

4 Configuring a Controller

In this chapter

- Configuring a controller 56
- Configuring ASCII messages 73
- Using configuration extensions 78
- Ethernet configuration 105

Configuring a controller

Before you can put your controller to work, you need to configure it for your network and I/O devices. Most of this is done in the Controller Configuration dialog box.

The Controller Configuration dialog box is divided into five tabs, each with a different set of options. Select a tab by clicking it. The options available under each tab depend on the type of controller you selected for the database.

➤ **To configure a controller:**

1. In the Network Editor, from the **Configuration** menu, click **Configuration** or click  on the toolbar.
2. Click the tab for the options you want to configure:

General: Configures a controller's memory, including the number of coils and registers, I/O cards and drops, the amount of memory set aside for Configuration Extensions, and the number and size of the ASCII messages you want to use.

Ports: Configures the controller's serial and ASCII ports.

Controller: Configures your controller's hardware options. You can also select a new controller here, if needed.

User Loadables: Moves *user loadables* from your Loadable Library into your database so they can be used with a controller, or removes them from the database.

Loadable Library: Converts *user loadables* into NxT format and adds or removes them from your Loadable Library.

3. Enter the new parameters.
4. Click **OK** to save your changes.

Controller configuration - General tab

This tab in the Controller Configuration window allows you to configure a controller's memory. It contains many important options, including the number of coils, registers, segments of logic, and I/O cards the controller will use, the amount of memory it sets aside for configuration extensions, and the number and size of its ASCII messages.



Configuring a controller's memory can be time consuming. It's a good idea to keep the controller's user's manual nearby in case you need to refer to it.

Before you begin, make sure you have the right type of controller selected for your database. The controller's type appears in the Status Bar at the bottom of the Network Editor. The options that appear in the General tab depend on the controller type, so if the wrong one is selected, you can spend time configuring options your controller does not support.

- ❖ To change the type of controller for your database, see "Controller configuration - Controller tab" on page 66.

Select the value you want to change, then type a new one.

The configuration options that appear depend on the controller you select in the Controller tab.

General	Ports	Controller	User Loadables	Loadable Library
0xxx: 00512	Total Messages: 0100	Battery Coil (0x): 00000		
1xxx: 00064	Message Words: 00100	Timer Register(4x): 00000		
3xxx: 00099	ASCII Ports: 00	Time of Day Clock (4x): 00000		
4xxx: 01800		Watch Dog Tm(*10ms): 0000		
Segments: 02		Conf Extension Size: 00064		
I/O Drops(1K): 02		Config Ext Used: 00025		
I/O Cards: 0022		<input checked="" type="checkbox"/> Enable Skips		
DCP Drop ID: 00				

Buttons: OK, Cancel, Help

Set the following parameters for your controller:

Parameter	General tab – Left side of window
Registers	For each of the 0xxxx, 1xxxx, 3xxxx, and 4xxxx register fields, type the number of registers the controller will use. The controller polls all of these registers each scan, so to keep your controller scanning quickly, set up only as many registers as you need.
Segments	Type the number of logic segments the controller will use.
I/O Drops	Different controllers use different I/O drops: <ul style="list-style-type: none"> • If you have a controller that uses Channels instead of I/O Drops, type the number the number of channels the controller I/O will use. Channels always come in pairs, so this must be an even number. • If you have a Micro series controller in parent mode, type the number of children (0 to 4) associated with it. • If you have a Micro series controller in child mode, type the Child ID (1 to 4).
I/O Words	Quantum controllers only: type the maximum number of I/O words that will be available for programming in the Traffic Cop. Check your controller documentation for the appropriate values.
DCP Drop ID	680, 685, 685E, 780, 785, 785E or 785L controllers only: If another computer uses this controller for distributed control, type the controller's Distributed Control Processor ID number (0 to 32).
I/O Time Slice	Type the amount of time the controller will devote to peripheral port communication, from 1 to 100 milliseconds. The default is 10 ms for 984A, B, and X controllers and 20 ms for Quantum controllers.
Remote Channels	984A-S901, 984B-S901, and 584 controllers only: Select from 2 to 32 remote I/O channels. Channels always come in pairs, so this setting must be an even number.

Parameters	General tab – Center of window
Total Messages	Controllers that support ASCII messaging only: Type the total number of ASCII messages the controller will use.
Message Words	Controllers that support ASCII messaging only: Type the length (in machine words) of your controller's ASCII messages. One machine word is equal to two ASCII characters.
ASCII Ports	Type the number of ASCII ports on the controller. This setting must be an even number.
B984 Controller	Select the type of B984 controller you have: B884 or B886.
Input Latched	Select this option if you want the controller to freeze the input state on power-down. Inputs remain frozen for one scan when the controller is powered back up.
Settling Time	<p>Group Settling Time allows you to set parameters for input digital filtering.</p> <ul style="list-style-type: none"> • Each input value is polled through a filter based on a <i>time window</i>. You can set the duration of the time window between 200 microseconds (μs) and 19.8 ms (in 200 μs increments). • The module samples each input every 200 μs and uses the time window associated with the input to determine the input state. If the scanned input state is OFF, the input state is set OFF. If the scanned input state is ON, the input state is set based on a history queue. If the input was ON one time window ago, the input is set ON; if the input was OFF, the input state is set OFF. • The inputs are divided into four groups. The Settling Time of each can be configured separately: <ul style="list-style-type: none"> • Group 1: Inputs 1-4 • Group 2: Inputs 5-8 • Group 3: Inputs 9-12 • Group 4: Inputs 13-16

ProWORX NxT User's Guide

Parameters	General tab – Right side of window
Battery Coil	Type the control address for the controller's battery.
Timer Register	Type the register used to store the controller's timer value.
Time of Day Clock	Type the register used to store the controller's clock value.
Watch Dog Timer	Type the timeout delay for the Watch Dog Timer, in tens of milliseconds. The controller adds 250 ms to the value you enter. The default is 0000, which represents 250 ms.
Configuration Extension Size	<p>Type the amount of memory (in machine words) to be set aside for Configuration Extensions. Use the following information to calculate the minimum Configuration Extension size:</p> <ul style="list-style-type: none"> • Configuration Overhead: 1 word • Extension Overhead: <ul style="list-style-type: none"> • TCP/IP: 100 words • Data Protect: 8 words • S980 Address: 8 words • Quantum Hot Standby: 17 words • VME Bus: 9 words • Profibus: Approximately 4096 words; check your Profibus manual for details • Peer Cop: 5 words <ul style="list-style-type: none"> • Links: 3 words per link, plus: <ul style="list-style-type: none"> • Global Output: 4 words per link • Global Input: 2 words overhead plus, for each of up to 64 devices, 1 + twice the number of up to 8 sub-entries (max. 8). The maximum is $2 + (64 * (1 + 2 * 8)) = 1090$ words. • Specific Output: 2 words overhead plus 2 words for each of up to 64 devices. The maximum is $2 + (2 * 64) = 130$ words. • Specific Input: 2 words overhead plus 2 words for each of up to 64 devices. The maximum is $2 + (2 * 64) = 130$ words. • SY/MAX: 52 words
Skips	Select this option to enable skips (segments can be skipped during solves).

Smart Configurator (online only)

While online, when you change the settings for a previously- configured controller, ProWORX NxT can attempt a Smart Configuration. This lets you make configuration changes without losing any logic or Traffic Cop information stored in the controller.

The Smart Configurator prevents you from saving changes to the controller if:

A reduction in controller memory for a particular setting reduces it below the amount used by logic.

For example, the controller might contain logic addresses up to 10160, but you tried to reconfigure the number of 1xxxx addresses to 32.

Modules are deleted that are used in logic.

If this happens, you have two choices:

Make changes to your new configuration so that the addresses currently used in the controller fall within the appropriate address ranges; or

Send configuration changes to the controller without attempting a Smart Configuration. You will lose all logic and Traffic Cop information stored in the controller.

➤ **To change your configuration without a loss of information:**

1. The **Smart Configurator** dialog box displays the required address ranges for a successful Smart Configuration. Make a note of the values displayed in the “Conf” column.
2. Click **Configurator** to return to the controller configuration dialog box. This resets the configuration values to their original values.
3. Allocate controller memory again, making sure the amount for each address type meets or exceeds the values required.
For example, if you need 100 3xxxx registers, you would have to allocate at least 100 addresses in the 3xxxx section.
4. Click **OK**. You’re prompted to attempt a Smart Configuration.
5. Click **Yes**. The new configuration is saved, with no loss of logic data.

- **To change your configuration, accepting a loss of information:**
1. Click **Configurator** to return to the Controller Configuration dialog box. This resets the configuration values back to their original states.
 2. Allocate controller memory again, using the values you require, then click **OK**. You're prompted to attempt a Smart Configuration.
 3. Click **No**. The controller is reconfigured. Logic and Traffic Cop information will be lost.

Controller configuration - Ports tab

From this tab in the Controller Configuration window, you configure the controller's Modbus and ASCII ports. These settings do not affect the communications setup of your PC in any way— only the controller's.



The communication parameters for your PC and PLC must be the same. Set communication parameters for your PC in the Communications Setup dialog box.

- ❖ For more information, see "Setting up communications for your PC" on page 37.

Some controllers support Bridge Mode, which allows you to connect to a Modbus Plus network through their Modbus ports 1. If a Bridge Mode check box appears in the *Controller Configuration - Ports Tab* dialog, your controller supports Bridge Mode while its MEM/DIP switch is in the MEM position. Select to enable Bridge Mode.

Select the Modbus port to configure, then click Setup Port.

If you have a Micro 311, 411, 512, or 612 controller, assign its ports with the drop-down menu.

	Mode	Parity	Stop/Data	Baud	Dev Addr	Delay
Modbus Port 1:	RTU	Even	1	9600	001	01
Modbus Port 2:	RTU	Even	1	9600	001	01
Modbus Port 3:	RTU	Even	1	9600	001	01

Bridge Mode

Setup Port

Micro Port Assignments:

COMM1 COMM2 I/O EXP

Modbus ASCII

➤ To edit a communication port's parameters:



The controller's MEM/DIP switch determines if its communication parameters are taken from memory or an internal DIP switch. If the MEM/DIP switch is in the DIP position, the controller ignores the parameters you set here.

1. Double-click the parameters of the port or select the parameters for the port, then click **Setup Port**.
2. For each parameter, select a value from its list:

Parameter	Description
Mode	Identifies which data protocol (R emote T erminal U nit or ASCII) to use. Both the PC and controller must use the same data protocol. Default is RTU. If you select ASCII protocol for a port, you can only configure its Parity, Stop/Data bits, and Baud settings.
Parity	Adds a check bit to a packet to make the number of binary ones always either odd (Odd Parity) or even (Even Parity). If parity is set to None, the check bit is not added. The PC and controller must use the same parity. Default is Even.
Stop/Data	Sets the number of bits at the end of a packet which prepare the receiving device for the next packet. The PC and controller must use the same number of stop bits. Default is 1.
Baud	Sets the data transfer speed of the controller's Modbus port in bits per second. The PC and controller must be set to the same baud rate. Default is 9600.
Head	Quantum controllers only: sets the slot number on the local drop backplane where the CPU resides.
Dev Addr	Assigns the Modbus address for the port. Default is 1.

Parameter	Description
Delay	Sets the amount of time in <i>tens of milliseconds</i> the controller waits after receiving a message before sending an acknowledgment. Default is 1 (10); max. is 20 (200).
Modbus Port 2 Type	Momentum M1 controllers only: Sets the port type of Modbus Port 2, to either RS232 or RS485.

3. When you are finished, do one of the following:
 - Press ESC. The lists for each parameter disappear.
 - Click another port. The first port's lists disappear.
 - Click another Controller Configuration tab. The Ports tab closes.
 - Click **OK**. The Controller Configuration window closes.

➤ **To change Micro port assignments for 311, 411, 512, or 612 controllers:**

Select port settings from the Micro Port Assignments drop-down list.

The assignments in the list are combinations of the controller's COM1, COM2, and I/O Expansion Link port. The assignments available depend on which Micro I/O Mode you set in the Edit Details section of the Controller tab in the Controller Configuration window.

- ❖ For more information, see "Editing a controller's hardware details" on page 67.

Configuring simple ASCII ports

Some Modicon controllers have RS-232 ports as part of Remote I/O drops. Advanced programmers can use these ports (called simple ASCII or ASCII/DAP ports), for serial communications between controllers and data terminal equipment. In ladder logic, use the Block Move (BLKM) instruction to send and receive Simple ASCII messages.

Set communication parameters for simple ASCII ports in the Ports tab of the Controller Configuration window.



Set the number of simple ASCII ports your controller has in the ASCII Ports field of the General tab. Otherwise, the ASCII ports will not appear in this window.

- ❖ For more information, see “Controller configuration - General tab” on page 57.

➤ **To edit simple ASCII port parameters:**

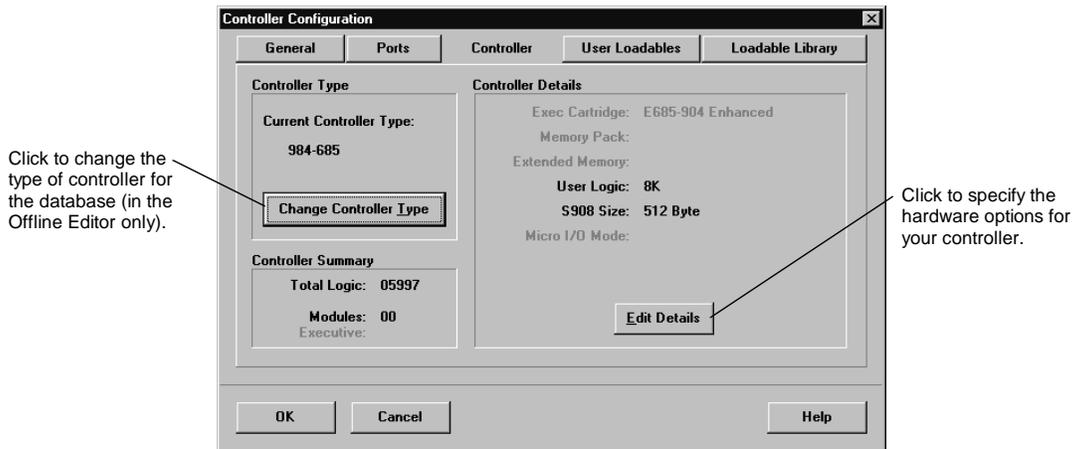
1. Do one of the following:
 - Double-click the parameters for the port to edit.
 - Select the parameters for a port, then click **Setup Port**.
2. For each parameter, select a value from its drop-down list:

Parameter	Description
Parity	Adds a check bit to a packet to make the number of binary ones always either odd (Odd Parity) or even (Even Parity). If parity is set to None, the check bit is not added. The PC and controller must use the same parity setting. Default is even.
Stop/Data	Sets the number of bits at the end of a packet which prepare the receiving device for the next packet. The PC and controller must use the same number of stop bits. Default is 1.
Baud	Sets the data transfer speed of the controller’s Modbus port in bits per second. The PC and controller must be set to the same baud. Default is 9600.

3. When you are finished, do one of the following:
 - Press ESC. The drop-down lists for each parameter disappear.
 - Click another port. The first port’s drop-down lists disappear.
 - Click another Controller Configuration tab. The Ports tab closes.
 - Click **OK**. The Controller Configuration window closes.

Controller configuration - Controller tab

This tab in the Controller Configuration window contains a summary of the hardware options you have chosen for your controller, the amount of memory it has available for logic (in machine words), and how many User Loadable modules are currently loaded into it.



The controller tab also allows you to change your hardware options and controller type, if needed.



You can only change your controller type in the Offline Editor.

Editing a controller's hardware details

Many Modicon controllers accept executive cartridges or optional upgrades such as memory packs. These hardware options should be specified in the Edit Controller Details dialog box so that NxT can adjust your database for them.

- **To specify hardware options for a controller (Offline Editor only):**
 1. In the Controller Tab, click **Edit Details**. NxT displays the options that your controller supports.
 2. Select a new value from the drop-down lists.

Parameter	Description
Executive Cartridge	Cartridges which determine the controller's instruction set. Select the one installed in your controller.
Memory Pack	The amount of both Extended and User Logic memory in the controller. Select the amount installed in your controller.
Extended Memory	Additional memory providing 6xxxx registers. Select the amount installed in your controller.
User Logic	Memory available for ladder logic. Select the amount of memory you want to use for ladder logic from the total amount available in your controller.
S908	Select either 512 or 1024 (1k) input and output points per drop.
Micro I/O Mode	Micro controllers only. Select: <ul style="list-style-type: none"> • Single: The controller is independent, not in a parent/child relationship. • Parent: The controller is the parent in a parent/child relationship. • Child: The controller is the child in a parent/child relationship.

3. When you are finished, click **OK**.

Changing the type of controller for your database

Change your controller type. You may need to do this if you replace an older controller with a new one (say, that is faster or handles more I/O) but want to keep the same ladder logic, or if the wrong controller type was selected while setting up a database. For example, you can convert a database from a 584 controller to a 984 controller, or from S901 remote I/O to S908 RI/O.



The settings in the Configuration Window - General tab **must be valid** for the new controller or NxT does not allow the change.

❖ For information on adjusting the General tab settings, see "Controller configuration - General tab" on page 57.

➤ To change the controller type for a database (Offline Editor only):

1. In the Controller tab, click **Change Controller Type**. The Change Controller Type dialog box appears.
2. From the list on the left, select a controller group. A list of controllers in that group appears on the right.
3. From the list on the right, select the type of controller for the database, then click **OK**.

Controller configuration - User loadables tab

User loadables are instructions that do not come with a controller. Instead, they are supplied by Modicon or third-party vendors and must be copied into a controller or database to be used.



You can only copy loadables that are already part of a loadable library. Using a loadable is a two-step process: reading it into a loadable library, then copying it from the library into the controller or database. To read a loadable from a disk, translate it into NxT format and place it into a library.

❖ For more information, see "Controller configuration - Loadable library tab" on page 70.

In the User Loadables Tab, you copy user loadables from a *loadable library* into a controller or database. You can also delete loadables from your controller or database, if needed.

Select a loadable from the list and click **Load** to copy it into your database (Offline) or controller (Online).

Click **Select Library** to switch to another library.

Loadables in the controller (Online Editor) or database (Offline Editor) are listed here.

To delete a loadable, select it, then click **Delete**.

➤ **To copy a user loadable into a controller or database:**

1. In the User Loadables tab, click **Select Library**. The Open dialog box appears.
2. Type the path and file name of the library or navigate to it, then click **OK**. NxT reads the loadables in the library, and displays them in the left column.



Loadables are meant to be used only with a certain controller. If you try to copy a loadable into the wrong type of controller, you receive an error message. The Library Type line tells you which family of controllers the library is meant for: 584, A, B, X, or x80 (which includes controllers such as 381, 685, 785e, and so on).

3. Select the loadables you want to place into your database or controller from the list.

4. Click **Load**. NxT copies the loadables into the database or controller, then displays them in the right column.



Each loadable is identified in the controller by a unique number called an *opcode*. If a loadable's opcode is already used in the database or controller, NxT prompts you to select a new one.

5. Click **OK**.

➤ **To delete a user loadable from a controller or database:**



You can only delete a loadable from your controller or database if it is not used anywhere in your ladder logic. If you try to delete a loadable that is still in use, you receive an error message.

1. In the right column of the User Loadables tab (labelled Loadables in Controller or Loadables in Database), select the loadable to delete.
Click **Select All** to delete all loadables in a controller or database.
2. Click **Delete**.
3. When NxT prompts you, click **Yes** to confirm the deletion or **Cancel** to exit without deleting the loadables.

Controller configuration - Loadable library tab

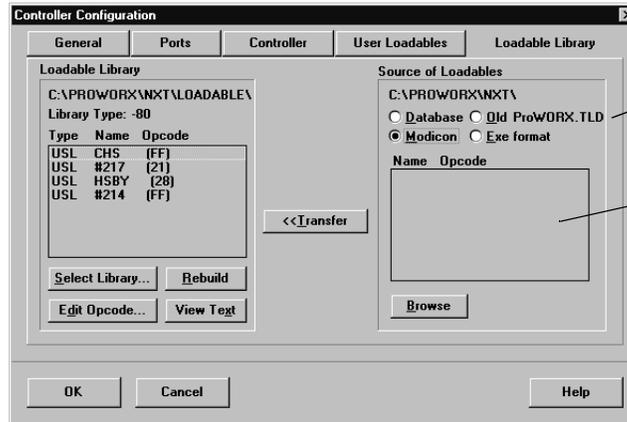
Before you can place a user loadable into a controller or database, it must be read into a *loadable library*. A loadable library holds a set of loadables for you so that you only have to translate them to NxT format once.

In Loadable Library tab, you read a loadable from disk, translate it to NxT format, and move it into a library. You can read loadables from an existing database, a third-party disk or a ProWORX .TLD library file.



Reading a user loadable into a loadable library does not copy it into your controller or database. Using a loadable is a two-step process: reading it into a loadable library, then copying it from the library into the controller or database.

- ❖ For more information, see "Controller configuration - User loadables tab" on page 68.
-



Select the type of file to translate, then click **Browse** to find the file.

Loadables in the source file are listed here. To add a loadable to your library, select it, then click **Transfer**.

➤ **To transfer loadables from a source file into a library:**

1. In the Source of Loadables column, select the type of source file (the file containing the user loadables) to translate to NxT format. You can read loadables from an existing database, a third-party disk, or a ProWORX .TLD library file.
2. Click **Browse** to navigate to the source file, then click **OK**. Most often, source files from other vendors come on floppy disks. NxT reads the loadables in the source file, then displays them in the right column.
3. Select the loadables you want to transfer to a library. You can select more than one loadable at a time.
4. In the **Loadable Library** column, click **Select Library**. The Open dialog box appears.
5. Type the path and file name of a library or navigate to it, then click **OK**. If NxT can't find the path and library name you typed, a new library is created with that name.
6. Click **Transfer**. NxT translates the loadables and copies them into the library.

➤ **To display the manufacturer's notes for a loadable (if available):**

1. Select a loadable from the **Loadable Library** column on the left.
2. Click **View Text**. If it is available, the manufacturer's documentation for the loadable appears.
3. To close the window, click **OK**.

Editing the opcode for a user loadable

An *opcode* is a unique number identifying a loadable in the controller or database. Because each opcode must be different, they may need changing.

➤ **To edit the opcode for a user loadable:**

1. Select a loadable from the **Loadable Library** column on the left.
2. Click **Edit OpCode**. The Edit Opcode window appears, displaying the Function Name, Current Opcode associated with the loadable, and a list of additional opcodes to select from.
3. Select an opcode, then click **OK**.

Rebuilding a damaged library

On rare occasions, a library of loadables can become damaged (for example, when a computer crashes and files are corrupted). NxT allows you to recreate a library from its .USL and .MSL files.

➤ **To rebuild a damaged library:**

Click **Rebuild**. NxT recreates the damaged library. This operation may take a few moments.



Don't rebuild libraries that are not damaged. Doing so may cause certain loadables not to function or to function erratically.

Configuring ASCII messages

ASCII messages let your controller report information to you in plain language. Use them, for example, if you want to see a written alert on your screen every time a motor switches on or a printout every hour of how many items have passed by a counter.

Use NxT's ASCII Message window to enter and edit the messages you want your controller or database to send.

Start by configuring your controller or database to use ASCII messages:

Confirm that your controller supports them.

In the General tab of the Controller Configuration window:

Set the **Total Messages** field to the number of messages you want to use.

Set the **Message Words** field to the number of machine words you want to allocate for each message. One word is equal to two ASCII characters. Remember that the SPACEBAR counts as one character.

- ❖ For more information, see “Controller configuration - General tab” on page 57.

Ensure your ASCII ports are configured correctly in the Ports tab of the Controller Configuration window.

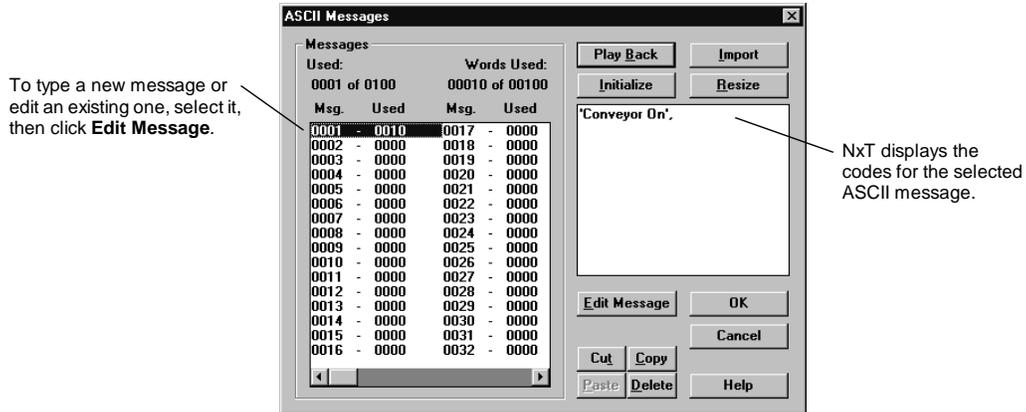
- ❖ For more information, see “Controller configuration - Ports tab” on page 62.



Use the ASCII Read Block (READ) and ASCII Write Block (WRIT) in your ladder logic to send a message from a controller to your output device (such as a screen, printer, or disk drive).

➤ **To display a controller or database's current ASCII messages:**

1. From the Network Editor, on the **Configuration** menu, click **ASCII**. The ASCII Messages window appears.



2. Select a message to view. The raw codes for the message appear on the right.
3. Click **Play Back**. The Message Playback window appears, showing the message as it will appear on a screen, on printer output, or in a text file on a hard drive.

Click **Next** and **Previous** to scroll through the other messages in the controller or database.

4. Click **Close** to close the Message Playback window.

➤ **To remove the ASCII messages in the controller:**

Click **Initialize**. All the ASCII messages are deleted from the controller.

- **To edit the ASCII messages in a controller or database:**
 1. From the Network Editor, on the **Configuration** menu, click **ASCII**. The ASCII Messages window appears.
 2. Select a message to edit, then click **Edit Message** or double-click a message to edit. The Edit ASCII Message window appears.

Click **Play Back** to preview the message as it will appear on a screen, on printer output, or in a text file on a hard drive.



Choose whether new instructions overwrite existing ones or are inserted into them.

These tools insert the codes that make up your ASCII message (see descriptions below).

3. Select **Insert** to add new instructions before the selected instruction or **Replace** to overwrite the selected instruction with new instructions.
4. Enter your edits with the ASCII Tools:

Tool	Description
Text	Enters up to 128 text characters (except for 584 controllers, which accept up to 254 text characters). Spaces at the end of the text are ignored.
Int	Inserts an <i>integer field</i> , a placeholder for an integer which is defined in ladder logic by a WRIT instruction.
Lead 0	Inserts a <i>leading zero integer field</i> , a placeholder for a leading zero integer which is defined in logic by a WRIT instruction.
Bin	Inserts a <i>binary field</i> , a placeholder for a binary number which is defined in ladder logic by a WRIT instruction.
Hex	Inserts a <i>hexadecimal field</i> , a placeholder for a hexadecimal number which is defined in logic by a WRIT instruction.
Oct	Inserts an <i>octal field</i> , a placeholder for an octal number which is defined in ladder logic by a WRIT instruction.

Tool	Description
ASCII	Inserts an <i>ASCII field</i> , a placeholder for ASCII characters defined in ladder logic by a WRIT instruction. You can also use this function to send ASCII characters you cannot enter into a text field (for example, line feeds and tabs).
Space	Inserts from one to 99 consecutive blank spaces. These are represented in code by the number of spaces followed by the symbol X (for example, the code for 42 spaces is 42X).
Repeat	Repeats an item. Click Repeat to start the function. NxT prompts you for the number repetitions. Type the items to repeat, then click Repeat again to end the function.
Control	Inserts a three-digit decimal control character (ranging from 000 - 255) or a three-digit octal control character (ranging from 000 - 377).
<cr>	Inserts a carriage return, represented by the symbol “/”.
Flush	The message buffer contains a 256-byte data field. These bytes contain data values ranging from 0 to FF hexadecimal. The four <i>Flush</i> commands (<i>Flush</i> , <i># Bytes</i> , <i>Incl</i> , and <i>Excl</i>) clear the message buffer. The <i>Flush</i> clears all characters.
# Bytes	This flush command removes from 1 to 255 bytes from the beginning of the message buffer.
Incl	This flush command clears specific groups of data from 1 to 255 times or until a match is found. The terminator value determines how many times the buffer is flushed. The controller stops the buffer flushing when it finds a match for the terminating characters.
Excl	This flush command clears the buffer until a match is found for the terminating character pair. It doesn't flush the match characters. <i>Excl</i> uses 2 registers: the first holds a type identifier and the second hexadecimal values of the terminating pair. Hexadecimal values range from 0000 to FFFF. The controller searches for this range in the buffer. If the second character of the matched pair is not a NUL (00), the next character in the buffer must be equal or the search continues. If the last character tests equal or null, the flush is performed up to, but not including the matched terminators.

➤ **To delete ASCII messages from a controller:**

1. Confirm that the controller is stopped.
 - ❖ For instructions, see “Stopping the controller” on page 121.
2. In the ASCII Message window, select a message to delete.

To delete a message range, select the first one, press SHIFT and click the last one. All the messages in the range will be selected.
3. Click **Delete**. NxT prompts you to confirm the deletion.



In the Online Editor, click **Initialize** to delete all the messages in a controller.

In the Online Editor, you can adjust the amount of memory a controller has set aside for ASCII Messages.

➤ **To change controller memory for ASCII messages:**

1. In the ASCII Message window, click **Resize**.
2. Type the amount of memory (in machine words) to set aside in the controller for ASCII messages, then click **Resize**.

➤ **To import ASCII messages from another database:**

1. From the Network Editor, on the Configuration menu, click **ASCII**. The ASCII Messages window appears.
2. Click **Import**. The Import Messages window appears.
3. Select a database from the list, type the database’s path and file name, or click **Browse** to navigate to it.
4. In the **Source** area, select a range of messages to import.
5. In the **Destination** area, type a location for the first imported message. The other messages follow sequentially. The range for this value is from 1 through the maximum number of messages your controller can support, less the total number of messages you are importing.
6. Select **Replace Always** to overwrite a message without being warned or **Prompt on Replace** to receive a warning message when an existing message is about to be overwritten.
7. Click **Start Import**. If Prompt on Replace is enabled, a warning appears before an existing message is replaced by an imported one.

Using configuration extensions

Configuration extensions are utilities which can be loaded into a controller.



Before using a configuration extension, set the configuration extension size in the General tab of the Controller Configuration window.

❖ For more information, see Controller configuration - General tab on page 57.

Available configuration extensions include:

TCP/IP: Configures controllers to connect to a TCP/IP network through a communication card.

Data Protect: Prevents specific blocks of 0xxxx and 4xxxx references from being modified by general Modbus data write commands.

S980 Address: Specifies the S980 station address.

Quantum Hot Standby: Allows additional configuration of the Quantum Hot Standby setup.

Peer Cop: Allows you to configure data blocks to be transferred between controllers on a Modbus Plus network.

VME Bus: Configures a VME controller for communications with a VME network.

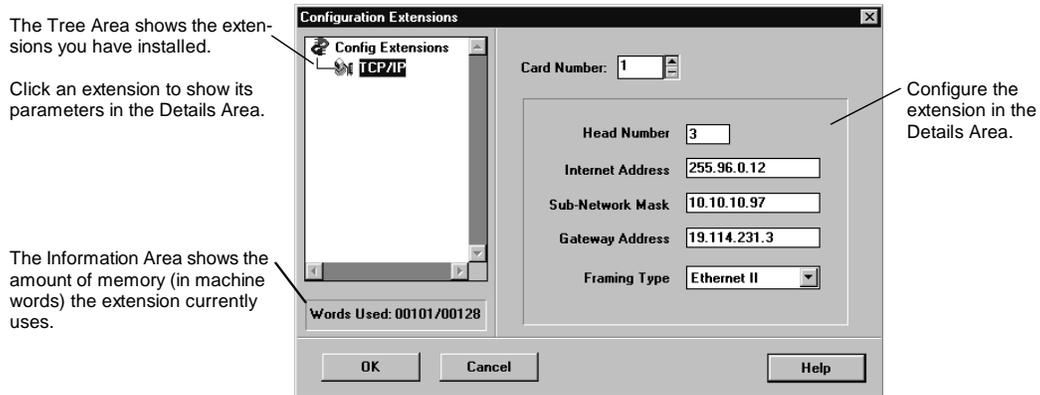
Profibus: Configures a Quantum controller for Profibus communications.

SY/MAX: Configures a Quantum controller for communication with SY/MAX drops.

Cyclic Data Exchange: Allows you to configure data blocks to be transferred between controllers on a TCP/IP network.

➤ **To add a Configuration Extension:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions dialog box appears.



The dialog box is divided into three areas:

Tree Control: Shows the extensions already configured for this controller.

Details: Displays extension-specific parameters.

Information: Displays the machine words of memory used (out of total words allocated).

2. Click **Add Extension** or right-click in the Tree Control area, then select **Add Extension**. The Add Configuration Extension dialog box appears.
3. Select the extension you want to add by clicking it, then click **OK**. You're returned to the Configuration Extensions window.
4. Click **OK** to complete the addition.

➤ **To delete an extension:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions window appears.
2. Click **Delete Extension** or right-click in the Tree Control area, then select **Delete Extension**. The Delete Configuration Extension dialog box appears.
3. Click the extension you want to delete, then click **OK**. You are returned to the Configuration Extensions window.
4. Click **OK** to complete the deletion.

TCP/IP extension

Before your controller can connect to a TCP/IP network, you must install and set up the TCP/IP configuration extension. This extension lets the controller recognize its TCP/IP communication card.

➤ **To set up the TCP/IP configuration extension:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extension dialog box appears.
2. In the Tree Control area of the window, click **TCP/IP**. The TCP/IP configuration parameters appear in the **Details** area.
3. In the **Card Number** field, select a communication card to set up.



Different controllers support different numbers of communication cards:

Quantum 113 Rev. 2 and 213 Rev. 2 controllers support two cards

Quantum 424 Rev. 2 supports up to six.

M1E Momentum controllers only support one card and the Head Number is fixed at 1.

4. (For M1E Controllers only) Select **Extension** to allow the PLC to use the IP Address setting in the TCP/IP Configuration Extension dialog box to the IP address on startup.
5. (For M1E Controllers only) Select **BOOTP Server** to allow the PLC to send an IP request to a BOOTP server. When the PLC Receives a valid reply it uses the information from the BOOTP server to set it's IP Address.

6. In the **Head Number** field, type the number (from 1 - 16) of the slot where the communication card is installed. If you have not yet installed a communications card or you do not want use an installed card, type **0**.
7. Type the TCP/IP address you want the card to have into the **Internet Address** field.
8. If your network uses a sub-network mask, type its address into the **Sub-Network Mask** field. This address is assigned by your network administrator when a network gateway is set up. It, in conjunction with the TCP/IP address, identifies which network the controller is on.
9. If a gateway connects the PLC to your PC's network, type its address in the **Gateway** field. A gateway connects two networks that would not normally be able to communicate with each other. Its address is assigned by your network administrator when the gateway is set up.
10. Select the framing protocol your network uses. Your network administrator determines this protocol when the network is set up.
11. Click **OK**.

Data Protect extension

Data Protect is used to protect specific 0xxxx and 4xxxx references from being modified by Process Monitoring and Control software. Write access is allowed for all 0xxxx and 4xxxx references within the specified block.

References outside the ranges specified are protected from general Modbus data write commands. By default, all 0xxxx and 4xxxx references are unprotected.

➤ **To edit a data protect extension:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions window appears.



If the Data Protection extension does not appear in the Tree Control area, it is not yet installed.

2. In the Tree Control area of the window, click **Data Protection**. The Data Protect dialog box appears in the **Details** area of the window.
3. In the **Start (0X)** field, type the first address in the unprotected range.

4. In the **Length** field (of the 0x area), type the number of references to be left unprotected. NxT shows the address ranges left protected (the ranges of addresses outside the range you just specified).

Suppose you type a Start value of 00017, and a Length of 1024. The first *protected* range will be from 00001 (the first possible value) to 00016 (the last value before the Start value you entered).

There will then be an *unprotected* range from 00017 to 01040 (00017 + 1024). All references above 01041 are also *protected*.

If you enter a length that is too high (goes beyond the addresses configured for the controller) NxT automatically sets the value to include all addresses above the Start field.

5. In the **Start (4X)** field, type the first address in the unprotected range.
6. In the **Length** field (of the 4xxx area), type the number of addresses to be left unprotected. NxT shows the address ranges left unprotected (the ranges of addresses outside the range you just specified).
7. Click **OK**. With Symbols turned on in system Preferences, when you enter an address to be protected and the address has a symbol reference, the symbol appears. If you forget the address for the symbol displayed, you can disable the symbol function in system Preferences.

S980 Address extension

The S980 Configuration Extension stores the S980 station address. This address is then used as part of the S980's mismatch detection mechanism; when the controller is powered up, the S980 checks to see if it has been moved to a different 984.



If the S980 Address extension does not appear in the Tree Control area, it is not yet installed.

➤ To enter the S980 Address:

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions window appears.
2. In the Tree Control area of the window, click **S980 Address**. NxT displays the current S980 address in the **Details** area of the window. If this is the first time you have activated the S980 Address function, a default address of 000000000 appears.
3. Type the new S980 address, then click **OK**.

Quantum Hot Standby extension

The Quantum Hot Standby extension allows additional configuration of the Quantum Hot Standby setup. This lets you set the type of state RAM transfer between the CHS110-00 modules. It also lets you set the Command Register and the Non-Transfer Area. This area is only available on version 2.x Quantum controllers that contain the CHS loadable in the configuration.



If the Quantum Hot Standby extension does not appear in the Tree Control area, it is not yet installed.

- **To configure the Quantum Hot Standby extension:**
1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions dialog box appears.
 2. Select the HSBY extension from the tree control. If the HSBY extension is not in the Tree Control, you have to add it, by clicking **Add Extension**, then selecting **Quantum Hot Standby**.
 3. In the **Command Register Address** field, type the 4xxx address of the command register used to configure the hot standby system. This register must be transferred every scan and cannot be in the non-transfer area.

The initial command register contains the settings that are loaded into the controller when it is started. If any changes need to be made while the controller is running, the command register must be used, and not the initial command register. Settings such as port address swapping, allowing an executive upgrade, setting the standby's mode on a logic mismatch, setting the controller's modes and overriding the key switch can be changed from the command register.

4. In the **Non-Transfer Start Address** field, type the starting address of the range of registers that are not to be transferred from the primary controller to the standby. This is commonly used to reduce scan time.

The first two registers are used in reverse transfer operations. These registers allow information to be passed from the standby to the primary controller.

The third register is the Status register, which stores the status of both controllers. This register provides information on how the hot standby system is operating, such as the power flow of the CHS instruction, position of the controller's A/B switch, and whether there is a logic mismatch between controllers.

All registers following the third register are ignored (not transferred) during the scan.

5. In the **Non-Transfer Area Length** field, type the length of the non-transfer register range. This value can be from 4 through the total number of registers configured in the controller.
6. Select one of the following **State Ram Transferred** options:

Default (12K): All 0xxxx and 1xxxx registers (up to 8192 each) are transferred.

If 10000 or fewer 3xxxx and 4xxxx (combined) registers are configured, then all are transferred.

If more than 10000 3xxxx and 4xxxx (combined) registers are configured, then (up to) 1000 3xxxx registers and all 4xxxx (up to a combined total of 10000) are transferred.

Routine only: All addresses defined in the routine transfer table are transferred every scan. There must be a minimum of 16 4xxxx registers to support the non-transfer area. The Routine Transfer Table is a range of discretely defined registers that must be configured as a multiple of 16.

Routine and Extra: All addresses defined in the Routine Transfer Table and in the extra tables are transferred. The range of each extra table must be a multiple of 16. The extra tables can be transferred over multiple scans.

All State RAM: All RAM configured in the controller is transferred every scan.

7. In the **Scans Needed to Transfer** field, type the number of scans (1-255) needed for the primary controller to transfer the extra transfer tables to the standby, then click **OK**.

VME Bus extension

The VME Bus extension lets a VME-424/X controller control data transfers between devices on a master/slave Quantum network. In a master/slave protocol, one device (the “master”) has control over other devices (“slaves”). As the network runs, each element can lose and gain master status, based on negotiations with other members of the network.



If the VME Bus extension does not appear in the Tree Control area, it is not yet installed.

- **To configure an installed Quantum VME Bus extension:**
 1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions dialog box appears.
 2. Select **VME Bus** from the list.
 3. Type the appropriate value in the **Slave Interrupt Level** field. Boards on a VME Bus can send and respond to messages on seven interrupt levels, numbered from 1 to 7. This field determines which interrupt level the board uses when it's acting as a slave.
 4. Type a value between 1 and 255 in the **Status ID** field. When the VME controller receives an interrupt while acting as a slave, this is the value it sends.
 5. Select an appropriate value for **Master Release Mode**. This field determines when a board acting as a master relinquishes its master status. Valid settings are Release on Request (ROR), Release When Done (RWD), Release On Clear (ROC), or Bus Capture and Hold (BCAP). The proper setting depends on how your Quantum network is configured.
 6. Select an appropriate value for **Master Arbitration Type**. This field determines how the controller will operate. Valid settings are Not System Controller, Primary Mode (PRI) or Round Robin Mode (RRS).
 7. Select the appropriate value for **Master VMEBus Request Level**. This field determines what priority the board has when trying to acquire master status. It can range from BR0 (the lowest) to BR3 (the highest).

8. For each interrupt level from **Interrupt 1** to **Interrupt 7**, select whether it should be Enabled or Disabled. These fields will only have an effect when the VME acts as a master. If an interrupt level is:
 - Enabled:** The controller responds to any messages sent on that interrupt.
 - Disabled:** The controller ignores them.
9. Click **OK**.

Profibus extension

The Profibus configuration extension allows you to communicate with a Profibus network. In order to successfully install and configure the Profibus configuration extension, you will have to use a total of three separate utilities:

The NxT Profibus utility

Softing's PROFI-KON software, included with the Profibus hardware

Modicon's SPU931 utility, also included with the Profibus hardware



PROFI-KON and SPU931 are not ProWORX NxT products. While every effort has been made to ensure the accuracy of these instructions, users of PROFI-KON and SPU931 do so at their own risk.

Installing the Profibus configuration extension

Step 1

Ensure your database is configured properly for your Quantum controller by checking:

Your controller has Revision 2.xx or later executive Flash RAM.

Enough controller memory is set aside for configuration extensions. If you do not have enough configuration extension memory available, you're prompted with an error when you transfer your Profibus configuration into your NxT database (Step 5).

Your Profibus DP or Profibus FMS card is configured in the Traffic Cop.

Step 2

Start the NxT Profibus Utility and create a Profibus Station from your NxT database's .MDF file.

A Profibus Station is a collection of Profibus configuration files contained in a specific directory structure. These files contain such information as:

- Controller configuration;
- Profibus bus and communication parameters;
- Profibus network topology; and
- Profibus-to-Modicon address maps.

➤ **To create the Profibus Station in the NxT Profibus Utility:**

1. In the **.MDF File** box of the NxT Profibus utility, type the name and path of your NxT database's .MDF file or click **Browse** to navigate to it.
2. In the **Profibus Station** area, select a drive for your Profibus Station from the drop-down menu.
3. Do one of the following:
 - Use existing Plant and Station Directories by selecting them from the drop-down menus.
 - Create new directories by typing a Plant name in the **Plant Directory** box, then a Station name in the **Station Directory** box.
4. In the **Options** area, select **Create a Profibus Station** for use with the SPU Software.
5. Click **Start**. The Profibus Utility creates a Profibus Station in the target directories with information from the .MDF file. Its progress is displayed in the Status box.

Step 3

Start the PROFI-KON software you received with your Profibus hardware and configure your Profibus network with it. Refer to its manual for more information. Save your work as a .CNF file to use in Step 4.

To set up a Profibus network, you must have Softing's PROFI-KON software. You received the correct version with your Profibus hardware:

Network	Software	Also called
Profibus DP	PROFI-KON-DP	SPU832
Profibus FMS	PROFI-KON-FMS	SPU932

When you save your work, PROFI-KON generates the .CNF file required by the SPU931 software, which you use in the next step.

Use PROFI-KON-DP to:

- Load device-specific data from .DDB files;
- Set Profibus Master/Slave communication parameters and addresses;
- Edit bus parameters; and
- Check the configuration.

Use PROFI-KON-FMS to:

- Load device master data from .DMD files;
- Edit bus parameters, vectors, communication relationships, and object dictionaries; and
- Define your network's topology.

Step 4

Start the SPU931 software you received with your Profibus hardware. Navigate to and open the Profibus Station you created in Step 2, then the .CNF file you created in Step 3. Map Profibus addresses to Modicon addresses. Refer to the SPU931 manual for more information. Save your work to your Profibus Station.

To set up a Profibus network, you must have Modicon's SPU931 software. You should have received the correct version of Modicon's SPU931 software with your Profibus hardware.

SPU931 requires two files:

- A .CFG file created by the Profibus Utility as part of your Profibus Station; and
- A .CNF file generated by the PROFI-KON software.

SPU931 can only save a Profibus configuration to an existing .CFG file, not create a new one. Create a .CFG file with the Profibus Utility before configuring your Profibus network with SPU931.

When you save your work, SPU931 updates the Profibus Station's .CFG file. The Profibus Utility then imports the updated information from the .CFG file back into your NxT database in Step 5.

Use SPU931 to:

Map Profibus I/O points and diagnostics to registers in your controller's memory; and

Assign bus controllers and the map the topology of the bus.

Step 5

In the Profibus Utility, transfer information from the updated Profibus Station into your NxT database. NxT now shows the Profibus configuration extension in the Configuration Extensions area.

➤ **To transfer a Profibus Station:**

1. In the **.MDF File** box, type the name and path of your NxT database's .MDF file or click **Browse** to navigate to it. The Profibus Utility transfers the Profibus configuration to this file.
2. In the **Profibus Station** area, select the drive where your Profibus Station is located from the drop-down menu.
3. Select your Profibus Station's **Plant** and **Station Directories**, from the drop-down menus. The Profibus Utility reads the Profibus configuration from the .CFG file in the Station Directory.
4. In the **Options** area, select **Move Profibus configuration to ProWORX database**.
5. Click **Start**. The Profibus Utility transfers the Profibus configuration from the Station Directory to your database's .MDF file. Its progress is displayed in the Status box. NxT now shows the Profibus configuration extension in the Configuration Extensions area.

Editing your Profibus configuration

To edit your Profibus configuration:

Without changing your controller configuration, start again at Step 3.

After changing your controller configuration, start again at Step 2.

Peer Cop extension

The Peer Cop extension provides data transfer between two or more controllers on a peer-to-peer network, as well as linking multiple networks using the S985 communication card. Peer Cop configures data blocks to be continuously transferred (once per scan) between nodes on a Modbus Plus network. Data can be broadcast to all nodes on a single link (Global I/O) or between specific Modbus Plus nodes on a link (Specific I/O).

A maximum of 32 data registers or 512 (for example, 32*16) I/O points can be transferred to or read from a controller at a time. Peer Cop is supported by the A145 and all E-Series and Quantum controllers. Up to three Links of the Peer Cop can be configured and edited.

Peer Cop includes a configuration wizard to help you set your extensions. To access the wizard, see "Peer Cop wizard" on page 97.

Adding links in the Peer Cop extension

The Peer Cop extension can be configured for up to three links. When you add a link you will have access to the 64 possible devices on another peer-to-peer network.

Link 1 is the internal link; all devices on the local Modbus Plus network can be accessed from Link 1.

Links 2 and 3 are remote links through S985 cards.

➤ To add a Peer Cop link

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions window appears.
2. In the Tree Control area of the window, click **Peer Cop**. Peer Cop details appear in the **Details** area of the window
3. Click **Add a New Link**. When you add a new link, the new link is also automatically added to the tree control display. Each link includes a Global I/O and Specific I/O transfer.

Editing links in the Peer Cop extension

You can edit the following components of the link:

- General link information;
- Global I/O; and
- Specific I/O.

General link information

The three general components of a link are the **Last Value**, **Timeout**, and **Head Number**.

Last Value specifies whether or not to hold the last value. When **On**, the input data area associated with an unhealthy transfer is left in its previous state (i.e. the last value with a health of OK).

Timeout specifies the health timeout interval. The default value is 500ms. This value specifies the minimum time period a Peer Cop configured communication must fail before the associated health bit is cleared. Valid timeout values range from 20ms to 2 seconds. If you type a value too big, the value truncates to a multiple of 20. For example, 230 truncates to 220 (it is not rounded up to 240).

Head Number specifies the head number on a Quantum rack. If you are using a Quantum Controller, you have the option of editing the head number for the second or third link. The first link is internal, therefore it cannot be edited.

Global I/O

Global I/O is one of two communication methods used by the Peer Cop extension (the other is Specific I/O). Global I/O is a *broadcast* communication method, where a message is broadcast (made available) to all controllers on the Modbus Plus network. Global I/O data transfers do not require an acknowledgment from the receiving controller, so there is no immediate overhead placed on the receiving controller.

There are two components to Global I/O:

Global Output broadcasts the specified range of discretely or registers to the devices on the Modbus Plus network. Each device used to access the data must also be configured to accept Global Input from the broadcasting device (1-32).

Global Input is used to receive global data from any device on a Modbus Plus network. One entry (line) is available for each device (1-64). Each device's global data can also be received in pieces determined by Subfields.

Global Output

Global Output broadcasts the specified range of discretes or registers onto the Modbus Plus network. Each device used to access the data must also be configured to accept Global Input from the broadcasting device.

➤ **To edit the Global Output:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions window appears.
2. In the Tree Control, select the Global I/O to edit by clicking it. The **Global I/O** fields appear in the Details area of the window.
3. Do one of the following:
 - Select the **Global Output** field by clicking it, then click **Edit**.
 - Double-click the **Global Output** field.The Global Output dialog box appears.
4. In **Length**, type the length (1-32) of the address range (i.e., the number of registers to broadcast).
5. In **Reference**, type the starting address (5 or 6 digit) for the range to broadcast.
6. If required, click **BIN/BCD** to toggle between binary and binary coded decimal modes.

Global Input

Controllers use the Global Input function to receive global data from any device communicating peer-to-peer on a Modbus Plus network.

➤ **To set a Global Input:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions dialog box appears.



Ensure the necessary Peer Cop links are defined before continuing.

2. In the Tree Control, select the Global I/O to edit by clicking it. The **Global I/O** fields appear in the **Details** area.
3. Select a **Global Input**, then click **Edit** or double-click the **Global Input** you want to edit. The Global Input dialog box appears.
4. In **Index**, type the starting point (1 to 8) of the broadcast data to read.
5. In **Length**, type the number of words (1 to 32) to read forwards from the Index value.

For example, suppose the source controller is broadcasting 10 words of data using the Global Output function, but the receiving controller only uses words 3 to 7. You would type an **Index** value of 3 and a **Length** of 5.

The length plus the index must be less than or equal to 33. In the **Reference** field, type the destination for the received data (i.e., where the received data is to be stored).

6. Select either **BIN** (default) or **BCD**. One entry is available for each device (1-64).

Each device's global data can also be received in sections determined by Subfields. You can define a subfield for each block of the broadcast data that you want the controller, to receive while ignoring the remainder.

➤ **To edit subfields:**

1. From the Configuration Extensions dialog box, select the **Global I/O** of the link you want to edit by clicking it.
2. Click the global input you want to edit from the **Global Input** area of the Configuration Extensions window.
3. Click **Subfields**. The Global Input Subfields window appears with a list of the subfields for the selected input. Select the subfield you want to edit.
4. Do one of the following:
 - Select a subfield and click **Edit**.
 - Double-click the subfield.The Global Input dialog box appears when you.
5. Type the new global input parameters.
6. (Optional) Click **Data** to view the data currently in the input registers.

Specific I/O

Specific I/O is one of two communication methods used by the Peer Cop extension (the other is Global I/O). Specific I/O uses a one-to-one communication method and requires an acknowledgment from the receiving device, which creates a certain amount of overhead. When using Specific I/O, the destination controller must accept the entire block of data from the source controller. This means that the Index value used in Global I/O is not necessary.

Specific I/O allows you to configure multiple defined data blocks for transmission to specific devices on the Modbus Plus network. The device receiving the data must be configured for Specific Input from the broadcast device. The length (in words) of the specific input (configured in the destination controller) must be identical in length to the specific output (configured in the source controller). The input data, however, can be stored in any type of reference desired. That is, five words of 4xxxx data can be stored into five words of 0xxxx memory area.

➤ **To edit a Specific Output:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions window appears.



Ensure the necessary Peer Cop links are defined before continuing.

2. In the Tree Control, click the Specific I/O to edit. The **Specific Input** and **Specific Output** fields appear in the **Details** area of the window.
3. In the **Specific Output** list, select a source controller (i.e., the output device) by clicking it.
4. Click **Edit**. The Specific Output window appears, with the selected controller shown in the **Device** field.
5. In the **Length** field, type the length of the data block (1-32) to be sent to the destination controller.
6. In the **Reference** field, type the starting address for the block of data to be sent to the destination controller.
7. Click **OK** to close the Specific Output window.
8. If required, click **BIN/BCD** to toggle between binary and binary coded decimal.
9. Click **OK**.

➤ **To edit a specific input:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions dialog box appears.



Ensure the necessary Peer Cop links are defined before continuing.

2. In the Tree Control, click the Specific I/O to edit. The **Specific Input** and **Specific Output** fields appear in the **Details** area of the window.
3. In the **Specific Input** list, select a destination controller (for example, the input device) and click **Edit**. The Specific Input dialog box appears, with the selected controller shown in the **Device** field.
4. In **Length**, type the length of the data block (1-32) to be received from the source controller.
5. In **Reference**, type the starting address of the block of data to be placed from the source controller.
6. Click **OK** to close the Specific Input dialog box.
7. If required, click **BIN/BCD** to toggle between binary and binary coded decimal.
8. Click **OK**.

➤ **To view the data for a reference address:**

1. In the Tree Control, click the Specific I/O to edit. The **Specific Input** and **Specific Output** appear in the Details area of the window.
2. In the **Specific Output** or **Specific Input** box, select a reference and click **Data**. Data associated with the reference address appears. This is a read-only function— you cannot edit the data.

Clearing and deleting links

There are two different functions available to reset or remove links:

Clearing links deletes all Global I/O and Specific I/O entries within a link, but does not delete the link itself.

Deleting a link removes the link itself from Peer Cop.

➤ **To clear all references within a link (keeping the link in the Peer Cop extension):**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions window appears.
2. In the Tree Control area, select the Peer Cop link you want to clear. You can click the link itself, or the Global I/O or Specific I/O in the link.
3. Click **Clear Link**. NxT will prompt you to confirm the operation.
4. Click **OK**.

➤ **To delete a link from the Peer Cop extension:**



You cannot delete Link 1, the internal link.

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions window appears.
2. Do one of the following:
 - In the Tree Control, select the link you want to delete by clicking it, then click **Delete Link**.
 - In the Tree Control, select Peer Cop by clicking it, then click **Delete a Link**.The Delete Link window appears.
3. Select the link you want to delete by clicking it.
4. Click **OK**. The link is automatically deleted from the tree control display along with its accompanying Global and Specific I/O.

Peer Cop wizard

NxT includes a configuration wizard to step you through transaction setup between a local device and a number of remote devices on a Modbus Plus network.

You can access the wizard from either the Online or Offline Network Editor. Offline configuration changes are made to each individual database and online changes are made directly to each attached controller.

➤ **To launch the Peer Cop wizard:**

On the **Configuration** menu in the Network Editor, select **Peer Cop Wizard**. The Peer Cop Wizard appears.

Click **Next** and **Previous** to step through the wizard.

SY/MAX extension

The SY/MAX configuration extension allows you to properly access and configure up to six SY/MAX RIO cards. This extension is only available when using Quantum controllers revision 2 or later.

➤ **To configure the SY/MAX extension:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions dialog box appears.
2. Select the SY/MAX extension from the Config Extensions tree.
If the SY/MAX extension is not in the Config Extensions tree, you have to add it. Select Config Extensions on the tree, click **Add Extension**, then select SY/MAX.
3. In **Card Number**, select the particular SY/MAX RIO card you want to configure. The Module Drop Number, Backplane Slot, Retry Count and Timeout values change to reflect the current settings of the selected card.



The actual value for the Card Number has no effect, as long as you choose a different Card Number for each SY/MAX RIO card you configure.

4. In **Module Drop Number**, type the number of the drop that the selected RIO card controls.
Type “-1” if the RIO card has no drop to control.

5. In **Backplane Slot**, type the slot number that the RIO card inhabits on the local rack.
If set to 0, the SY/MAX extension assumes that no card exists.
6. In **Retry Count**, type the number of times the controller tries to communicate with the RIO card before it gives up.
7. In **Timeout**, type the number of 10s of milliseconds the controller waits for communications from the RIO card before it gives up.
8. Click **OK**.

➤ **To remove an RIO card from the SY/MAX extension:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The **Configuration Extensions** dialog box appears.
2. Select the SY/MAX extension from the Config Extensions tree.
3. In **Card Number**, select the RIO card number you want to remove.
4. In **Drop Number**, type “-1”.
5. In **Backplane Slot**, type 0, then click **OK**.

I/O Scanner extensions

The I/O Scanner extension provides data transfer between two or more controllers on a TCP/IP network. The I/O Scanner lets you to simultaneously configure up to 128 communication transactions, depending on your controller. Because the TCP/IP connection is established only once and remains connected during an entire session, it makes this type of communication very efficient.

I/O Scanner includes a configuration wizard to help you set your extensions. To access the wizard, see “I/O Scanner wizard” on page 102.

➤ **To configure the I/O scanner extension:**

1. From the Network Editor, on the **Configuration** menu, click **Config Extensions**. The Configuration Extensions dialog box appears.
2. In the Config Extensions tree, right-click on **Config Extensions** and select **Add Extension**.
3. Select **I/O Scanner**. The I/O scanner details appear.

- In the Health Block field, type a 1xxxxx or 3xxxxx address. All 1xxxxx addresses are based on a 16-bit boundary. For example, 100001, 100017, 100033, etc.

IP Address	Unit ID	Health Timeout	Rep Rate	Read Ref Local	Read Ref Remote
001: 010.010.010.113	001	00000	00000	40001	40001
002: 010.010.010.113	001	00000	00000	-	-
003: 010.010.010.113	001	00000	00000	40010	40003 Client
004: 001.001.001.001	000	00000	00000	40001	-
005: 001.001.001.001	000	00000	00000	-	-
006: 001.001.001.001	000	00000	00000	40001	Client Server
007: 001.001.001.001	000	00000	00000	40001	Server
008: 001.001.001.001	000	00000	00000	-	-
009: 001.001.001.001	000	00000	00000	40001	Server
010: Empty					
011: Empty					
012: Empty					
013: Empty					
014: Empty					
015: Empty					
016: Empty					
017: Empty					

- Double-click on an empty transaction to add a transaction or double-click on an existing transaction to edit it. The Transaction dialog box appears.

Transaction dialog box parameters:

- Remote Device IP Address: 10.10.10.52
- Unit ID (0-255): 000
- Health Timeout (0-50000): 00000 mS
- Repetition Rate (0-50000): 00000 mS
- Function: Read/Write
- Read Ref Master: 40001
- Read Ref Slave: 40001
- Read Count (1-125): 001
- Write Ref Master: 40001
- Write Ref Slave: 40001
- Write Count (1-100): 001

- Set the transaction parameters using the information below:

Remote Device IP Address: Type the IP address of the remote device that you are communicating with.

Unit ID: Type the value of the destination Unit ID. This is an identifier for a pair of transactions (specifically Link Client/Server transactions). The transaction pair must have matching Unit ID's.

An example situation would include a single Server Write that sends data to the matching Client Reads in a remote device. All Client Read transactions accept the data sent from the single Server Write transaction as long as the Unit ID's match.

Health Timeout: Type a millisecond value representing the length of time to wait for a reply for each transaction.

Repetition Rate: Type a millisecond value representing the length of time to wait before repeating the transaction. A value of 0 indicates the quickest possible rate.

Function: Supports the following function types

Function	Description
Read	A unilateral read in which a local device reads data from a remote device.
Write	A unilateral write in which a local device writes data to a remote device.
Read/Write	A unilateral read/write in which a local device reads data from and writes data to a remote device.
Link Client Read	A paired function type in which a local device responds to a write transaction from a remote device which must have a matching server write.
Link Client Write	A paired function type in which a local device writes to a remote device which must have a matching server read.
Link Client Read/Write	A paired function type in which a local device reads and writes data to and from a remote device which must have a matching server read and write.
Link Server Read	A paired function type in which a local device initiates a read from a remote device. The remote device must have a matching client write.

Function	Description
Link Server Write	A paired function type in which a local device writes to a remote device. The remote device must have a matching client read.
Link Server Read/Write	A paired function type in which a local device reads and writes data to and from a remote device which must have a matching client read/write.



For unilateral function types, no intervention required for the remote devices. They respond to any Read or Write without the need to set up an I/O Scanner transaction. Linked function types require two complementary transactions, one in each device.

Fallback Value:

Zero: Select to reset the data values for the selected transaction to zero in the event of a power failure.

Hold Last: Select to retain the last data values for the selected transaction and make them available at restart in the event of a power failure.

Read Ref Local: Type the local data address receiving data from the remote controller.

Read Ref Remote: Type the remote address where the read data is coming from.

Read Count: Type the number of sequential registers to read. Up to 125 registers are allowed.

Write Ref Local: Type the local data address that sends data to the remote controller.

Write Ref Remote: Type the remote address where the write data is going to.

Write Count: Type the number of sequential addresses to write. There are up to 100 registers allowed.

I/O Scanner wizard

NxT includes a configuration wizard to step you through data transaction setup between a local device and a number of remote devices on a TCP/IP network.

You can access the wizard from either the Online or Offline Network Editor. Offline configuration changes are made to each selected database only and online changes are made directly to each controller selected.

While online, this wizard only operates using TCP/IP communications and access to the wizard is inaccessible for other communication types.

➤ **To launch the I/O Scanner wizard:**

On the **Configuration** menu in Network Editor, select **I/O Scanner Wizard**. The I/O Scanner Wizard appears.

Click **Next** and **Previous** to step through the wizard.

Compact Phase II extensions

Compact Phase II extensions allow the programming of functions specific to the Compact Phase II controllers. This provides support for:

PLC based password access;

Secure Data Area (SDA), and

Comm 1 CTS/RTS delay time settings.

COMPACT PHASE II	
PLC Password Protection	
Old Password:	<input type="text"/>
New Password:	<input type="text"/>
Confirm New Password:	<input type="text"/>
Secure Data Area (SDA)	
SDA Size (K Words):	<input type="text" value="64"/>
RTU Specific Parameters	
RTS Delay (x10 ms):	<input type="text" value="5"/>
CTS Delay (x10 ms):	<input type="text" value="10"/>

The following are features of the Compact Phase II Configuration Extensions dialog box:

PLC Password Protection: Type a password consisting of a maximum of 16 characters (A-Z, 0-9 and _). The PLC password can only be changed while online.

SDA Size (K Words): Type a Secure Data Area value between 0 and 128. A zero value disable the feature.

RTS Delay (x10 ms) Type a value between 0 and 50. A zero value indicates the feature is disabled.

CTS Delay (x10 ms) Type a value between 0 and 50. A zero value indicates the feature is disabled.

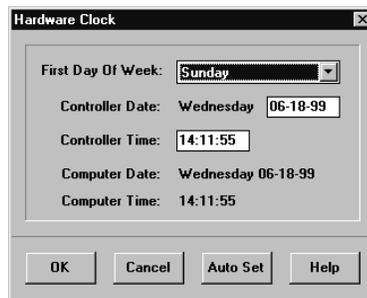
Configuring the hardware (or time of day) clock

Many controllers have a built-in Time of Day clock. You can set these clocks if it's starting register is configured in the General tab of the Controller Configuration window, NxT is set to Program Mode, and the controller is running.

- ❖ For more information, see Controller configuration - General tab on page 57.

➤ To configure the hardware clock:

1. From the Network Editor, on the **Configuration** menu, click **Hardware Clock**. The **Hardware Clock** dialog box appears.



2. In **First Day of Week**, type the day the controller will use as the first day of the week.
3. Do one of the following, then click **OK**.

Auto Set: Click to synchronize the controller's date and time with your computer.

Controller Date: Type the date in mm-dd-yy format.

Controller Time: Type the time in hh-mm-ss format.

The Time of Day clock requires eight 4xxxx registers in your controller.

This register...	Contains...
4xxxx	Control information. From the left: Bit 1: Set Clock Values Bit 2: Read Clock Values Bit 3: Done Bit 4: Error
4xxxx+1	Day of the week (from 1 to 7)
4xxxx+2	Month
4xxxx+3	Day
4xxxx+4	Year
4xxxx+5	Hour (in 24 hour format)
4xxxx+6	Minutes
4xxxx+7	Seconds

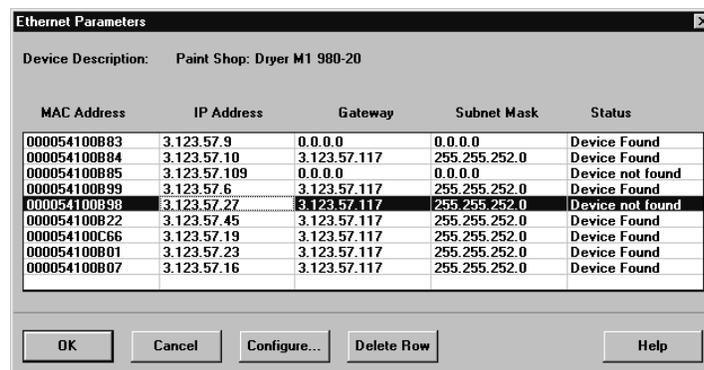
Ethernet configuration

BOOTP support

NxT supports the configuration of the IP Address of an ENT module or Momentum controller via an Ethernet network. This lets you record and configure a device's IP Address and, optionally, the Gateway IP Address and Sub network mask.

➤ **To access Ethernet Parameter setup:**

From the **Tools** menu, select **Ethernet Setup (BOOTP)**. The Ethernet Parameters dialog box appears.



Ethernet Parameters

Below are the features of the Ethernet Parameters dialog box:

Device Description

Displays the description of the associated row selected in the Ethernet Parameters table.

MAC Address (*Media Access Control Address*)

A 12 digit hexadecimal number uniquely identifying an Ethernet device. A device's MAC address cannot be changed. The MAC Address is on a label (currently marked as the "IEEE GLOBAL ADDRESS") on each Schneider Ethernet device.

IP Address

A logical 32-bit address used to identify a TCP/IP device. Each IP address has two parts: the network ID and the host ID. The network IP identifies all hosts (devices) that are on the same physical network. The host ID identifies a specific host on a network. Each computer that runs TCP/IP requires a unique IP address.

The IP Address may be available from or assigned by your network administrator.

Gateway (Default)

For communication with a host on another network, an IP host must be configured with a route to the destination network. If a configured route is not found, the host uses the gateway to transmit the traffic to the destination host. The default gateway is where the IP sends packets that are destined for remote networks. If a default gateway is not specified, communications are limited to the local network.

The Gateway may be available from or assigned by your network administrator.

Subnet Mask

Used to mask a portion of the IP address so that TCP/IP can distinguish the network ID from the host ID. TCP/IP hosts communicate by using the subnet mask to determine whether the destination host is located on a local or remote network.

The Subnet Mask may be available from or assigned by your network administrator.

Status

The existing condition of the ping:

Device Found: NxT has found a Schneider device with this MAC address and IP address

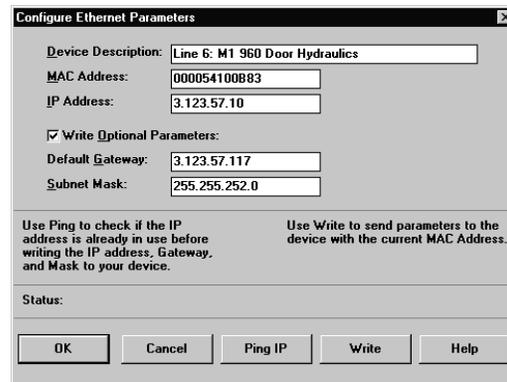
Device not found: NxT could not find a Schneider device with this MAC address and this IP address.

Delete Row

Click to permanently remove a selected row causing all rows below the deleted row to shuffle one row upwards.

Configure

Click to edit the Ethernet setup for the selected device from the Configure Ethernet Parameters dialog box.



The Device Description, MAC Address, IP Address, Default Gateway and Subnet Mask fields default to the settings of the row selected in the Ethernet Parameters dialog box.

Configuring Ethernet parameters

Below are the features of the Configure Ethernet Parameters dialog box.

Write Optional Parameters

Select to write the IP Address, the Gateway Address, and the Subnet Mask to the device.

Ping IP

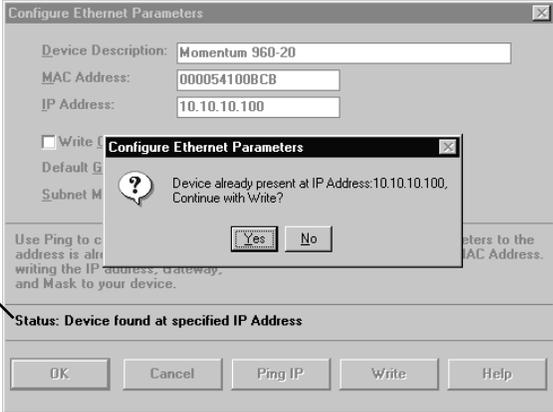
Click to check if there is a device at the specified IP Address. If successful, the Status field displaying the message: "Device found at specified IP Address".

Write

Click to send parameters to the device with the current MAC Address.

➤ **To write the Ethernet parameters to a device:**

1. From the Configure Ethernet Parameters dialog box, click **Write**. The selected device is pinged to see if it exists at the specified IP address. The following message appears:



The Status field keeps you informed during the Write process.

If the device isn't found, you're prompted to send Ethernet parameters to a device with the specified MAC address.



You can stop the Write process by clicking No. "Write aborted" appears in the Status field.

2. Click **Yes** to continue configuring Ethernet parameters. The following message appears:



3. Click **OK** to send the parameters. The message "You should power cycle the Ethernet module now." appears.
4. Unplug the device, wait for about a minute, then plug it in again. As the Ethernet device boots up, it becomes receptive to the IP parameters NxT is sending.



When power cycling ENT modules, wait at least five minutes.

5. Click **OK**. NxT waits for the device's request for IP parameters. If the device replies, NxT uses the information from the request to direct the new IP parameters back to the device making the request.



NxT only replies to IP parameter (BOOTP) requests from a device with the same MAC address as in the Configure Ethernet Parameters dialog box.

NxT sends 10 BOOTP messages containing IP parameters with a three second interval between each message. The parameters are sent and the Status field displays "Writing bootp message (x)".

Ensuring the IP address was accepted by the device

Once the broadcasting of IP parameters is finished, NxT searches for the device at the specified IP address. The Status field displays "Attempting to find, IP address xxx.xxx.xxx.xxx".

If a device is found, NxT verifies that the device has the correct MAC and IP address. If so, NxT verifies which Schneider Electric device is present. NxT then attempts to connect to the device at the new IP. If the device responds, the device type is retrieved.



If the device isn't found, the Status field displays "Write failed. IP address xxx.xxx.xxx.xxx not found."

Unplug the device, wait five minutes and plug it in, then try Writing again.

Below are the following devices that BOOTP can find:

ENT Communication Adapter

Momentum Ethernet PLCs (980-20 and 960-20 both respond as "M1 Ethernet PLC")

NOE771-00 Network Option Module

NOE771-10 Network Option Module.

For all devices, except ENT modules, this is the end of the BOOTP process.

ENT modules: The IP parameters are written to the ENT's flash RAM.

M1E Momentum: Transfer all memory contents including the IP parameters to flash memory by selecting **EEPROM/Flash** on the **Controller** menu.

➤ **To make the TCP/IP address permanent with Quantum controllers:**

1. Once the NOE771 is assigned an address with BOOTP, attach to the controller using the NOE771 via TCP/IP.
2. Configure the TCP/IP configuration extension to the IP address assigned by BOOTP.
3. In the configuration extension match the head number with the location of the NOE771.



For Quantum controllers with flash memory, the IP setting (and rest of the PLC's program) can be written to flash selecting **EEPROM/Flash** on the **Controller** menu.

ENT flash write details

Once the ENT has accepted its new IP address, NxT connects to the ENT with the new IP address. If the connection is successful, NxT sends a command that commits the IP address to the ENT's flash memory.

NxT then attempts to connect to the ENT again. If the connect is successful, the message "Second ENT connect succeeded. To test the new IP, cycle ENT power. After 20 seconds, PING will find ENT." appears.

If the second connect fails, the message "Second ENT connect failed. Repeat BOOTP to write IP to flash." appears.

After successfully configuring the IP parameters of an Ethernet device, return to the Online or Offline main menu to ensure NxT is ready to configure another device.