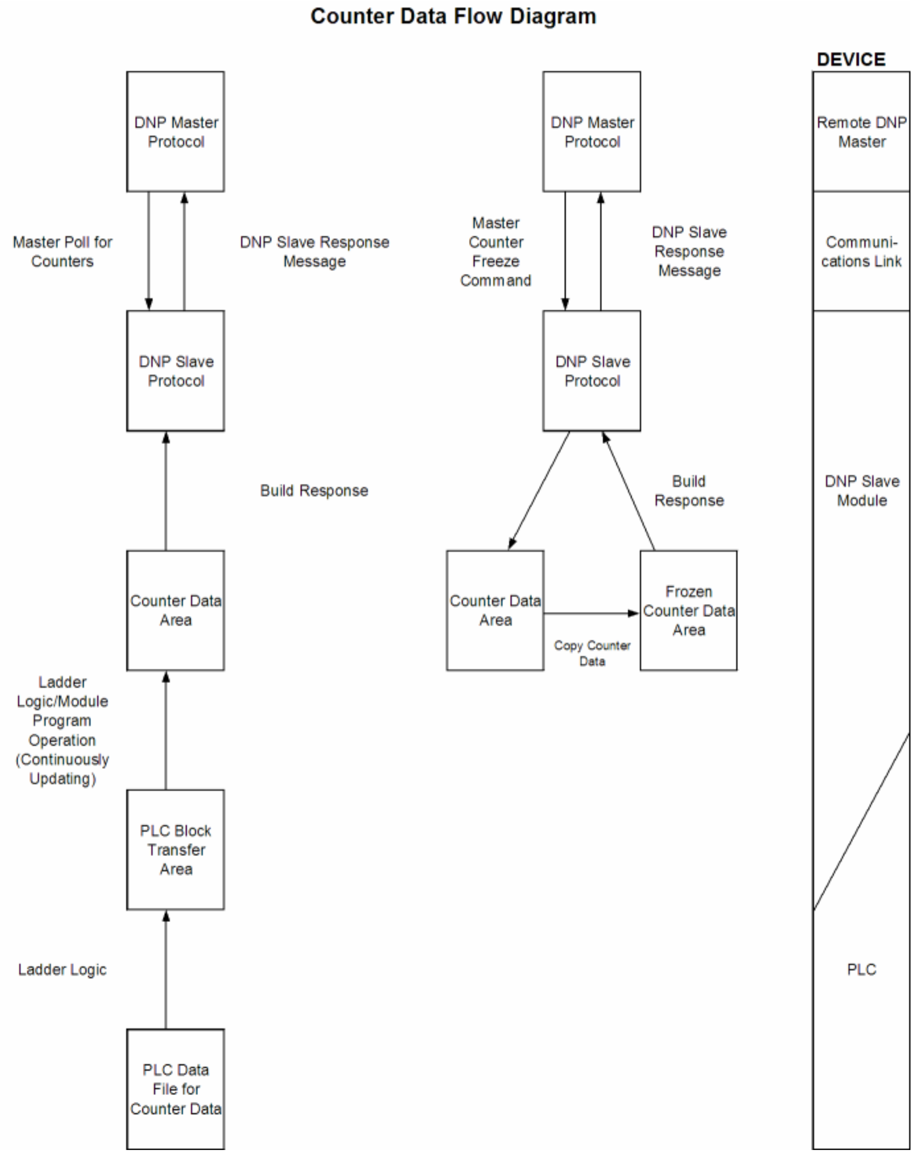

Binary Output Data Flow Diagram
DEVICE
DNP Master Protocol
Master Poll for Binary Outputs
DNP Slave Response Message
DNP Slave Protocol
Build Response
Binary Output Data Area
DNP Master Protocol
Master Binary Ouput Command
DNP Slave Response Message
DNP Slave Protocol
Change value(s) in Database
Build Response
Binary Output Data Area
Ladder Logic/Module Program Operation (Updated After Command)
PLC Block Transfer Area
Ladder Logic
PLC Data File for Binary Output Data
Remote DNP Master
Communi-cations Link
DNP Slave Module
PLC

DNP Counter Data

This data type stores accumulated count data. These data are stored in the module in a double word value and have a data range of 0 to 4,294,967,296. The size of this data area is determined from the configuration parameter Counters. The PLC transfers data of this type to the module using the read operation. The module maintains two values for each counter point: a current running value and a frozen value. The DNP master must send the freeze command to the module in order to transfer the current running values to the frozen area.

Note: The freeze-reset command is not supported in the data transfer operation. There is no way to guarantee counts will not be lost using the freeze-reset operation, therefore, this feature is not implemented.

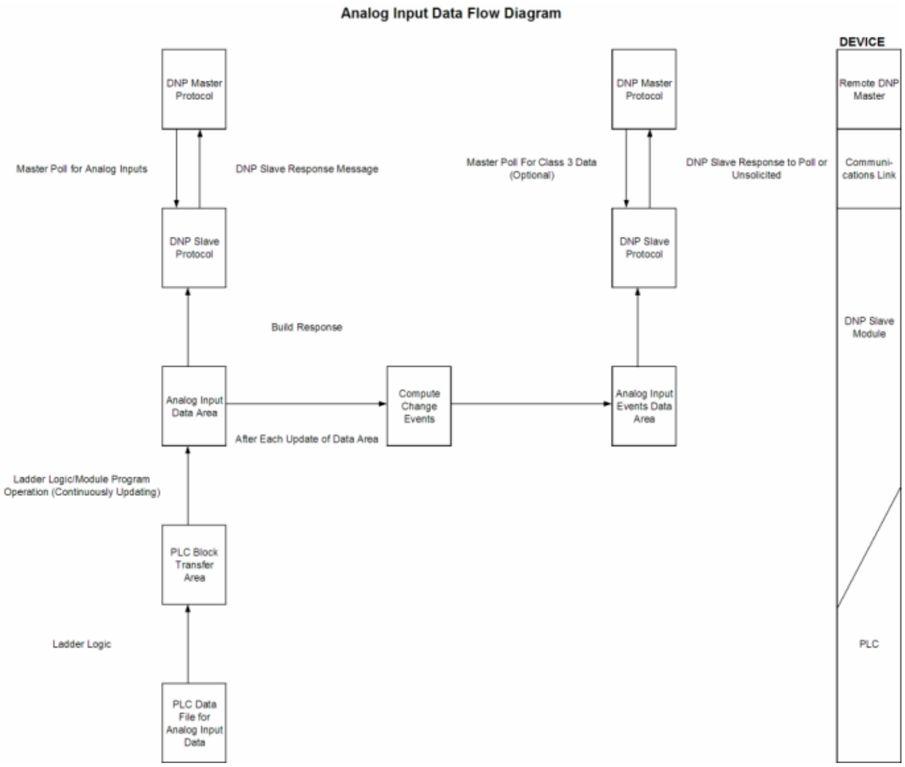
A data flow diagram for the counter data is shown in the following figure.



DNP Analog Input Data

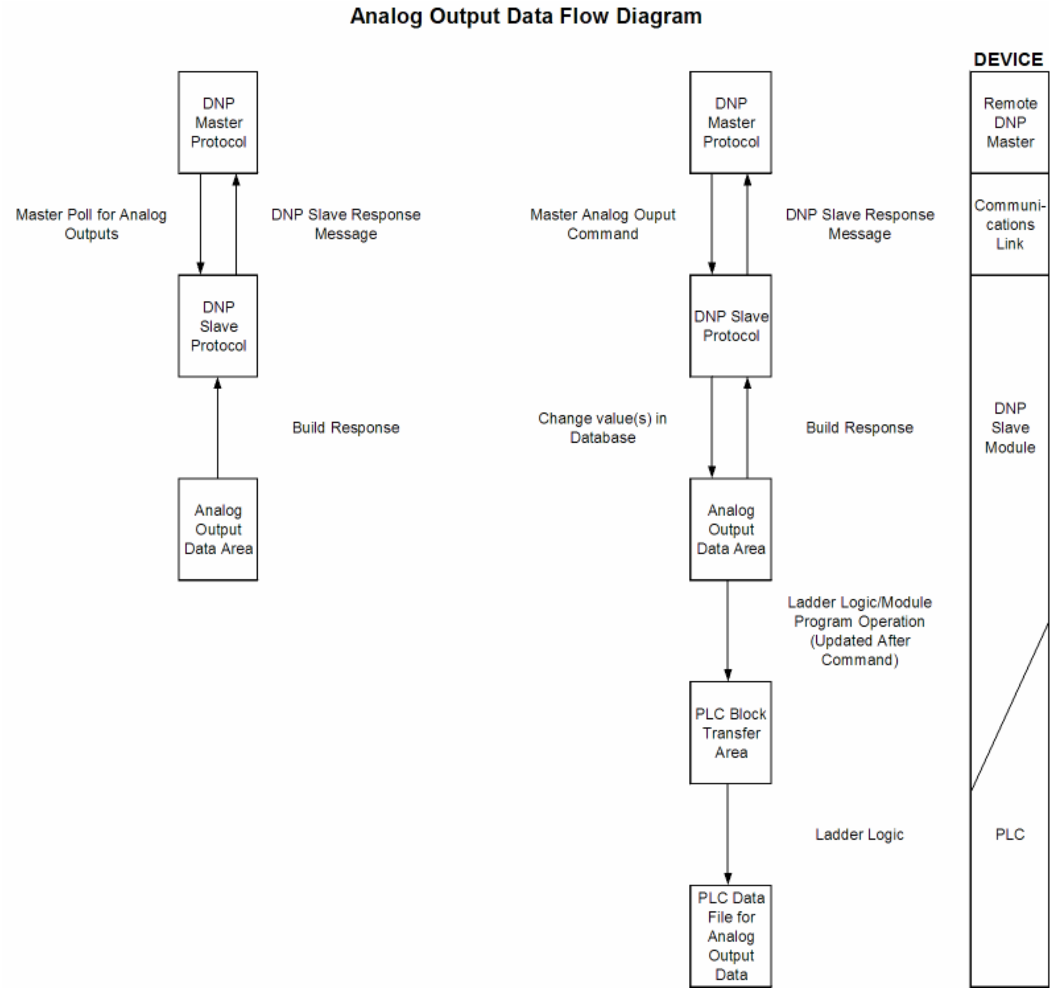
This data type stores analog data with a data range of 0 to 65535 or -32768 to 32767. The size of this data area is determined from the configuration parameter Analog Inputs. These data are transferred to the module from the PLC using the read operation. Therefore, these data are read-only for the module and the DNP master unit. When the module receives a new block of this data from the PLC, it compares the new values to those currently in the database. If there is a change in any of the data, the module will generate an event message for the points that change. The dead-band parameter configured for the module determines the variance required for the event message.

The DNP master unit can read the current value data and the event data from the module. Event messages generated by the module can be retrieved using a poll for Class 3 data, as all analog input events are considered a Class 3 data type. If unsolicited message generation is enabled in the application, the events will automatically be sent by the module to the DNP master unit when the maximum event count for Class 3 data is reached or when the timeout for unsolicited messages is exceeded. A data flow diagram for the analog input data is shown in the following figure.


Analog Input Data Flow Diagram
DEVICE
DNP Master Protocol
DNP Master Protocol
Remote DNP Master
Master Poll for Analog Inputs
DNP Slave Response Message
Master Poll For Class 3 Data (Optional)
DNP Slave Response to Poll or Unsolicited
Communi-cations Link
DNP Slave Protocol
DNP Slave Protocol
DNP Slave Module
Build Response
Analog Input Data Area
Compute Change Events
Analog Input Events Data Area
After Each Update of Data Area
Ladder Logic/Module Program Operation (Continuously Updating)
PLC Block Transfer Area
Ladder Logic
PLC
PLC Data File for Analog Input Data

DNP Analog Output Data

This data type stores analog values sent from the DNP master unit to the module and PLC with a data range of 0 to 65535 or -32768 to 32767. The size of this data area is determined from the configuration parameter Analog Outputs. These data are transferred from the module to the PLC using the write operation. Therefore, these data are read-only for the PLC, as the PLC cannot directly alter these values in the module. It is the responsibility of the DNP master unit to maintain this data. For example, if the DNP master sends a value of 3405 to the module for a specific point, the value will be stored in the module until changed by the master. A data flow diagram for the analog output data is shown in the following figure.


Analog Output Data Flow Diagram
DEVICE
DNP Master Protocol
Master Poll for Analog Outputs
DNP Slave Response Message
DNP Slave Protocol
Build Response
Analog Output Data Area
DNP Master Protocol
Master Analog Ouput Command
DNP Slave Response Message
DNP Slave Protocol
Change value(s) in Database
Build Response
Analog Output Data Area
Ladder Logic/Module Program Operation (Updated After Command)
PLC Block Transfer Area
Ladder Logic
PLC Data File for Analog Output Data
Remote DNP Master
Communi-cations Link
DNP Slave Module
PLC

Functionality

This phase of design defines the features of the DNP Level 2 Subset supported by the module and to be utilized in the specific application. For example, will the unit use unsolicited messaging? Coordination with the DNP master developer is required to verify that the host will support the functionality you select. The features that must be defined in this design step are as follows:

- Will analog events be returned with or without a time value?
- Will events be logged before time synchronization has occurred?
- Will the module start with database values initialized by the processor?

For a complete description of the module configuration, refer to the **Module Setup** section.

Data Transfer at Startup

The module can be configured to have the internal databases initialized with data contained in the processor. This feature requires ladder logic. Data to be initialized are as follows: Binary and Analog Output data. This feature can be used to bring the module to a known state (last state set in controller) when the module is first initialized. For example, in order to have the module startup using the last set of binary output values and setpoint values (analog outputs), enable this feature.

Module Operation

After the system has been designed and the system is set up, the module will be ready to operate. When the module is first initialized, it will read the configuration file After the file is processed, the module will use the data to set up the data structures of the application. If any errors are encountered during the initialization process, the default value for the parameter will be assigned and used.

The module will next check if the output initialization feature is utilized. The option permits the PLC to set these read-only data at startup. There is no static memory available on the module to remember the last values for these data types. In order to prevent a "shock" to the system at boot time, this option can be used to set the module's database to the last transferred set of data.

If the module is configured for unsolicited messaging, the module will immediately send an unsolicited response once the remote master connects to the module, informing the master of a module restart. The module will not log events or process any data read operations from the master until the master clears the restart IIN data bit. The master must also synchronize the time with the module before events will be generated if the module is so configured. The master is also responsible for enabling the unsolicited message facility in the module by sending the Enable Unsolicited Messaging command to the module.

If the module is not configured for unsolicited messaging, the DNP master must clear the restart IIN bit before the module will start logging events. The master must also synchronize the time with the module before events will be generated if the module is so configured.

Additionally, the program will listen on Port 1 for requests. This is the debug port for the module and transfers module information to an attached terminal. Refer to **Diagnostics and Troubleshooting** (page 77) for a complete discussion on the use of this important feature.

9.3 Module Error Codes

If the module's program encounters an error during execution, it will log the error to the error list. This list is transferred to the Quantum / Unity processor using block identification code 100 (see section above) in at offsets 62 to 119. This data is also available for viewing on the debug monitor port. The following tables list the error codes generated by the program with their associated description. Use the errors to help define where problems exist in the system.

9.3.1 Configuration Error Word

Bit	Code	Description
0	0x0001	Invalid baud rate selected
1	0x0002	Invalid address assigned (0 to 65534)
2	0x0004	Database defined will not fit into memory
3	0x0008	Invalid binary input point count
4	0x0010	Invalid binary output point count
5	0x0020	Invalid counter point count
6	0x0040	Invalid analog or float input point count
7	0x0080	Invalid analog or float output point count
8	0x0100	
9	0x0200	
10	0x0400	
11	0x0800	
12	0x1000	
13	0x2000	
14	0x4000	
15	0x8000	

9.3.2 Slave Port Communication Errors

Error Code	Name	Description
0	OK	The module is operating correctly and there are no errors.
10	DNP synchronization error (Physical Layer Error)	Extra bytes are received before the start bytes (0x05 and 0x64).
11	DNP overrun error (Physical Layer Error)	Mainline Data Link Layer routine could not read data received on DNP port before it was overwritten.
12	DNP length error (Physical Layer Error)	Length of message does not match length value in message.
13	DNP bad CRC error (Data Link Layer Error)	Computed CRC value for message does not match that received in message.
14	DNP user data overflow error (Transport Layer Error)	Application layer received a message fragment buffer which is too small.

Error Code	Name	Description
15	DNP sequence error (Transport Layer Error)	Sequence numbers of multi-frame request fragments do not increment correctly.
16	DNP address error (Transport Layer Error)	Source addresses contained in multi-frame request fragments do not match.
17	DNP bad function code error (Application Layer Error)	Function code received from DNP master is not supported for selected object/variation.
18	DNP object unknown error (Application Layer Error)	Slave does not have the specified objects or there are no objects assigned to the requested class.
19	DNP out of range error (Application Layer Error)	Qualifier, range or data fields are not valid or out of range for the selected object/variation.
20	DNP message overflow error (Application Layer Error)	Application response buffer overflow condition. The response message from the slave is too long to transmit.
21	DNP master multi-frame message error (Application Layer Error)	Received a multi-frame message from the DNP master. This application does not support multi-frame messages from the master.

9.3.3 System Configuration Errors

Error Code	Name	Description
100	Too many binary input points	Too many binary input points are configured for the module. Maximum value is 512.
101	Too many binary output points	Too many binary output points are configured for the module. Maximum value is 512.
102	Too many counter points	Too many counter points are configured for the module. Maximum value is 128.
103	Too many analog input points	Too many analog input points are configured for the module. Maximum value is 512.
104	Too many analog output points	Too many analog output points are configured for the module. Maximum value is 512.
107	Invalid analog input deadband	Deadband value for analog input events is out of range. Value must be in the range of 0 to 32767.
108	Not enough memory	There is not enough memory in the module to configure the module as specified.
123	Too many float input or output points	Too many float input or output points are configured for the module. Maximum value is 128 for each type.
334	Baud rate of secondary port	The baud rate configured for the secondary slave port is invalid.

9.3.4 DNP Port Configuration Errors

Error Code	Name	Description
212	Invalid DNP address	The DNP address specified in the configuration is not valid (0 to 65534).
213	Invalid DNP port baud rate	The baud rate code specified in the configuration is not valid.
219	Invalid DNP data link layer confirm mode	The data link confirmation mode code is not valid in the configuration.
220	Invalid DNP data link confirm time-out	The data link time-out period specified in the configuration is 0. It must be an integer in the range of 1 to 65535.
222	Invalid DNP select/operate arm time duration	The select/operate arm timer is set to 0. It must be an integer in the range of 1 to 65535.
223	Invalid DNP application layer confirm time-out	The application layer confirm time-out value is set to 0. It must be an integer in the range of 1 to 65535.
224	Invalid DNP write time interval	The write time interval is not in the data range in the configuration. The value must be in the range of 0 to 1440.
225	Invalid DNP unsolicited response mode	The unsolicited response mode code is not valid in the configuration.
226	Invalid DNP unsolicited response minimum quantity for Class 1	The unsolicited response minimum quantity for Class 1 is not valid in the configuration. Value must be an integer in the range of 1 to 255.
227	Invalid DNP unsolicited response minimum quantity for Class 2	The unsolicited response minimum quantity for Class 2 is not valid in the configuration. Value must be an integer in the range of 1 to 255.
228	Invalid DNP unsolicited response minimum quantity for Class 3	The unsolicited response minimum quantity for Class 3 is not valid in the configuration. Value must be an integer in the range of 1 to 255.
230	Invalid DNP unsolicited response destination address	The unsolicited response destination address is not valid in the configuration. Value must be in the range of 1 to 65534.

9.3.5 Error Status Table

The program maintains an error/status table. This table of data is available to the Quantum / Unity processor automatically through block 100. Ladder logic should be programmed to accept this block of data and place it in the module's controller tag. You can use the error/status data to determine the "health" of the module.

The data in the block is structured as shown in the following table:

Word	Block Offset	Variable Name	Description
0	2	Current DNPS Slave Port status	This value represents the current value of the error code for the port. This value will only be valid if the port is configured as a slave. The possible values are described in the application documentation.
1	3	DNPS Slave Port last transmitted error code	This value represents the last error code transmitted to the master by this slave port.
2	4	DNPS Slave Port total number of message frames received by slave	This value represents the total number of message frames that have matched this slaves address on this port. This count includes message frames which the slave may or may not be able to parse and respond.
3	5	DNPS Slave Port total number of response message frames sent from slave	This value represents the number of good (non-error) responses that the slave has sent to the master on this port. The presumption is that if the slave is responding, the message was good. Note: This is a frame count.
4	6	DNPS Slave Port total number of message frames seen by slave	This value represents the total number of message frames received by the slave, regardless of the slave address.
5	7	DNPS Slave synchronization error count (Physical Layer Error)	This value counts the number of times a sync error occurs. The error occurs when extra bytes are received before the start bytes (0x05 and 0x64) are received.
6	8	DNPS Slave overrun error count (Physical Layer Error)	This value counts the number of times the overrun error occurs. This error occurs when the mainline Data Link Layer routine cannot read the data received on the communication port before it is overwritten.
7	9	DNPS Slave length error count (Physical Layer Error)	This value counts the number of times an invalid length byte is received. If the length of the message does not match the length value in the message, this error occurs.
8	10	DNPS Slave bad CRC error (Data Link Layer Error)	This value counts the number of times a bad CRC value is received in a message.
9	11	DNPS Slave user data overflow error (Transport Layer Error)	This value counts the number of times the application layer receives a message fragment buffer which is too small.
10	12	DNPS Slave sequence error (Transport Layer Error)	This value counts the number of times the sequence numbers of multi-frame request fragments do not increment correctly.
11	13	DNPS Slave address error (Transport Layer Error)	This value counts the number of times the source addresses contained in a multi-frame request fragments do not match.
12	14	DNPS Slave Binary Input Event count	This value contains the total number of binary input events which have occurred.
13	15	DNPS Slave Binary Input Event count in buffer	This value represents the number of binary input events which are waiting to be sent to the master.
14	16	DNPS Slave Analog Input Event count	This value contains the total number of analog input events which have occurred.
15	17	DNPS Slave Analog Input Event count in buffer	This value represents the number of analog input events which are waiting to be sent to the master.

Word	Block Offset	Variable Name	Description
16	18	DNPS Slave bad function code error (Application Layer Error)	This value counts the number of times a bad function code for a selected object/variation is received by the slave device.
17	19	DNPS Slave object unknown error (Application Layer Error)	This value counts the number of times a request for an unsupported object is received by the slave device.
18	20	DNPS Slave out of range error (Application Layer Error)	This value counts the number of times a parameter in the qualifier, range or data field is not valid or out of range.
19	21	DNPS Slave message overflow error (Application Layer Error)	This value counts the number of times an application response message from the slave is too long to transmit.
20	22	DNPS Slave multi-frame message from DNPS Master error (Application Layer Error)	This value counts the number of times the slave receives a multi-frame message from the master. The application does not support multi-frame master messages.
21	23	Total blocks transferred	Total BTR/BTW or side-connect interface transfers attempted by the module.
22	24	Successful blocks transferred	This value represents the total number of transfer operations between the Quantum / Unity processor and module that are successful.
23	25	Total errors in block transfer	Total number of transfers that resulted in an error condition.
24	26	Total BTR or write errors	Total number of BTR or write transfers that resulted in an error.
25	27	Total BTW or read errors	Total number of BTW or read transfers that resulted in an error.
26	28	Block number error	Number of BTW requests that resulted in an incorrect BTW identification code.
27	29	Continuous block error counter	Count of sequential data transfer errors. When this value exceeds that specified for the data transfer operation, the error flag below will be set.
28	30	Reserved	Not used
29	31	Configuration Type	This is a coded field that defines the configuration of the module. The codes are as follows: 0=Single Slave Configuration, 1=Dual Slave Configuration, 2=Slave/Master Configuration
30 to 31	32 to 33	Product Name (ASCII)	These two words contain the product name of the module in ASCII format.
32 to 33	34 to 35	Revision (ASCII)	These two words contain the product revision level of the firmware in ASCII format.
34 to 35	36 to 37	Operating System Revision (ASCII)	These two words contain the module's internal operating system revision level in ASCII format.
36 to 37	38 to 39	Production Run Number (ASCII)	These two words contain the production 'batch' number for the particular chip in the module in ASCII format.
38	40	DNPS Master Port Slave Count	This is the total number of slaves configured for the DNPS Master port. This may not represent the number of active slaves as it includes slaves that are not enabled.

Word	Block Offset	Variable Name	Description
39	41	DNPS Master Port Command Count	This is the total number of commands configured for the DNPS Master port. This may not represent the number of active commands as it includes commands that are disabled.
40	42	DNPS Master Port Device Memory Block Count	This value represents the number of memory allocation blocks for slave devices. This number should be one greater than the number of slave devices. The extra device is held for the broadcast device.
41	43	DNPS Master Port Frame Block Count	This value represents the number of physical layer frame memory allocation blocks used by the program.
42	44	DNPS Master Port Data Link Receive Block Count	This value represents the number of receive data link layer memory blocks allocated.
43	45	DNPS Master Port Data Link Transmit Block Count	This value represents the number of transmit data link layer memory blocks allocated.
44	46	DNPS Master Port Application Layer Receive Block Count	This value represents the number of application layer receive memory blocks allocated.
45	47	DNPS Master Port Application Layer Receive Block Count	This value represents the number of application layer transmit memory blocks allocated.
46	48	DNPS Master Port Device Memory Allocation Error Count	This value represents the number of memory allocation errors for device blocks.
47	49	DNPS Master Port Physical Layer Memory Allocation Error Count	This value represents the number of memory allocation errors for physical layer frame blocks.
48	50	DNPS Master Port Data Link Layer Receive Memory Allocation Error Count	This value represents the number of memory allocation errors for data link layer receive blocks.
49	51	DNPS Master Port Data Link Layer Transmit Memory Allocation Error Count	This value represents the number of memory allocation errors for data link layer transmit blocks.
50	52	DNPS Master Port Application Layer Receive Memory Allocation Error Count	This value represents the number of memory allocation errors for application layer receive blocks.
51	53	DNPS Master Port Application Layer Transmit Memory Allocation Error Count	This value represents the number of memory allocation errors for application layer transmit blocks.
52	54	DNPS Master Synchronization Error Count (Physical Layer Error)	This value counts the number of times a sync error occurs. The error occurs when extra bytes are received before the start bytes (0x05 and 0x64) are received.
53	55	DNPS Master Length Error Count (Physical Layer Error)	This value counts the number of times an invalid length byte is received. If the length of the message does not match the length value in the message, this error occurs.

Word	Block Offset	Variable Name	Description
54	56	DNPS Master Bad CRC Error Count (Physical Layer Error)	This value counts the number of times a bad CRC value is received in a message.
55	57	Scan Counter LSB	Program scan counter
56	58	Scan Counter MSB	
57	59	Free Memory LSB	Free memory in module
58	60	Free Memory MSB	
59	61	DNPS Slave Port Transmit State	Value of the DNPS Slave state machine for transmit.
60	62	DNPS Float Event Count	Total number of events generated for analog floating-point input data points.
61	63	DNPS Double Event Count	Total number of events generated for analog double, floating-point input data points.
62	64	Event Message Queue Count	Number of event messages waiting to send to processor.
63	65	Event Message Queue Overflow	Flag to indicate if the event message queue has overflowed. If more than 200 event messages are received on the master port and they are not sent to the processor, this flag will be set (1). The flag will clear after the messages are sent to the processor.
64 to 77	66 to 79	Reserved	Future Use
78	80	Error_List[0]	First value in error list
79	81	Error_List[1]	Second value in error list
-	-	-	-
137	139	Error_List[59]	Last value in error list

9.3.6 Command Error Codes

General Command Errors

Error Code	Name	Description
-1 (65535)	Current command being issued on the port	Command has been issued out the port, and the module is waiting for the slave to respond.
0	OK	The command was issued and responded to correctly.
1	Device not defined	The IED slave address referenced in the command is not defined in the module. Check to make sure there is an entry in the slave table for each slave device referenced in the command list.
2	Invalid command	This command is not valid. Check to make sure the slave address parameter is greater than or equal to zero and that the point count is not set to zero.
3	Object not supported	The data object in the command is not supported by the module. Refer to the DNPS subset for the Master Port.

Error Code	Name	Description
4	Command function not supported	The function specified in the command is not supported for the object type selected. Refer to the DNPS subset for the Master Port.
10	Invalid binary input poll command	This binary input object command is not valid.
11	Invalid binary input event poll command	This binary input event object poll command is not valid.
20	Invalid binary output command function	This binary output function command is not valid.
30	Invalid counter poll command function	The counter object poll command contains an invalid function code.
31	Invalid counter poll command	This counter object poll command is not valid.
40	Invalid frozen counter poll command	This frozen counter object poll command is not valid.
50	Invalid analog input poll command	This analog input poll command is not valid.
51	Invalid analog input event poll command	This analog input event poll command is not valid.
60	Invalid analog output poll function command	This analog output poll command contains an invalid function code.
61	Invalid analog output poll command	This analog output poll command is not valid.
70	Invalid time/date poll command	This time/date object poll command is not valid.
80	Invalid event poll command	This event poll command is not valid.

Application Layer Errors

Error Code	Name	Description
1000	Device index invalid	The device index in the request or response message is not found in the slave list.
1001	Duplicate request in application layer queue	The newly submitted message to the application layer already exists in the queue. The message is ignored.
1002	COM port device removed from system	The communication port for the message has been uninstalled on the system. This error should never occur as the communication ports are only uninstalled when the module's program is terminated.
1003	Sequence number error	The application sequence number in the response message does not match that based on the last request message. This indicates application layer messages are received out of order.
1004	Response to select before operate does not match	The select response message received from the slave module is not that expected from the last select request. This indicates a synchronization problem between the master and slave devices.
1005	Response does not contain date/time object	The response message from the slave device does not contain a date/time object. The master expects this object for the response message.

Error Code	Name	Description
1006	Time-out condition on response	The slave device did not respond to the last request message from the master within the time-out set for the IED device. The application layer time-out value is specified for each IED unit in the slave configuration table in the module. This table is established each time the module performs the restart operation.
1007	Function code in application layer message not supported	The function code returned in the response message is not valid for the application layer or not supported by the module.
1008	Read operation not supported for object/variation	The application layer response message contains an object that does not support the read function.
1009	Operate function not supported for the object/variation	The application layer response message contains an object that does not support the operate function.
1010	Write operation not supported for the object/variation	The application layer response message contains an object that does not support the write function.

9.4 Master Port DNPS Slave Configuration Values (DNPS Master Slave List)

Column	Variable Name	Data Range	Description	IF Error	Config. Value
1	DNPS Slave Address	0 to 65534	This is the slave address for the unit to override the default values.	Ignore	
2	Data Link Confirm Mode	Coded Value (0=Never, 1=Sometimes, 2=Always)	This value specifies if data link frames sent to the remote device require a data link confirm. This value should always be set to zero for almost all applications.	0	
3	Data Link Confirm Time-out	1 to 65535 milliseconds	This parameter specifies the time to wait for a data link confirm from the remote device before a retry is attempted.	300	
4	Maximum Retries for Data Link Confirm	0 to 255 retries	Maximum number of retries at the Data Link level to obtain a confirmation. If this value is set to 0, retries are disabled at the data link level of the protocol. This parameter is only used if the frame is sent with confirmation requested.	3	

Column	Variable Name	Data Range	Description	IF Error	Config. Value
5	Application Layer Response Time-out	1 to 65535 milliseconds	Time-out period the master will wait for each response message fragment. If data link confirms are enabled, make sure the time-out period is set long enough to permit all data confirm retries.	5000	
6	Slave Mode	Coded Value (Bit 0=Enable, Bit 1=Unsol Msg, Bit 2=Use DM, Bit 3=Auto Time Sync)	This word contains bits that define the slave mode. The slave mode defines the functionality of the slave device and can be combined in any combination. The fields have the following definition: Enable-- determines if this slave will be used. Unsol Msg-- causes an enabled unsolicited response message to be sent to the slave when its RESTART IIN bit is set. This parameter is also required for unsolicited message reporting by the IED unit.Use DM--uses delay measurement. Auto Time Sync--time synchronization used when NEED TIME IIN bit set.	5	

9.5 Command List Entry Form

	0	1	2	3	4	5	6	7	8	9
#	Port/ Flags	Slave Add.	Object	Variation	Function	Address	Pnt Count	DNP DB	IED DB	Poll Interval
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										

	0	1	2	3	4	5	6	7	8	9
#	Port/ Flags	Slave Add.	Object	Variation	Function	Address	Pnt Count	DNP DB	IED DB	Poll Interval
14										
15										
16										
17										
18										
19										
20										
21										
22										
23										
24										
25										
26										
27										
28										
29										
30										
31										
32										
33										
34										
35										
36										
37										
38										
39										
40										
41										
42										
43										
44										
45										
46										
47										
48										
49										

9.6 Command List

	Digital input	Digital input Events	Digital Output	Counter
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address
2	1	2	12	20
3	0, 1 or 2	0, 1, 2 or 3	1*	0, 5 or 6
4	1*	1*	3, [4], 5 or 6	1, 7, 8, 9 or 10
5	Address in Slave	Address in Slave	Address in Slave	Address in Slave
6	# of Points	# of Points	# of Points	# of Points
7	DNPS DB Address		DNPS DB Address	DNPS DB Address
8	IED DB Address		IED DB Address	IED DB Address
9	Poll Interval	Poll Interval	Poll Interval	Poll Interval

	Frozen Counter	Analog Input	Analog Input Events	Analog Output
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address
2	21	30	32	41
3	0, 9 or 10	0, 1, 2, 3 or 4	0, 1, 2, 3 or 4	2*
4	1*	1*	1*	3, [4], 5 or 6
5	Address in Slave	Address in Slave	Address in Slave	Address in Slave
6	# of Points	# of Points	# of Points	# of Points
7	DNPS DB Address	DNPS DB Address		DNPS DB Address
8	IED DB Address	IED DB Address		IED DB Address
9	Poll Interval	Poll Interval	Poll Interval	Poll Interval

	Time and Date	Class 0	Class 1	Class 2
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address
2	50	60	60	60
3	1*	1	2	3
4	2*			
5				
6	1	1	1	1
7				
8				
9	Poll Interval	Poll Interval	Poll Interval	Poll Interval

	Class 3	Cls 1, 2 & 3	Cls 0, 1, 2 & 3	[Clear Restart Bit]
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address

	Class 3	Cls 1, 2 & 3	Cls 0, 1, 2 & 3	[Clear Restart Bit]
2	60	60	60	80
3	4	5	6	1
4				2
5				7
6	1	1	1	1
7				
8				
9	Poll Interval	Poll Interval	Poll Interval	

	Cold Restart	Warm Restart	Enable Unsol. Msg	Disable Unsol. Msg
0	Port/Flags	Port/Flags	Port/Flags	Port/Flags
1	Slave Address	Slave Address	Slave Address	Slave Address
2	0	0	0	0
3				
4	13	14	20	21
5				
6	1	1	1	1
7				
8				
9				

9.7 DNP Subset Definition

Note: Objects that we support that are not required within the Level II specification are grayed out. Refer to the associated notes to determine our response to the message.

OBJECT			REQUEST		RESPONSE			
Obj	Va r	Description	Func Code s	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
1	0	Binary Input - All Variations	1	06				Slave will return variation 1 data (user can override and have variation 2 returned)
	1	Binary Input	1	06	129, 130	00, 01	1	Slave will return this variation
	2	Binary Input with Status	1	06	129, 130	00, 01	8	Slave will return this variation
2	0	Binary Input Change - All Variations	1	06, 07, 08				Slave will return variation 2 data (user can override and have variation 1 returned)
	1	Binary Input Change Without Time	1	06, 07, 08	129, 130	17, 28	8	Slave will return this variation
	2	Binary Input Change With Time	1	06, 07, 08	129, 130	17, 28	56	Slave will return this variation

OBJECT			REQUEST		RESPONSE			
Obj	Va r	Description	Func Code s	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	3	Binary Input Change With Relative Time	1	06, 07, 08	129, 130	17, 28	24	Slave will parse this message and return no data
10	0	Binary Output - All Variations	1	00, 06				Slave will return variation 2 data (user can override and have variation 1 returned)
	1	Binary Output	1	00, 06			1	Slave will return this variation
	2	Binary Output Status	1	00, 06	129, 130	00, 01	8	Slave will return this variation
12	0	Control Block - All Variations					88	Slave will use variation 1 control
	1	Control Relay Output Block	3, 4, 5, 6	17, 28	129	Echo of request	88	Slave will respond correctly to this variation
	2	Pattern Control Block					88	Slave will return Unknown Object to this request
	3	Pattern Mask					16	Slave will return Unknown Object to this request
20	0	Binary Counter - All Variations	1, 7, 8, 9, 10	06				Slave will return variation 5 data (user can override and have variation 1 returned)
	1	32-Bit Binary Counter	1, 7, 8, 9, 10	06	129, 130	00, 01	40	Slave will return this variation
	2	16-Bit Binary Counter	1, 7, 8, 9, 10	06	129, 130	00, 01	24	Slave will return this variation (counter upper 16-bits removed)
	3	32-Bit Delta Counter			129, 130	00, 01	40	Slave will return Unknown Object to this request
	4	16-Bit Delta Counter			129, 130	00, 01	24	Slave will return Unknown Object to this request
	5	32-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	32	Slave will return this variation
	6	16-Bit Binary Counter Without Flag	1, 7, 8, 9, 10	06	129, 130	00, 01	16	Slave will return this variation (counter upper 16-bits removed)
	7	32-Bit Delta Counter Without Flag			129, 130	00, 01	32	Slave will return Unknown Object to this request
	8	16-Bit Delta Counter Without Flag			129, 130	00, 01	16	Slave will return Unknown Object to this request
21	0	Frozen Counter - All Variations	1	06				Slave will return variation 9 data (user can override and have variation 1 returned)
	1	32-Bit Frozen Counter	1	06	129, 130	00, 01	40	Slave will return this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	2	16-Bit Frozen Counter	1	06	129, 130	00, 01	24	Slave will return this variation (counter upper 16-bits removed)
	3	32-Bit Frozen Delta Counter					40	Slave will return Unknown Object to this request
	4	16-Bit Frozen Delta Counter					24	Slave will return Unknown Object to this request
	5	32-Bit Frozen Counter With Time Of Freeze					88	Slave will return Unknown Object to this request
	6	16-Bit Frozen Counter With Time Of Freeze					72	Slave will return Unknown Object to this request
	7	32-Bit Frozen Delta Counter With Time Of Freeze					88	Slave will return Unknown Object to this request
	8	16-Bit Frozen Delta Counter With Time Of Freeze					72	Slave will return Unknown Object to this request
	9	32-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	32	Slave will return this variation
	10	16-Bit Frozen Counter Without Flag	1	06	129, 130	00, 01	16	Slave will return this variation (counter upper 16-bits removed)
	11	32-Bit Frozen Delta Counter Without Flag					32	Slave will return Unknown Object to this request
	12	16-Bit Frozen Delta Counter Without Flag					16	Slave will return Unknown Object to this request
22	0	Counter Change Event - All Variations	1	06, 07, 08				Slave will parse this request and return no data
	1	32-Bit Counter Change Event Without Time			129, 130	17, 28	40	Slave will return Unknown Object to this request
	2	16-Bit Counter Change Event Without Time			129, 130	17, 28	24	Slave will return Unknown Object to this request
	3	32-Bit Delta Counter Change Event Without Time					40	Slave will return Unknown Object to this request

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	4	16-Bit Delta Counter Change Event Without Time					24	Slave will return Unknown Object to this request
	5	32-Bit Counter Change Event With Time					88	Slave will return Unknown Object to this request
	6	16-Bit Counter Change Event With Time					72	Slave will return Unknown Object to this request
	7	32-Bit Delta Counter Change Event With Time					88	Slave will return Unknown Object to this request
	8	16-Bit Delta Counter Change Event With Time					72	Slave will return Unknown Object to this request
23	0	Frozen Counter Event - All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Counter Event Without Time					40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Counter Event Without Time					24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Delta Counter Event Without Time					40	Slave will return Unknown Object to this request
	4	16-Bit Frozen Delta Counter Event Without Time					24	Slave will return Unknown Object to this request
	5	32-Bit Frozen Counter Event With Time					88	Slave will return Unknown Object to this request
	6	16-Bit Frozen Counter Event With Time					72	Slave will return Unknown Object to this request
	7	32-Bit Frozen Delta Counter Event With Time					88	Slave will return Unknown Object to this request
	8	16-Bit Frozen Delta Counter Event With Time					72	Slave will return Unknown Object to this request
30	0	Analog Input - All Variations	1	06				Slave will respond with variation 4 data (user can override and have variation 2 returned)

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1	32-Bit Analog Input	1	06	129, 130	00, 01	40	Slave will return this variation (Note: Data will only be 16-bit)
	2	16-Bit Analog Input	1	06	129, 130	00, 01	24	Slave will return this variation
	3	32-Bit Analog Input Without Flag	1	06	129, 130	00, 01	32	Slave will return this variation (Note: Data will only be 16-bit)
	4	16-Bit Analog Input Without Flag	1	06	129, 130	00, 01	16	Slave will return this variation
	5	Short Floating Point Analog Input	1	06	129, 130	00, 01	40	Slave will return this variation
	6	Long Floating Point Analog Input	1	06	129, 130	00, 01	72	Slave will return this variation
31	0	Frozen Analog Input - All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Analog Input					40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Analog Input					24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Analog Input With Time To Freeze					88	Slave will return Unknown Object to this request
	4	16-Bit Frozen Analog Input With Time To Freeze					72	Slave will return Unknown Object to this request
	5	32-Bit Frozen Analog Input Without Flag					32	Slave will return Unknown Object to this request
	6	16-Bit Frozen Analog Input Without Flag					16	Slave will return Unknown Object to this request
	7	Short Floating Point Frozen Analog Input					40	Slave will return Unknown Object to this request
	8	Long Floating Point Frozen Analog Input					72	Slave will return Unknown Object to this request
32	0	Analog Change Event - All Variations	1	06, 07, 08				Slave will return variation 2 data (user can override and have variation 4 returned)
	1	32-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	40	Slave will return this variation (Note: Data only 16-bit)
	2	16-Bit Analog Change Event Without Time	1	06, 07, 08	129, 130	17, 28	24	Slave will return this variation

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	3	32-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Slave will return this variation (Note: Data only 16-bit)
	4	16-Bit Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	72	Slave will return this variation
	5	Short Floating Point Analog Change Event	1	06, 07, 08	129, 130	17, 28	40	Slave will return this variation
	6	Long Floating Point Analog Change Event	1	06, 07, 08	129, 130	17, 28	72	Slave will return this variation
	7	Short Floating Point Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	88	Slave will return this variation
	8	Long Floating Point Analog Change Event With Time	1	06, 07, 08	129, 130	17, 28	120	Slave will return this variation
33	0	Frozen Analog Event - All Variations						Slave will return Unknown Object to this request
	1	32-Bit Frozen Analog Event Without Time					40	Slave will return Unknown Object to this request
	2	16-Bit Frozen Analog Event Without Time					24	Slave will return Unknown Object to this request
	3	32-Bit Frozen Analog Event With Time					88	Slave will return Unknown Object to this request
	4	16-Bit Frozen Analog Event With Time					72	Slave will return Unknown Object to this request
	5	Short Floating Point Frozen Analog Event					40	Slave will return Unknown Object to this request
	6	Long Floating Point Frozen Analog Event					72	Slave will return Unknown Object to this request
	7	Short Floating Point Frozen Analog Event With Time					88	Slave will return Unknown Object to this request
	8	Long Floating Point Frozen Analog Event With Time					120	Slave will return Unknown Object to this request

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
40	0	Analog Output Status - All Variations	1	06			24	Slave will return variation 2 data
	1	32-Bit Analog Output Status	1	06	129,130	00,01	40	Slave will return this variation but data only 16-bit accuracy
	2	16-Bit Analog Output Status	1	06	129, 130	00, 01	24	Slave will return this variation
	3	Short Floating Point Analog Output Status	1	06	129, 130	00, 01	40	Slave will return this variation
	4	Long Floating Point Analog Output Status	1	06	129, 130	00, 01	72	Slave will return this variation
41	0	Analog Output Block - All Variations					24	Slave will respond to this request using variation 2 data
	1	32-Bit Analog Output Block	3, 4, 5, 6	17, 28	129,130	00,01	40	Slave will respond to this request but data only 16-bit
	2	16-Bit Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	24	Slave will respond to this request
	3	Short Floating Point Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	40	Slave will respond to this request
	4	Long Floating Point Analog Output Block	3, 4, 5, 6	17, 28	129	Echo of Request	72	Slave will respond to this request
50	0	Time and Date - All Variations	2	07, With Quant=1			48	Slave will use variation 1
	1	Time and Date	2	07, With Quant=1			48	Slave will respond to this variation
	2	Time and Date With Interval					80	Slave will return Unknown Object to this request
51	0	Time and Date CTO - All Variations						Slave will return Unknown Object to this request
	1	Time and Date CTO			129, 130	07, With Quant=1	48	Slave will return Unknown Object to this request
	2	Unsynchronized Time and Date CTO			129, 130	07, With Quant=1	48	Slave will return Unknown Object to this request
52	0	Time Delay - All Variations						

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
	1	Time Delay Coarse			129	07, With Quant= 1	16	Slave will never return this variation
	2	Time Delay Fine			129	07, With Quant= 1	16	Slave will return this variation to functions 0D, 0E, and 17
60	0	Not Defined						Not Defined in DNP
	1	Class 0 Data	1	06				Slave will respond to this variation with all static data
	2	Class 1 Data	1	06, 07, 08				Slave will respond to this variation (No class 1 data defined in application)
	3	Class 2 Data	1	06, 07, 08				Slave will respond to this variation with all class 2 data (binary input events)
	4	Class 3 Data	1	06, 07, 08				Slave will respond to this variation with all class 3 data (analog input events)
70	0	Not Defined						Not Defined in DNP
	1	File Identifier						Slave will return Unknown Object to this request
80	0	Not Defined						Not Defined in DNP
	1	Internal Indications	2	00, Index=7			24	Slave will respond to this variation
81	0	Not Defined						Not Defined in DNP
	1	Storage Object						
82	0	Not Defined						Not Defined in DNP
	1	Device Profile						
83	0	Not Defined						Not Defined in DNP
	1	Private Registration Object						
	2	Private Registration Objection Descriptor						
90	0	Not Defined						Not Defined in DNP
	1	Application Identifier						
100	0							
	1	Short Floating Point					48	
	2	Long Floating Point					80	
	3	Extended Floating Point					88	

OBJECT			REQUEST		RESPONSE			
Obj	Var	Description	Func Codes	Qual Codes (hex)	Func Codes	Qual Codes (hex)	Data Size (bits)	NOTES
101	0							
	1	Small Packed Binary-Coded Decimal					16	
	2	Medium Packed Binary-Coded Decimal					32	
	3	Large Packed Binary-Coded Decimal					64	
110	0	Not Defined						Not Defined as the variation determines the string length
	1 to 100	Octet String	1	00, 01, 06, 07, 08, 17, 28	129, 130	00, 01, 07, 08, 17, 28	8 * Var #	The module will return this variation for the points defined in the module. The variation determines the returned string length.
No Object			13					Slave supports the Cold Restart Function and will return Obj 52, Var 2, Qual 7, Cnt 1
			14					Slave supports the Warm Restart Function and will return Obj 52, Var 2, Qual 7, Cnt 1
			20					Slave supports the Enable Unsolicited Function
			21					Slave supports the Disable Unsolicited Function
			23					Slave supports the Delay Measurement & Time Synchronization Function and will return Obj 52, Var 2, Qual 7, Cnt 1

9.8 Internal Indication Word

First Byte

Bit	Description
0	All stations message received. Set when a request is received with the destination address set to 0xffff. Cleared after next response. Used to let master station know broadcast received.
1	Class 1 data available. Set when class 1 data is ready to be sent from the slave to the master. Master should request class 1 data when this bit is set.
2	Class 2 data available. Set when class 2 data is ready to be sent from the slave to the master. Master should request class 2 data when this bit is set.

Bit	Description
3	Class 3 data available. Set when class 3 data is ready to be sent from the slave to the master. Master should request class 3 data when this bit is set.
4	Time synchronization required from master. The master should write the date and time when this bit is set. After receiving the write command the bit will be cleared.
5	Slave digital outputs are in local control. This bit is not used in this application.
6	Not Used
7	Device restart. This bit is set when the slave either warm or cold boots. It is cleared after a master writes a 0 to the bit.

Second Byte

Bit	Description
0	Bad function code. The function code contained in the master request is not supported for the specified object/variation.
1	Requested object(s) unknown. Object requested by master is not supported by the application.
2	Parameters in the qualifier, range or data fields are not valid or out of range for the slave.
3	Event buffer(s) or other application buffers have overflowed. This bit is also set if the slave receives a multi-frame message from the master.
4	Request understood but requested operation is already executing. The slave will never set this bit.
5	Bad configuration. The slave configuration is invalid and should be re-configured. If the configuration is invalid, the slave will set the invalid parameters to default values and continue to run. Check error log using debug port.
6	Reserved, always 0.
7	Reserved, always 0.

9.9 Frequently Asked Questions

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.

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 since it is intended to solve very specialized operations its size must be restricted.

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

The only time you should need it is if you are using the DNPS, 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.

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.

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.

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 section on "Backplane Data Exchange" offers to samples to help in the few instances where ladder is required.

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.

Is a .MDC available for configuration of the Module?

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

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 ProLinx family such as the 4202-MNET-DFCM. (if you require DF1 connectivity for instance 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.

Can I use the ProTalk 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 ProLinx Communication products. Please call our technical support technicians when considering this application.

9.9.1 DNPS Specific Questions

What does "Initialize Output Data" in the configuration file mean?

The default of this user parameter is NO. When the module reboots it will reset all of its internal registers to a zero value. In some applications this will cause a problem as the master wishes to see what he/she believes he/she put in that register during the last access. If this is true you should set this parameter to YES, which will cause the module to convert the writes (command function 2) in the [BACKPLANE DATA EXCHANGE] section to reads for one scan and one scan only. This will reload the registers in the module with the information contained within the PLC.

Where do the individual data types actually exist in the modules memory?

The placement of the individual data types is in a pre-defined order, which is the same as they are placed in the configuration file for easy reference. They will be placed in memory sequentially as follows:

- Binary Inputs
- Analog Inputs
- Counter Data
- Binary Outputs
- Analog Outputs

When you describe the database in the DNPS configuration file you should create sufficient data size for your application plus any anticipated growth. If for instance you describe 10 Binary Inputs today and later increase the size to 20, you will have effectively changed the location of your Analog Inputs, Counter Data, Binary Outputs and Analog Outputs by 10 locations.

If you choose not to do this then you should enter one or more commands for each data transfer. In this instance you could change the data AND change the [BACKPLANE DATA EXCHANGE] commands to maintain your mapping in the PLC.

10 Support, Service & Warranty

In This Chapter

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

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

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

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

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

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

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

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

10.1 How to Contact Us: Sales and Support

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

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

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

Asia Pacific Sales (office in Malaysia)

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

Asia Pacific Support (office in China)

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

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

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

Brasil (office in Sao Paulo)

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

10.2 Return Material Authorization (RMA) Policies and Conditions

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

10.2.1 All Product Returns

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

The following policy applies for Non-Warranty Credit Returns:

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

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

10.3 Procedures for Return of Units Under Warranty

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

a) If no defect is found, a credit is issued.

b) If a defect is found and is determined to be customer generated or if the defect is otherwise not covered by ProSoft's Warranty, or if the module is not repairable, a credit is not issued and payment of the replacement module is due.

10.4 Procedures for Return of Units Out of Warranty

1 Customer sends unit in for evaluation.

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

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

3 For an immediate exchange, a new module may be purchased and sent to Customer while repair work is being performed. Credit for purchase of the new module will be issued when the new module is returned in accordance with ProSoft's shipping instructions and subject to ProSoft's policy on non-warranty returns. This is in addition to charges for repair of the old module and any associated charges to Customer.

4 If, upon contacting ProSoft Customer Service, the Customer is informed that unit is believed to be unrepairable, the Customer may choose to send unit in for evaluation to determine if the repair can be made. Customer will pay shipping, duties and taxes that may apply. If unit cannot be repaired, the Customer may purchase a new unit.

10.4.1 Un-repairable Units

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

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

Purchasing Warranty Extension

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

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

10.5 LIMITED WARRANTY

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

10.5.1 What Is Covered By This Warranty

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

B *Warranty On Services*: Material and labor used by ProSoft to repair a verified malfunction or defect are warranted on the terms specified above for new Product, provided said warranty will be for the period remaining on the original new equipment warranty or, if the original warranty is no longer in effect, for a period of 90 days from the date of repair.

C The Warranty Period for RadioLinx accessories (such as antennas, cables, brackets, etc.) are the same as for RadioLinx modules, that is, three years from the date of shipment.

10.5.2 What Is Not Covered By This Warranty

A ProSoft makes no representation or warranty, expressed or implied, that the operation of software purchased from ProSoft will be uninterrupted or error free or that the functions contained in the software will meet or satisfy the purchaser's intended use or requirements; the Customer assumes complete responsibility for decisions made or actions taken based on information obtained using ProSoft software.

B With the exception of RadioLinx accessories referenced in paragraph 1(c) this Warranty does not cover any product, components, or parts not manufactured by ProSoft.

C This Warranty also does not cover the failure of the Product to perform specified functions, or any other non-conformance, defects, losses or damages caused by or attributable to any of the following: (i) shipping; (ii) improper installation or other failure of Customer to adhere to ProSoft's specifications or instructions; (iii) unauthorized repair or maintenance; (iv) attachments, equipment, options, parts, software, or user-created programming (including, but not limited to, programs developed with any IEC 61131-3 programming languages, or "C") not furnished by ProSoft; (v) use of the Product for purposes other than those for which it was designed; (vi) any other abuse, misapplication, neglect or misuse by the Customer; (vii) accident, improper testing or causes external to the Product such as, but not limited to, exposure to extremes of temperature or humidity, power failure or power surges outside of the limits indicated on the product specifications; or (viii) disasters such as fire, flood, earthquake, wind or lightning.

D The information in this Agreement is subject to change without notice. ProSoft shall not be liable for technical or editorial errors or omissions made herein; nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. The user guides included with your original product purchased by you from ProSoft, contains information protected by copyright. No part of the guide may be duplicated or reproduced in any form without prior written consent from ProSoft.

10.5.3 DISCLAIMER REGARDING HIGH RISK ACTIVITIES

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

10.5.4 DISCLAIMER OF ALL OTHER WARRANTIES

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

10.5.5 LIMITATION OF REMEDIES**

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

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

10.5.6 Time Limit for Bringing Suit

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

10.5.7 No Other Warranties

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

10.5.8 Intellectual Property

A Any documentation included with Product purchased from ProSoft is protected by copyright and may not be photocopied or reproduced in any form without prior written consent from ProSoft.

B ProSoft's technical specifications and documentation that are included with the Product are subject to editing and modification without notice.

C Transfer of title shall not operate to convey to Customer any right to make, or have made, any Product supplied by ProSoft.

D Customer is granted no right or license to use any software or other intellectual property in any manner or for any purpose not expressly permitted by any license agreement accompanying such software or other intellectual property.

E Customer agrees that it shall not, and shall not authorize others to, copy software provided by ProSoft (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the Product; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of Product.

10.5.9 Additional Restrictions Relating To Software And Other Intellectual Property

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

10.5.10 Allocation of risks

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

10.5.11 Controlling Law and Severability

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

Index