Network Configuration Editor User's Guide

GC30-4031-00

Network Configuration Editor User's Guide

GC30-4031-00

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page iii.

First Edition (July 1999)

This edition applies to 8371 Networking Multilayer Ethernet Switch Release 2.0 and subsequent releases until replaced.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for readers’ comments is provided at the back of this publication. If the forms have been removed, you may address your comments to:

International Business Machines Corporation Design & Information Development
Department CGF
PO Box 12195
RESEARCH TRIANGLE PARK NC 27709-9990
U.S.A.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Notices

References in this publication to IBM products, programs, or service do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
AIX/6000
NetView
Nways
Operating System/2
OS/2
RS/6000
SystemView

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

Intel and Pentium are trademarks of Intel Corporation in the United States and/or other countries.

Other company, product, and service names may be trademarks or service marks of others.

About This Manual

This manual explains how to install and use the network configuration editor on various platforms to configure supported IBM network devices for your network.

This manual assumes a knowledge of networking concepts.

Who Should Read This Manual

The intended user of this book is the person who will configure an IBM network device using the network configuration editor.

How This Manual Is Organized

This manual is divided into the following chapters:

- "Chapter 1. Obtaining and Installing the Program" on page 1, which describes the installation requirements for the network configuration editor and how to install it.
- "Chapter 2. Starting the Program" on page 5, which describes how to start the network configuration editor and some operational considerations.
- "Chapter 3. Using the Program" on page 9, which describes how to use the network configuration editor, some of its navigational features, and an overview of the configuration process.

Contents

Features of the Program

The network configuration editor enables you to do the following tasks for the devices that it supports:

- Work with multiple configurations at the same time, on all supported devices at all supported releases.
- Save a configuration for later completion.
- Import an existing configuration from a device and either update it or use it as a template for configuring another device.
- Configure multiple protocols. (You may configure them in any order unless there are inter-protocol dependencies.)
- Verify the completeness of a configuration before exporting it to a device.
- Use guided configuration, which displays dynamic help information for configuration tasks.
- Use context-sensitive helps for panels and parameters to get more information as needed.
- Export configurations as backups or store multiple versions for the same device.
- Use the command line facility to automate many tasks including importing and exporting configurations.
- Create, edit, and read ASCII-formatted configuration files.

The network configuration editor provides a point-and click-interface for doing these tasks for the supported devices and models, which you may find more convenient to use than the command line interface

Library Ordering Information

Obtaining Softcopy Information

Displayable softcopy publications are available on a CD-ROM (order number SK2T-0446) for the IBM 8371. This CD-ROM is shipped with initial orders for the IBM 8371.

Chapter 1. Obtaining and Installing the Program

This chapter describes the minimum requirements to run the network configuration editor and the procedure for getting and installing it. It consists of the following sections:

- "Obtaining the Program Electronically" on page 2
- "Product Installation and Starting Information Tables" on page 2
- "Installing the Program" on page 3

Important: Before proceeding, see the README file for the latest information about requirements and any changes to installation procedures.

Hardware and Software Requirements for the Program

Table 1 lists the minimum network configuration editor requirements.

Table 1. Network Configuration Editor Requirements for Supported Operating Systems

Requirement	IBM AIX	IBM Operating System/2	Microsoft Windows 95 or Windows NT
Workstation	RS/6000 Model 250/410 or higher with minimum 80-Mhz CPU	IBM-compatible PC with Intel Pentium 166-MHz or higher processor (see Note 2)	IBM-compatible PC with Intel Pentium 166-MHz or higher processor (see Note 2)
RAM (minimum)	64 MB	64 MB	64 MB
CD-ROM	yes	yes	yes
Free Disk Space	54 MB Additionally, you need 1 to 3 MB for each configuration.	37 MB Additionally, you need 1 to 3 MB for each configuration.	54 MB Additionally, you need 1 to 3 MB for each configuration.
Swapper Size	N/A	10 MB	N/A
Display (minimum)	Graphics (1024 x 768 x 256 colors)	SVGA (1024 x 768 x 256 colors)	SVGA (1024 x 768 x 256 colors)
Mouse	3-button	2-button (see Note 3)	2-button (see Note 3)
Software	AIX for RS/6000 V4.2 or higher AIXWindows Environment/6000	OS/2 3.0 or higher IBM TCP/IP 1.2.1 for OS/2 or higher	Microsoft Windows 95 or Windows NT

Notes:

1. For AIX, the AIX Common Desktop environment (X11.dt) is required to access Help.
2. More memory and faster processors improve the program's performance.
3. Functionally equivalent to a 3-button mouse.

4. You must make sure that TCP/IP is ***installed*** and ***running*** before using either Import or Export for a network device (through SNMP) functions. On MS Windows platforms, the configuration program will lock up if TCP/IP is not installed and running.

Obtaining the Program Electronically

You can obtain the network configuration editor from the IBM Networking Technical Support Home Page. Follow this procedure to download the network configuration editor:

1. Access the IBM Networking page at:

   http://www.networking.ibm.com
2. Select **Support** from the **Fast-Path Site-Links** menu to access the Technical Support page.
3. Select the product from the **Downloads** menu.
4. Select the desired code level for the network configuration editor that you want to download. On the next panel, in the **Files in this Package** box at the bottom of the panel, download the file you want into a temporary directory. (See the README file to determine which version of the network configuration editor and which file you need, and ensure that you have enough space on the hard drive.)
5. To install the network configuration editor, continue at “Installing the Program” on page 3.

Product Installation and Starting Information Tables

The network configuration editor supports the IBM 8371 Networking Multilayer Ethernet Switch on multiple platforms. The following tables list the product-specific information that you will need when you install or start the network configuration editor.

For example, if you want to use the 8371 Networking Multilayer Ethernet Switch on the AIX platform, download the file shown in the *aixname* column of Table 2 (*ibmnce10.lpp*) from the IBM Networking page and follow the instructions at “Installing on AIX” on page 3.

When you are ready to start the program, you need to change to the directory shown in the *aixpath* column of the same table (*/usr/lpp/ibmnce*) and follow the instructions at “Starting the Program on AIX” on page 5.

Table 2. AIX Product Information

Product Name	aixpath	aixname
8371 Release 2.0	/usr/lpp/ibmnce	ibmnce10.lpp

Table 3 on page 3 lists Windows-specific information. Use it if you want to use the program on the Windows platform.

Table 3. Windows Product Information

Product Name	winfolder	winprogram	winname
8371 Release 2.0	IBM NCE	IBM NCE	ibmnce10.exe

Table 4 lists OS/2-specific information. Use it if you want to use the program on the OS/2 platform.

Table 4. OS/2 Product Information

Product Name	os2folder	os2program	os2name
8371 Release 2.0	IBM NCE	IBM NCE	ibmnce10.cmd

Installing the Program

Once you have either obtained the appropriate CD-ROM or downloaded the appropriate file from the IBM Networking home page, the procedure for installing the network configuration editor depends on the operating system under which the program will run. “Installing on AIX”, “Installing on Windows” on page 4, and “Installing on OS/2” on page 4 describe the steps you take to install the program, based on your operating system.

Before installing the network configuration editor, make sure that you have write access to the directory in which you are installing it. The installation process writes program files to the directory during installation and the network configuration editor writes information to the directory while it is running. See the appropriate README file for more detailed information.

Installing on AIX

1. Ensure that you are logged on as the “root” user. (The **whoami** command will show root at the command prompt.) If not, enter **su root**.
2. If you are installing from the CD-ROM, create directory **/cdrom** if it does not exist, insert the appropriate network configuration editor CD-ROM into the CD-ROM drive, and mount the CD-ROM:

   ```
   mount -v cdrfs -r' '  /dev/cd0 /cdrom
   ```
3. Change to the directory where the INSTALLP file (*aixname*—see Table 2 on page 2 for the specific name) is located. On the CD-ROM it will be in the **/cdrom/config/aix** directory. If you downloaded the *aixname* file from the IBM Networking home page, change to the directory where the file exists.
4. Run SMIT and do the AIX 4.2 procedure as follows:

   AIX 4.2 Procedure:

   a. Select **Software Installation and Maintenance**.

   b. Select **Install and Update Software**.

   c. Select **Install and Update from Latest Available Software**.

   d. Select **Input device/directory for software.**.

   e. Enter the fully qualified file name of the INSTALLP file (*aixname*—see Table 2 on page 2 for the specific name) from the CD-ROM **/cdrom/config/aix** directory.

f. Press **PF4** to list the software and select the installation program.

g. Select **OK**.

h. Select **OK**.

i. Select **Done**.

j. Select **Cancel**.

5. If installing from the CD-ROM, unmount the CD-ROM:

   ```
   unmount /cdrom
   ```

6. Enter **exit** to end the root user session.

Installing on Windows

1. If installing from the CD-ROM, insert the appropriate CD-ROM into the CD-ROM drive. If autoread is on, the installation program starts and displays an installation dialog. Otherwise, change to the **/config/win** directory and run setup on the CD-ROM.
2. If installing after downloading the Windows installation file from the IBM Networking page, change to the directory into which you downloaded the file and run the *winname* program (see Table 3 on page 3 for the specific name) to install the product.

Installing on OS/2

1. Open an OS/2 command window.
2. If installing from the CD-ROM, change to the **/config** directory on the CD-ROM.
3. If installing after downloading the OS/2 installation command from the IBM Networking page, change to the directory into which the file was downloaded.
4. Enter **os2name** (see Table 3 on page 3 for the specific name) and follow the prompts that display on the panel to complete the installation.

The program creates a folder that contains the Network Configuration Editor icon. There will also be an icon for the README files in the folder.

Specifying the Configuration Repository Location

Configurations are stored in a directory specified in the config.ini file. The default directory is .\cfgdb. You may specify a different directory by adding the following lines to the config.ini file:

```
[CCE]
cfgrepositorypath=<yourdirectoryname>
```

You must have write access to the specified directory. The program does not support multi-user access to the configuration repository.

Chapter 2. Starting the Program

This chapter describes the procedures for starting the network configuration editor and how to upgrade configuration files created by earlier versions of the program. It consists of the following sections:

- “Starting the Program on AIX”
- “Starting the Program with AIX NetView/6000”
- “Starting the Program on Windows” on page 6
- “Starting the Program on OS/2” on page 6
- “After the Program Starts” on page 6
- “Migrating a Configuration to the Latest Release” on page 7

Starting the Program on AIX

Before starting the network configuration editor, X-Windows and the common desktop environment must be running on the workstation.

To start the network configuration editor:

1. Change directories to the directory that contains the network configuration editor, for example:

   cd /usr/lpp/*aixpath* (see Table 2 on page 2 for the specific path)
2. Enter **config &** to start the program in the background.

Starting the Program with AIX NetView/6000

If you have NetView/6000 you can start the network configuration editor directly from the menu bar. If the network contains devices of different releases, you can start any network configuration editor installed on the machine directly from the menu bar.

To use multiple network configuration editors, make sure that each network configuration editor is in a separate subdirectory from the other network configuration editors and NetView/6000.

For each network configuration editor that you want to start from the menu bar:

1. Create a file in the /usr/OV/registration/C subdirectory that contains the code for the menu bar selection item.
2. Change the command statement in the file to point to the path of the desired network configuration editor.
3. Save the file with a name that indicates the type, version, and release of the network configuration editor you are starting.

Figure 1 on page 6 is an example of the code contained in a file named NCER10 that points to the network configuration editor in directory `/usr/lpp/ibmnce`. The **_C** in the line under MenuBar "Tools" represents the accelerator character for the menu item—the underscored character in the Navigation Window menu bar’s Configure option. This character must be in the character string directly to the left of the action, and cannot be duplicated in another menu item.

```
/*
**  Global Tools
*/
Application "8371 R2 Configuration"
    Description {
         "8371 R2 Configuration"
    }
    /*
    **  Tools
    */
    MenuBar "Tools"
    {
       "IBM Network Configuration Editor" _C f.action "cfgtool";
    }
    Action "cfgtool"
    {
        Command "/usr/lpp/ibmnce/config";
    }
}
```

Figure 1. Netview/6000 Menu File NCER10

For more information, such as how to create a sub-menu that contains all of the network configuration editor menu items, refer to the *AIX SystemView NetView/6000 V2R1 Programmer's Guide,* SC31-7022.

Starting the Program on Windows

To start the network configuration editor under Windows, select:

Start | **Programs** | *winfolder* | *winprogram*

(See Table 3 on page 3 for the specific folder name and program name.)

Starting the Program on OS/2

To start the network configuration editor under OS/2:

1. Locate and open the network configuration editor *os2folder* folder (see Table 4 on page 3 for the specific name).
2. Double-click the network configuration editor *os2program* icon (see Table 4 on page 3 for the specific name) to start the program. The title window for the network configuration editor displays.

After the Program Starts

When the program starts, it uses the options defined in the **cfg.ini** file. If the file does not exist in the network configuration editor's directory, the program uses default options and creates a new **cfg.ini** file. To change the options, select **Options** from the network configuration editor menu bar. See "Changing the Preferences" on page 11 for the options that you can change.

Migrating a Configuration to the Latest Release

To migrate an 8371 Release 1 configuration that resides in a network device for use by the network configuration editor:

1. Start the network configuration editor.
2. In the Network Configuration Editor window, select **Import** from the *Configurations* menu.
3. Select **8371, 8371r20**, and **SNMP**, and then click Parameters and specify the SNMP import parameters (IP address, community, and so forth) and click **OK**, then click **Import**.

   The program migrates the configuration during the import and creates a Release 2.0 configuration.
4. Upgrade the software in the network device to Release 2.0.
5. Select the Release 2.0 icon for the upgraded configuration, then select **Export** from the *Selected* menu and select **to device (by SNMP)** from the pop-up menu.
6. Complete the entries in the Exporting... window and the Export parameters in the SNMP Export Parameters window, and click **Export**. The upgraded configuration loads into the device.

Chapter 3. Using the Program

This chapter describes the network configuration editor's navigation features, explains how to use the program, and helps you to diagnose problems. It consists of the following sections:

The network configuration editor enables you to manage multiple 8371 configurations efficiently. Once you define a configuration to the program, it is kept in the program's repository. To make changes to a configuration contained in the repository, load the configuration into the program's memory. (As you work on a configuration, you may want to save it to the repository periodically.) If you are working on several configurations simultaneously and it becomes necessary to conserve memory, you may unload a configuration. To transfer a configuration between the program and a network device, use the Import and Export commands.

Before Using the Program

Before using the network configuration editor to create configurations for your network device please read the README file that is contained in the directory in which the program is installed or on the CD-ROM. The README file contains the most current directions for using the network configuration editor.

Using the Program's Graphical Interface

When you first start the network configuration editor, it displays the Copyright Notice window and the Network Configuration Editor window (similar to that shown in Figure 2 on page 10). Select **OK** in the Copyright Notice window to close it. The Network Configuration Editor window's *Configurations* and *Selected* menus enable you to select high-level tasks. Once you select a task, the Navigation and Configuration windows open. The Navigation window provides a choice of configuration categories and the Configuration window enables you to define specific configuration parameters. See the descriptions of these windows for further details.

"Setting Up the Network Device for SNMP" on page 18 shows the typical steps to configure a new network device or to change an existing configuration. "Setting Up the Program" on page 18 shows the typical steps for setting up the network configuration editor to communicate with a network device.

Navigating Through the Program

The network configuration editor features a point-and-click interface. You select a configuration from the Network Configuration Editor window, select a category from the Navigation window, and then fill in fields in the related pages in the Configuration window. The items in the Navigation window are arranged generally in logical top-to-bottom order so that you move down the list as you progress through the configuration tasks, and in left-to-right order as you specify detailed information. The network configuration editor provides *key combinations* to minimize the number of keystrokes needed to do tasks. To display field-specific help information for panels in the Configuration window, select the field and then press **F1**.

Using the Network Configuration Editor Window

The Network Configuration Editor window is shown in Figure 2. It contains:

- A title bar
- A menu bar
- An area containing a labeled icon for each configuration contained in the network configuration editor repository. If you load a configuration into the network configuration editor's memory by double-clicking on its icon, the program indicates that it is loaded by prefixing the configuration name under the icon with an asterisk.

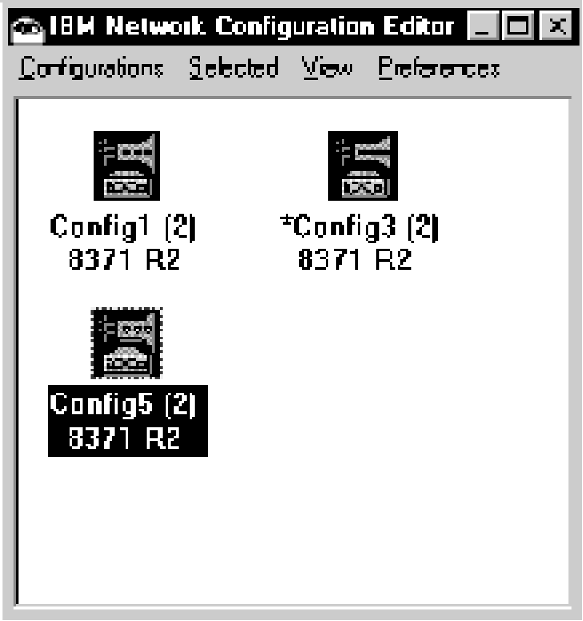


Figure 2. The Network Configuration Editor Window

The Network Configuration Editor window's menu bar contains the following options:

Configurations
Selected
View
Preferences
Help

In the Network Configuration Editor window you may load a configuration from the program's repository into its memory either by double-clicking its icon or by selecting it and selecting **Open Configuration** from the *Selected* menu. You may use the View option to display detailed information about the defined configurations or to display information about them in hierarchical order.

Changing the Preferences

Use the Preferences menu to change:

- Colors
- Guided configuration
- Message prompting
- ATM address format

Using Guided Configuration

The network configuration editor contains a guided configuration facility to help you configure a network device's features and protocols. Guided configuration consists of elementary procedures that step you through the configuration activity for a feature or protocol. Continuous guided configuration assistance is active by default. To toggle it off or on, display the Network Configuration Editor window and select Guided configuration from the *Preferences* menu. If it is turned off and you want to view guided configuration text in a help window without making it active continuously, press **F1**.

Notes:

1. There may be multiple ways to configure a feature or protocol. Guided configuration is the recommended procedure.
2. Guided configuration does not keep track of where you are in a procedure. If you follow a link to other help information from within a procedure, then when you return to the procedure the help window displays information for the beginning of that procedure.

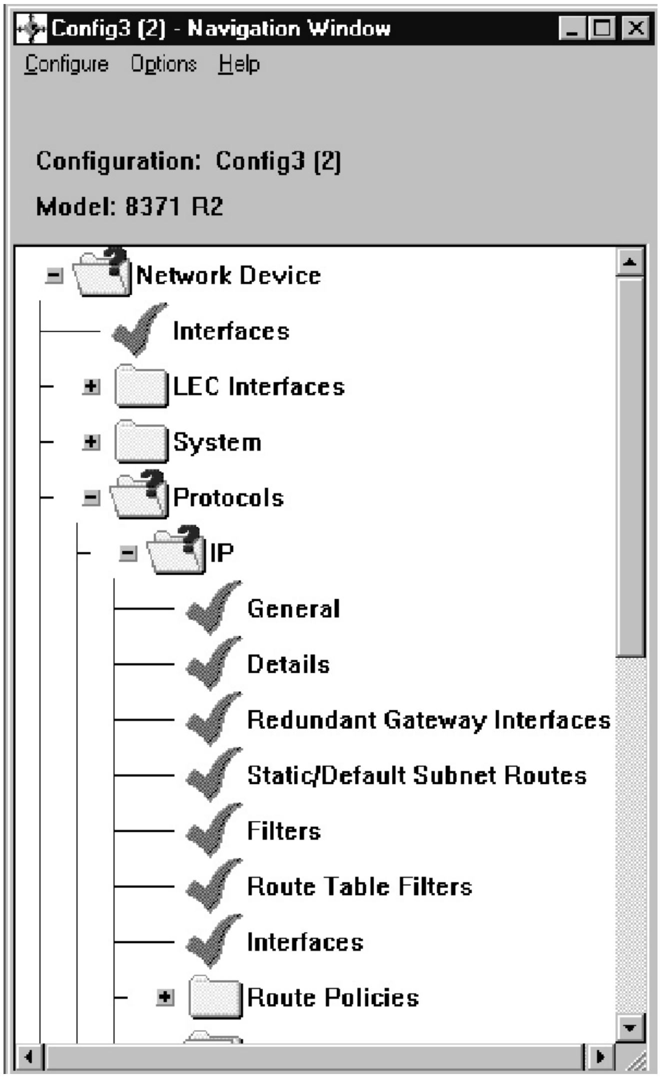
To create a new configuration, select **New** from the *Configurations* menu. To change an existing configuration, select a configuration icon, then select **Open Configuration** from the *Selected* menu. The network configuration editor then displays the Navigation and Configuration windows, similar to those shown in Figure 3 on page 12 and Figure 5 on page 15 respectively.

Using the Navigation Window

The Navigation window is shown in Figure 3 on page 12. It contains:

- A title bar
- A menu bar
- A configuration description area
- A scrollable list of folders (indicated by folder icons), which contain features and protocols that you can configure. The folders and their contents are arranged logically so that you perform configuration activities by starting at the top of the window and working toward the bottom.

If you select a category in the window, the Configuration window's contents change, enabling you to enter parameters related to that category. When you have finished entering those parameters, select a new category from the Navigation window.



Configuration: Config3 (2)

Model: 8371 R2

Figure 3. The Initial Navigation Window

If you select any item in the Navigation window and Guided Configuration is active (the default setting), the program displays appropriate configuration assistance information as you select new items. See “Using Guided Configuration” on page 11 for further information.

If an error occurs or you have not supplied required information in the current configuration, a question mark (?) appears next to the related item. If the configuration information is correct, a check mark (✓) appears next to the item. For an example of these indicators, see Figure 4 on page 14.

The Navigation window’s menu bar contains the following options:

Configure
Help

The area below the menu bar is the configuration description area. It displays the following information:

- The name of the currently selected configuration. The default name is **config**.

- The network device model and, if applicable, adapter type you are configuring (the default is set in the Options menu)

In the Navigation window, you may display a folder's contents either by clicking the plus (+) icon next to the folder or by double-clicking the folder itself. You may also choose an action to perform by selecting a folder and then pressing mouse button 2 (middle mouse button, right mouse button on a 2-button mouse) on the folder. This displays a pop-up menu that allows you to:

- Fully expand the items in the Navigation window.
- Expand the items in a folder.
- Display a history menu of configuration items you have selected. From the history menu you can access any of the previously selected configuration items. For OS/2, Windows 95, Windows NT, and AIX, press and hold mouse button 2 (middle mouse button, right mouse button on a 2-button mouse) until you select the item from the submenu displayed by the history menu item. (You do not have to hold mouse button 2 for Windows 95.) You can invoke the history menu from any item just as you would the pop-up menu.
- Validate a selection item in error. The Navigation window indicates erroneous items by displaying a large yellow question mark next to the item. The validate function displays messages for any configuration errors or required fields that do not have correct values based on the item selected. Using the validate function on a folder displays validation messages for all of the items within that folder.

  If you correct an error or fill in a required field and then leave the panel that contained the error, the program removes the question mark.

Using the Configure Menu

The Navigation window's *Configure* menu enables you to access the network configuration editor's features. The following list describes the menu choices and their usage. The underscored character and the text in parentheses indicate the key combination (shortcut) for the choice.

Save configuration (Alt+S)

Save the current configuration into the program's repository.

Create device configuration (Alt+C)

Export a binary configuration file (**.cfg**) that you may TFTP to the network device.

Communications (Alt+M)

- Send or retrieve a configuration to or from a single network device, restart a single device, or query a single device's information.
- Send or retrieve configurations to or from multiple devices or restart multiple devices.

  Note: Before using the communication features, you must establish communications between the network configuration editor and the device. See “Establishing Communications Between the Program and a Network Device” on page 18 for details.

Create ASCII configuration (Alt+A)

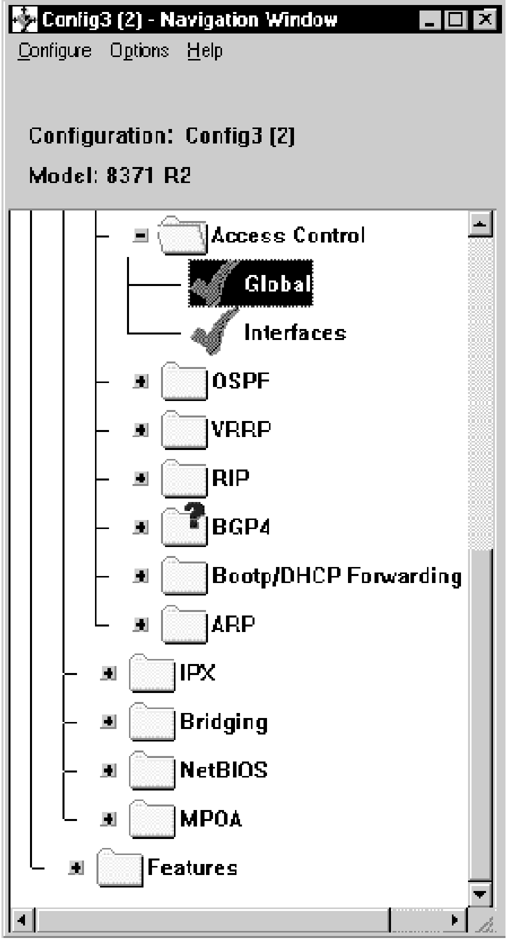
Export an ASCII version of a configuration. You can use this configuration as documentation of a device's configuration.

You can also read in an existing ASCII configuration file to load into a device.

Close (Alt+L) Close the configuration that you are working on.

Using the Configuration Window

To display the Configuration window, select a configuration category in the Navigation window (shown in Figure 4).

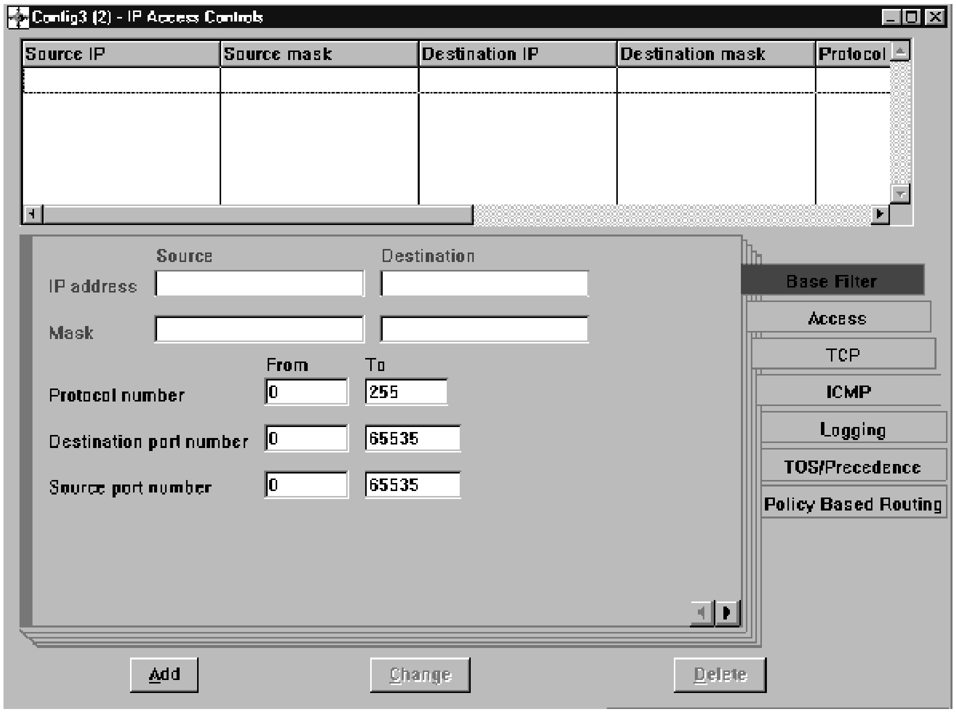


Configuration: Config3 (2)

Model: 8371 R2

Figure 4. Typical Navigation Window

The program displays related configuration information in the Configuration window (shown in Figure 5 on page 15) with the name of the selected item on its title bar. The display can have the appearance of a single page of configurable items or of a notebook with several pages of configurable items. The contents and format of the Configuration window depend upon the item selected in the Navigation window. You enter configuration parameters in this window.



	Source	Destination
IP address		
Mask		

	From	To
Protocol number	0	255
Destination port number	0	65535
Source port number	0	65535

Figure 5. The Configuration Window

To validate information that you have entered on a notebook page, right-click the page. The program examines the fields and if it detects an error, it issues an error message. If, at any time, the program detects a problem with a required field, it highlights that page's tab in color (as indicated by the Base Filter tab in Figure 5). Additionally, you can validate a configuration at any time by right-clicking in the Navigation window and then selecting the Validate option from the pop-up menu. The program displays a list of any errors that exist, which you can correct through the normal process of selecting items in the Navigation window and entering new information in the Configuration window.

Selecting Multiple Items in a List

If the network configuration editor displays a selection list, you can pick multiple items from those lists. The method for selecting items depends on whether you want to start at one point and end at another point or just choose items at random from the list.

To select multiple individual items:
Press and hold the **Ctrl** key, point to each item, and then press mouse button 1.

To select a range of items:
Select the first item in the range. Press and hold the **Shift** key. Select the last item in the range and then press mouse button 1.

To deselect items:
Press and hold the **Ctrl** key, point to the item you want to deselect and press mouse button 1. If you do not hold the **Ctrl** key, you will deselect all previously selected items except the item just selected.

Using the Mouse

The mouse is the primary means for moving through the various windows in the network configuration editor.

The network configuration editor supports the following buttons on a 3-button mouse. (The left and middle buttons on a 3-button mouse are mapped to the left and right mouse buttons on a 2-button mouse.)

- Left (button 1) - to select an item in a panel
- Middle (button 2) - to perform the following actions:
  - In the Navigation window
    - Display a pop-up menu for a selected item in the scrollable display.
    - Move through and select items in a menu.
    - Validate a selection item in error. The Navigation window indicates erroneous items with a yellow question mark next to the item. The validate function displays a message for each required field that does not have a valid value and for any configuration errors.
  - In the Configuration window
    - Drag-and-drop items. The program indicates that drag-and-drop is active by changing the cursor into a hand icon. Use this function to reorder filters, dump entries, and other configuration lists. For example, you can drag and drop items in a list of IP addresses to reorder them and establish a new default gateway.

      Note: Some lists do not allow you to reorder items in them.
    - Validate the data you have entered on a page.
- Right (button 3) is not used currently

Using the Keyboard

Some keys and key-combinations have specific functions. Table 5 shows the key assignments.

Table 5. Keyboard Functions

Keys	Function	How to Use
F1 (Help)	Request help (see note).	**To obtain help for configuring a feature or protocol:** See “Using Guided Configuration” on page 11. **To get help on a panel:** 1. Select an item in the Navigation window, then press **F1**. **To get help for a parameter, a button, or a drop-down list:** 1. Select the parameter field or button by tabbing to the field, button, or list, then press **F1**. **Note:** If you tab to a field or button, a colored box highlights the item and you can then request help. Clicking a button causes the program to do the task indicated by the button or list. In the help text, additional information is hypertext-linked to the information on a particular panel. By default, this link is highlighted in blue. If you press F1 on a panel or a parameter without closing a previously opened help window, the new help text overlays the currently displayed text.
Up/Down Arrow Keys	Move vertically.	Use these keys to move through lists on any panel, including the Navigation window or items in any of its menu bar selection menus. To select an item, press the Space bar.
Left/Right Arrow Keys	Move Horizontally	Use these keys to move through both menu items (after you have selected one) and their drop-down lists in the Navigation window. Also use them to move through parameter fields in the Configuration window without affecting the existing values. To select an item, press the Space bar.
Tab and Back-tab	Next or previous entry field.	These keys work only in the Configuration window. The keys move you to the various entry items in the Configuration windows. If there are items in a scrollable list, such as in the Device Interfaces panel, you will move to items that are below the scrolling window.
Alt+C	Configure menu	Displays the **Configure** menu so you can select items from it.
Alt+P	Options menu	Displays the **Options** menu so you can select items from it.

Table 5. Keyboard Functions (continued)

Keys	Function	How to Use
Alt+H	Help menu	Displays the **Help** menu so you can select items from it.
Esc	Escape	Clears the menus.

Establishing Communications Between the Program and a Network Device

You must set up both the network device and the network configuration editor before they can communicate with one another. If you do not have a configuration loaded into the device, you must perform the steps in both “Setting Up the Network Device for SNMP” and “Setting Up the Program”. If you already have a configuration loaded into the device, you need perform only the steps described in “Setting Up the Program”.

Setting Up the Network Device for SNMP

This procedure describes how to set up the network device to communicate with the network configuration editor. If the device does not have a configuration loaded, you must do the following procedure to prepare it, then do the procedure at “Setting Up the Program”. After that you can export and load a configuration to the device. A device does not have a configuration loaded if any of the following conditions are true:

- The device is new and contains no configuration
- You have cleared the device’s configuration
- The device encountered a problem that cleared the configuration

1. Establish access to the device. You can do this either locally through an ASCII terminal or remotely using Telnet, as described in your device’s setup guide. To use Telnet, you must at least configure the IP address for the device by doing the following steps.
2. Enter the Talk 6 **protocol ip** command.
3. Specify the IP address with the Talk 6 **add address** command.
4. Specify the protocol with the Talk 6 **protocol snmp** command.
5. Define a community with the Talk 6 **add community** command.
6. Specify the write_trap option with the Talk 6 **set community access write_read_trap** command.
7. Enable the trap with the Talk 6 **enable trap all** command.
8. Restart or reload the device.

When the device completes the restart sequence, it will be able to communicate with the network configuration editor.

Setting Up the Program

The easiest way to set up the network configuration editor is to first configure the IP address and SNMP community for the network device and load them into the device using the procedure at “Setting Up the Network Device for SNMP”, then import the configuration into the program by selecting **Import** from the Network Configuration Editor window’s Configurations menu. The alternative is to use the

New selection from the Network Configuration Editor window’s *Configurations* menu and enter the configuration options manually as shown in the following procedure.

To manually set up the network configuration editor, do the following:

1. Select **New**, and the appropriate device and release options from the Network Configuration Editor window’s *Configurations* menu.
2. Select the appropriate model from the *New Configuration* menu.
3. Select **Interfaces** in the IP folder.
4. In the Configuration window:
   a. Select **IP Addresses** next to the LAN interface with which the network configuration editor will communicate.
   b. Enter the IP address and Subnet Mask for this interface.

      Note: These addresses must match the addresses that are configured on the device’s interface.
   c. Select **Add**
5. Select **Communities** under the SNMP folder in the Navigation window.
6. In the Configuration window:
   a. Enter the name of the SNMP community in the Name field on the General page. Use the SNMP community defined in step 5 on page 18 at “Setting Up the Network Device for SNMP” on page 18.
   b. Select **Read-write trap** in the drop-down list under Access type.
   c. Select **Add**

      Note: The name and the access type must match the name and access type that is configured on the device with which the network configuration editor will communicate.

The network configuration editor can now communicate with the network device and perform functions such as importing and exporting configurations.

Working with Configurations

The network configuration editor supports configurations in the following formats:

- Network Device-Native (.cfg) File)
- Configuration Storage Format (.csf) File
- ASCII File

Using Network Device-Native (.cfg) Files

Network Device-Native files reside in the devices themselves or on the program’s hard drive (as **.cfg** files). They contain configurations for network devices. You may import such a file into the program, change it, and either export it to the same (or a different) device or save it on the network configuration editor’s hard drive for backup or transportability purposes.

Using Configuration Storage Format (.csf) Files

Configuration storage format files reside on the network configuration editor’s hard drive (as **.csf** files). They contain configurations for network devices. You may import such a file into the program, change it, and export it to any of the supported formats, including exporting it to a network device and loading it into the device's SRAM so that it becomes the current configuration. You may also use these files for backup or transportability purposes.

Using ASCII Files

ASCII format files reside on the program's hard drive and contain configurations for network devices. ASCII files enable you to alter many configurations at the same time without loading each configuration into the graphical user interface. For example, if you want to update many IP addresses throughout a network, it may be faster to save the affected configurations in an ASCII file and then manually (or automatically with a program) update the IP addresses. An ASCII file is also useful if you want to scan the configuration online or print it for review.

After changing a configuration in an ASCII file, you can import the file into the network configuration editor and export it to either a network device or any of the supported file formats.

Adding Comments to an ASCII File

You may add comments to an ASCII file to help you to use the ASCII file as a template for other configurations. The comments can include:

- A brief description of the parameter
- The type of value
- The valid values or value ranges, if applicable

Using an ASCII File as a Configuration Template

The ability to import and export ASCII files can help you automate the process of creating or updating a large number of device configurations. The following procedure explains how to perform a mass update, using an ASCII file as a template.

1. Create a file containing the new information for the devices you are configuring. You can use a full or partial template. Make sure that any interfaces or adapters you specify match the models and adapters specified in the header.
2. Create a copy of the template file, update the new copy with the device information, and save the file with a name that will help you to identify the device for later transmittal.
3. Locate a device in your network that contains a configuration that closely matches that of the target devices. Ideally, the only things you should need to change are items like IP addresses, MAC addresses, dial circuit numbers, and so forth. The devices should support and use the same filters and protocols and have the same interfaces.
4. Import the device configuration from the device you located in step 3. You may import the configuration with the command line facility of the network configuration editor.
5. Export an ASCII file of the configuration.
6. Create the new configurations using the file you created in step 1 and the ASCII file you created in step 5.

After you create the new configurations, you can use the network configuration editor to import the resulting ASCII files, save them as binary files, if needed, and then export the files to the appropriate devices. "Importing Multiple ASCII Files" on page 38 contains more information about how the command line facility can help you automate this task.

Interpreting an ASCII File

Figure 6 on page 22 contains a sample of a small portion of an ASCII file with comments. When interpreting the ASCII file, note that:

- Header information begins with an exclamation mark (!).
- Comments begin with an asterisk (*) in column 1. The comments contain descriptions of the parameters and the valid values and ranges. When necessary, the comments indicate the type of the parameters. The types are:

  Multiple Segments Used for data composed of well-defined parts. For example, a four-segment numeric address that is used to define IP addresses and IP address masks.

  Numeric Typically used for data that may have mathematical calculations performed on it. Data value ranges are enforced on all numeric data types.

  String Each string must be within a certain length and can be composed only of a specific set of characters.

  Enumerated Used for multiple-choice values. When an ASCII file is saved with comments, the comments explain the possible values. For example:

  Enable/Disable 1 = enable, 2 = disable. Used mainly for enabling/disabling adapter ports and protocols.

- Configuration groups and parameters begin with a pound sign (#).
- A group of items is contained within { and }.
- Elements in a list of items are contained within { and }.
- The word ″nil″ signifies that the parameter currently has no value.
- All parameters appear in the file even if you did not configure them. When you did not explicitly specify the value, the program uses the default value. If a default value for the parameter does not exist, the program uses “nil”.
- Encrypted values are preceded by an “E.” To specify a new value to be encrypted by the network configuration editor, change the “E” to a “U” and overwrite the old value with the new value. When the network configuration editor reads in the ASCII file, it will encrypt the new value.

```
8371 Configuration
06/14/1999 at 10:34:16 AM
  !  'Title:' = 'ACXE0000   8371 Configuration '
  !  'Output On:' = '06/14/1999 at 10:34:16 AM'
  !  'Configuration:' = 'Config5 (2)'
  !  'Description:' = '-- Configuration 5 --'
  !  'Device/Release:' = '8371 R2'
  !  'Version:' = '61000'
  !  'Model:' = 'model 8210'
  !  'Created:' = '06/14/1999 at 10:31:35 AM'
  !  'Last Update:' = '06/14/1999 at 10:31:35 AM'


*  slots
*      a list composed of 1 to 2 of the following entry:
*    adaptor = {
*      adaptorType = {
*        slotType : Slot type
*         value:  one of the following strings:
*                   emptySlot
*                   atm
*                   fddi
*      }
*      ports
*          a list composed of 0 to 8 of the following entry:
*        port = {
*          interfaceNumber : Interface number
*           value:  range from 0 to 65535
*          portNumber : Port number
*           value:  range from 0 to 65535
*        }
*    }
#'slots' = {  }

#'appletalk' = {
*    at2General = {
*      at2Enable : Enable Apple Talk 2
*       value:  either enable or disable
*      at2Checksum : Checksum
*       value:  either enable or disable
*      at2Translation : Translate
*       value:  either enable or disable
*      at2FastPath : Fast path cash size
*       value:  range from 100 to 10000
*      at2FastPathEnable : Enable fast path
*       value:  either enable or disable
*    }
  #'at2General' = {
    #'at2Enable' = 'disable'
    #'at2Checksum' = 'disable'
    #'at2Translation' = 'disable'
    #'at2FastPath' = '500'
    #'at2FastPathEnable' = 'enable'
  }
*    at2Interfaces
*        a list composed of 0 or more of the following entry:
*      at2Interface = {
*        at2InterfaceGeneral = {
*          at2FirstNetworkRange : Lowest number of network range
*           value:  range from 1 to 65279
```

Figure 6. Sample ASCII Configuration with Comments (Part 1 of 3)

```
*          at2LastNetworkRange : Highest number of network range
*           value:  range from 1 to 65279
*          at2NodeNumber : Node number
*           value:  range from 0 to 254
*          at2DefaultZoneName : Default zone name
*           value:  a string
*                     no more than 34 character(s) in length
*                     MsgWithAnyAsciiCharacter
*          at2InNFilter : Input NFilter
*           value:  one of the following strings:
*                     None
*                     Inclusive
*                     Exclusive
*          at2OutNFilter : Output NFilter
*           value:  one of the following strings:
*                     None
*                     Inclusive
*                     Exclusive
*          at2InZFilter : Input ZFilter
*           value:  one of the following strings:
*                     None
*                     Inclusive
*                     Exclusive
*          at2OutZFilter : Output ZFilter
*           value:  one of the following strings:
*                     None
*                     Inclusive
*                     Exclusive
*          at2EnableInterface : Enable interface
*           value:  either enable or disable
*          at2InputNetworkFilter : Input NFilter
*           value:  one of the following strings:
*                     None
*                     Inclusive
*                     Exclusive
*          at2OutputNetworkFilter : Output NFilter
*           value:  one of the following strings:
*                     None
*                     Inclusive
*                     Exclusive
*          at2InputZoneFilter : Input ZFilter
*           value:  one of the following strings:
*                     None
*                     Inclusive
*                     Exclusive
*          at2OutputZoneFilter : Output ZFilter
*           value:  one of the following strings:
*                     None
*                     Inclusive
*                     Exclusive
*          at2SplitHorizon : Split horizon
*           value:  either enable or disable
*        }
```

Figure 6. Sample ASCII Configuration with Comments (Part 2 of 3)

```
*        at2ZoneNames
*            a list composed of 0 or more of the following entry:
*          at2ZoneName = {
*            at2ZName : Zone name
*             value:  a string
*                       no less than 1 character(s) in length and
*                       no more than 32 character(s) in length
*                       MsgWithAnyAsciiCharacter
*          }
*        at2NFilters
*            a list composed of 0 or more of the following entry:
*          at2NFilter = {
*            at2FirstFilterNetworkRange : First network range number
*             value:  range from 1 to 65279
*            at2LastFilterNetworkRange : Last network range number
*             value:  range from 1 to 65279
*            at2NetworkFilterType : Filter type
*             value:  one of the following strings:
*                       In
*                       Out
*          }
*        at2ZFilters
*            a list composed of 0 or more of the following entry:
*          at2ZFilter = {
*            at2FilterZoneName : Zone name
*             value:  a string
*                       no less than 1 character(s) in length and
*                       no more than 30 character(s) in length
*                       MsgWithAnyAsciiCharacter
*            at2ZoneFilterType : Filter type
*             value:  one of the following strings:
*                       In
*                       Out
*          }
*      }
  #'at2Interfaces' = {
    #'at2Interface' = {
      #'at2InterfaceGeneral' = {
        #'at2FirstNetworkRange'
        #'at2LastNetworkRange'
        #'at2NodeNumber' = '0'
        #'at2DefaultZoneName'
        #'at2InNFilter' = 'None'
        #'at2OutNFilter' = 'None'
        #'at2InZFilter' = 'None'
        #'at2OutZFilter' = 'None'
        #'at2EnableInterface' = 'disable'
        #'at2InputNetworkFilter' = 'None'
        #'at2OutputNetworkFilter' = 'None'
        #'at2InputZoneFilter' = 'None'
        #'at2OutputZoneFilter' = 'None'
        #'at2SplitHorizon' = 'enable'
      }
      #'at2ZoneNames' = {  }
      #'at2NFilters' = {  }
      #'at2ZFilters' = {  }
    }
⋮
```

Figure 6. Sample ASCII Configuration with Comments (Part 3 of 3)

Understanding How the Program Validates ASCII Files

The network configuration editor performs the following actions while reading the ASCII files:

- If an adapter is specified in the header, the program compares that value with the adapter type in the Devices group of the configuration file.
- If a model is specified in the header, the program:
  1. Checks to see that the defined interfaces match the model capabilities. If they do not, the program stops reading the configuration and logs an error.
  2. Verifies that the number of interfaces is correct for the device configuration identified in the header.

     If too many interfaces are defined, the program stops reading the configuration and logs an error.

     If too few interfaces are defined, the program uses defaults for the undefined interfaces. For example, if the device has two WAN ports and you supply values for one, the program uses defaults for the missing interface after the configured interface.

     Note: Undefined interfaces must be the last interfaces in the interface list. For example, if you define interfaces 0, 2, and 3, the network configuration editor does not use defaults for the missing interface. In this case, you must define the consecutive interfaces 0, 1, 2, and 3 for the program to provide defaults for the missing interface. For interface 1, you ***must*** provide a default interface.
- Compares the number of device interfaces to the number of elements in the interface protocol list.

  If fewer elements are in the interface protocol list than there are device interfaces specified, the program fills the list with default protocols for the remaining interfaces.

  If more elements are in the interface protocol list than there are device interfaces specified, the program stops reading the configuration and logs an error.

Creating a New Configuration

To configure a network device using the network configuration editor:

1. Select **Configurations** from the Network Configuration Editor window menu bar.
2. Select **New** from the *Configurations* menu.
3. Select **8371** and **8371r20** from the pop-up menus.
4. Complete the entries in the New Configuration window and click **OK**. The program displays the Navigation and Configuration windows.
5. Select **Configure** from the Navigation window.
6. Select the model you want to configure from the New configuration submenu.
7. Select **Adapter Interfaces** from the Devices folder and configure any adapters for the device in their appropriate slots.
8. Select **Interfaces** from the Devices folder and configure the interfaces for the device.
9. Expand the General folder under the Devices folder, if it is not already expanded.
10. Select the items in the General folder and configure the parameters in each of those items.

11. Select the items in the System folder and configure the parameters in each of those items.
12. Select and configure the remaining folders as necessary.
13. Export the configuration to the device.

Changing an Existing Configuration

Note: While using the network configuration editor to change an existing configuration, make sure that no one is updating the configuration with the device command line interface. Any configuration data entered from the command line is lost if you send the configuration to the device from the network configuration editor.

To change an existing configuration:

1. Select a configuration icon in the Network Configuration Editor window, then select **Open Configuration** from the *Selected* menu and choose a view option.
2. The program opens the configuration.
3. Use the network configuration editor to change the parameters.
4. Save the configuration.
5. Export the configuration to the device.

Using Import

You may import the following entities into the network configuration editor's repository for modification or exporting:

- A configuration from a network device (through SNMP)
- A network device-native (**.cfg**) file
- A configuration storage (**.csf**) file
- An ASCII file

Importing from a Network Device (through SNMP)

To import a configuration from a network device:

1. Select **Import** from the Network Configuration Editor window's *Configurations* menu and select **SNMP** from the drop-down list.
2. The program displays a dialog in which you specify the device from which the program imports the configuration.
3. Select **OK**.

After importing the configuration, you can:

- Export the configuration to another format using **Export**.
- Save the configuration using **-save config**.

Note: If you import a configuration from a network device into the repository and load it into memory, the network configuration editor keeps the current configuration in memory, but makes the imported configuration the current one.

Importing from Network Device-Native (.cfg) Format

To import a **.cfg** file:

1. Select **Import** from the Network Configuration Editor window's *Configurations* menu and select **Device Configuration** from the drop-down list.

2. The program displays a dialog in which you specify the path and file name from which the program imports the **.cfg** file.

   Note: For the **Import** function to read a **.cfg** file successfully, the file must either have been created with the **Export** function or have come from the device.
3. Select **OK**.

After importing the configuration, you can:
- Export the configuration to a device using **Export**.
- Save the configuration using **-save config**.

Note: If you import a network device-native (**.cfg**) file into the repository and load it into memory, the network configuration editor keeps the current configuration in memory, but makes the imported configuration the current one.

Importing from Configuration Storage Format (.csf)

To import a **.csf** file:
1. Select **Import** from the Network Configuration Editor window's *Configurations* menu and select **CSF** from the drop-down list.
2. The program displays a dialog in which you specify a path. Then the program lists the configuration names in the specified directory, from which you select a **.csf** file to import.

   Note: For the **Import** function to read a **.csf** file successfully, the file must have been created with the **Export** function.
3. Select **OK**.

After importing the configuration, you can:
- Export the configuration to a device using **Export**.
- Save the configuration using **-save config**.

Note: If you import a configuration storage format (**.csf**) file into the repository and load it into memory, the network configuration editor keeps the current configuration in memory, but makes the imported configuration the current one.

Importing from ASCII Format

See “Interpreting an ASCII File” on page 21 for a description of ASCII files that can help you create your own ASCII file.

To import an ASCII file:
1. Select **Import** from the Network Configuration Editor window's *Configurations* menu and select **ASCII** from the drop-down list.
2. The program displays a dialog in which you specify the path and file name from which the program imports the ASCII file.

   Note: For the **Import** function to read an ASCII file successfully, you must either:
   - Create the file with the **Export** function, or
   - Create an ASCII file with the same parameter structure that the **Import** function creates.

“Examples of Using the Command Line Facility With ASCII Files” on page 38 explains how to use the network configuration editor command line facility to process ASCII files.

3. If you want the network configuration editor to create an ASCII log file that lists errors in the configuration, select **Log validation errors**, which is the default. The program always validates configurations and displays errors so you do not need to log validation errors, but the feature is useful if:
   - You want to check the configuration and correct any problems before transmitting the binary file to the device.
   - You are importing ASCII files from multiple devices, by enabling you to detect validation errors when the read process completes.

   In this dialog you can prevent the network configuration editor from recording validation errors in a validation log file. By default, the validation file has a **.vlg** file extension.
4. Select **OK**.

After importing the configuration, you can:
- Export the configuration to a device using **Export**.
- Save the configuration using **-save config**.

Note: If you import an ASCII file into the repository and load it into memory, the network configuration editor keeps the current configuration in memory, but makes the imported configuration the current one.

Using Export

You may export the following entities from the network configuration editor’s repository:
- A configuration to a network device (through SNMP)
- A configuration (**.cfg**) file to the hard drive (for backup) or a floppy disk (for transporting)
- A configuration storage (**.csf**) file to the hard drive (for backup) or a floppy disk (for transporting)
- An ASCII file to the hard drive (for backup) or a floppy disk (for transporting)

If you have imported an entity in one format and are exporting it to another format, the network configuration editor does any conversion internally without any explicit action on your part. The following topics discuss additional information that you may find useful when doing specific export tasks.

Exporting to a Network Device (through SNMP)

To export a configuration to a network device, select it in the Network Configuration Editor window, then:
1. Select **Export** from the Network Configuration Editor window’s *Selected* menu and select **to device (by SNMP)** from the drop-down list and specify the parameters, for example, address, community, and so forth.
2. Select **Export**.

Exporting to Network Device-Native (.cfg) Format

To export a configuration to network device-native format, select it in the Network Configuration Editor window, then:

1. Select **Export** from the Network Configuration Editor window’s *Selected* menu and select **Device Configuration** from the drop-down list.
2. Select **Export**.

Exporting to Configuration Storage Format (.csf)

To export a configuration to csf format, select it in the Network Configuration Editor window, then:

1. Select **Export** from the Network Configuration Editor window’s *Selected* menu and select **CSF** from the drop-down list.
2. Select **Export**.

Exporting to ASCII Format

To export a configuration to ASCII format, select it in the Network Configuration Editor window, then:

1. Select **Export** from the Network Configuration Editor window’s *Selected* menu and select **ASCII** from the drop-down list.
2. The program displays a dialog in which you specify the path and file name to which the program exports the ASCII file.
3. Select **Export**.

The network configuration editor exports an ASCII file containing the configuration parameters and their current values. See “Interpreting an ASCII File” on page 21 for a description of ASCII files that can help you create your own ASCII file.

Using the Command Line Facility

The Command Line facility enables you to automate network configuration editor operations. You can use the **config** command to enter a complete set of commands into an argument file named cfgargs, which the network configuration editor uses to direct its operation, or you can use it to enter commands on the command line. (See Figure 7 on page 30 for an example of a cfgargs file.) To use only the **config** command, without displaying the GUI interface, the last command in the file or on the command line must be **-exit**. If you do not meet this requirement, the program displays the GUI interface’s Navigation and Configuration windows.

Note: You can use the **config** command only in a windowed operating system.

CFGARGS File Example

If you enter the **config** command, the program looks for a file named cfgargs. Figure 7 on page 30 shows the contents of a cfgargs file that:

- Opens a modified ASCII configuration file.
- Sends the open configuration to a device.
- Causes the current configuration to become the active configuration by rebooting the device immediately.
- Saves the current configuration into the configuration repository.

```
-displayMessages off 1
-storeMessages on cfgMessages.log 2
-asciiRead deviceOne.acf withoutLogging 3
   –send 1.1.1.1 private 4
   –reboot 1.1.1.1 private 5
   –save * deviceTwo 6
-exit
```

Figure 7. Example of CFGARGS File Contents

1. Turns off message display. If any message offers a choice, takes the default choice.
2. Logs all messages to the log file.
3. Opens the specified ASCII file.
4. Sends the configuration to a device.
5. Causes the configuration to become the active configuration by rebooting the device immediately.
6. Stores the configuration into the directory that was specified by the previous **save** command, using the specified configuration name, deviceTwo.

Specifying the Config Command

You may enter the **config** command with or without additional commands and their arguments.

config
Causes the program to search for a cfgargs file in the network configuration editor directory and, if the file exists, performs the commands in it. When the program finishes processing the commands, it ends *and deletes the file*. If a cfgargs file does not exist, the program waits for additional commands on the command line.

Note: To facilitate making future configuration changes, you may want to prevent automatic deletion of the cfgargs file. You can do this by using a text editor to open the cfgargs file and keeping it open while the network configuration editor processes the commands in it. As an alternative, you may want to save a copy of the file under a different name.

config -*command* **[***args***] [...]**
If you specify additional commands and arguments on the command line, the program processes them. Each command must start with a minus sign (−) and be followed by as many additional arguments as needed.

If an argument, *args*, contains embedded blanks, you must enclose the argument in double quotes (“ ”).

Note: If you plan to enter commands on the command line, you must ensure that a cfgargs file does not already exist in the network configuration editor directory. This is necessary because if one does exist, the program processes only the commands in it and ignores the commands you enter on the command line.

The following list describes the **config** command-related commands and arguments.

Notes:

1. Commands and arguments must be separated from each other by either spaces, tabs, or new lines.
2. You can enter the commands in any case.
3. Any *filename* argument that does not contain the path before the physical file name uses the current directory as the path.

-asciiRead *filename* [**newName** *name*] [**withoutLogging**] **validateRequiredFields on|off**]

filename
Required argument. Specifies the name of an ASCII configuration file and reads it into the network configuration editor as the current configuration in memory. Note: For AIX, if the *filename* contains embedded blanks, you must enclose it in double quotes (“ ”).

newName *name*
Optional argument. If you provide this argument, the configuration that is read is saved with the specified name. Otherwise, it is saved with the current name. In either case, the configuration that is read becomes the current configuration.

withoutLogging
Optional argument. Specifies that validation errors should not be logged.

validateRequiredFields on|off
Optional argument. Specifies whether the program is to display an error indication if a required field does not contain a value. The default value is off.

Notes:

1. The ASCII file comments are provided for information only and they are ignored when present in an ASCII file specified by an **-asciiRead** command.
2. By default, the network configuration editor writes validation errors to an ASCII file, which has the same name as the file you read, but with a **.vlg** extension.

-asciiWrite *filename* [**configurationName** *name*] [**withComments**]

filename
Required argument. Specifies the name of the ASCII configuration file and creates the file from the network configuration editor’s current configuration in memory.

configurationName *name*
Optional argument. If you provide this argument, the named configuration is written.

withComments
Optional argument. Indicates whether the output file should contain extra commentary describing the ASCII configuration file.

Note: The ASCII file comments written by this command are provided for information only and are ignored if they are present in an ASCII file specified by an **-asciiRead** command.

-commandsFrom *filename*

filename
Required argument. Specifies the name of a file containing additional commands, and reads and processes those commands. The **-commandsFrom** command may be nested in a maximum of 10 files.

-createConfiguration *filename* [**configurationName** *name*]

filename
Required argument. Specifies the name of the file being created (in the binary format recognized by the device) from the program's current configuration in memory.

configurationName *name*
Optional argument. If you provide this argument, the named configuration is written.

-csfRead *directory config* [**newName** *name*]

directory
Required argument. Specifies the name of the directory containing the configuration you want to read.

config Required argument. Specifies the name of the configuration you want to read. The command stores the contents of the program's current configuration in memory. If the configuration name contains embedded blanks, you must enclose the name in double quotes (“ ”).

newName *name*
Optional argument. If you provide this argument, the configuration that is read is saved with the specified name. Otherwise, it is saved with the current name. In either case, the configuration that is read becomes the current configuration.

Note: This command is functionally identical to the **-open** command. However, it is preferred instead of **-open** because it can read configurations that exist in the old database (**.cdb**) format.

-csfWrite *directory config* [**configurationName** *name*]

directory
Required argument. Specifies the name of the directory to which you want to write the configuration.

config Required argument. Specifies the name of the configuration you want to write. The command writes the contents of the program's current configuration ifrom memory. If the configuration name contains embedded blanks, you must enclose the name in double quotes (" ").

configurationName *name*
Optional argument. If you provide this argument, the named configuration is written.

Note: This command is functionally identical to the **-save** command. However, it is preferred instead of **-save** because it can write configurations in the old database (**.cdb**) format.

-displayMessages **on | off**

on Enables the displaying of messages. Allows you to control the displaying of messages.

off Disables the displaying of messages. If a message would have offered you a choice of actions, the program takes the default action.

Notes:

1. The **-displayMessages** command can appear anywhere on the command line, in a cfgargs file, or in any of the **-commandsFrom** files.
2. You may provide multiple **-displayMessages** commands, but the network configuration editor only honors the last one.

-exit Closes the network configuration editor's GUI windows and ends the program. If you specify multiple commands, **-exit** must always be the last one. Any commands that follow it are not processed.

-exitOnError If an error occurs while processing the command line, ends the program without starting the GUI. The errors.log file or the file specified in the **-storeMessages** command identifies the source of the error.

-exitWhenDone
Closes the program after processing command line commands (whether or not command line errors occur) and ends the program. This command performs all the functions of, and is preferred over both **-exit** and **-exitOnError**.

-noLogo
Suppresses displaying the network configuration editor’s logo window during startup. Because the logo window requires you to select the OK button, specify this command to run the program in non interactive mode.

-open
directory config [**newName** *name*]

directory
Required argument. Specifies the name of the directory containing the configuration you want to open.

config
Required argument. Specifies the name of the configuration you want to open. The command stores the contents of the program’s current configuration in memory. If the configuration name contains embedded blanks, you must enclose the name in double quotes (“ ”).

newName *name*
Optional argument. If you provide this argument, the configuration that is opened is saved with the specified name. Otherwise, it is saved with the current name. In either case, the configuration that is opened becomes the current configuration.

Note: This command is functionally identical to the **-csfRead** command. However, **-csfRead** is preferred instead of **-open** because it can read configurations that exist in the old database (**.cdb**) format.

-queryinfo
host name **|** *ip address community* [*timeout*]

host name **|** *ip address*
Required argument. Specifies either the name of the device, which the program can resolve into an IP address, or the device’s actual IP address. The command retrieves a configuration summary from the device. The program displays and logs the summary based on the setting of the **-displayMessages** and **-storeMessages** commands, respectively.

community
Required argument. Specifies an SNMP community that is defined on the device with read access.

timeout
Optional argument. Specifies the time (in seconds) before the SNMP request expires. The default value is 10.

-read
filename [**newName** *name*] [**withoutLogging**]

filename

Required argument. Specifies the name of a configuration file and reads it into the network configuration editor as the current configuration in memory. Note: For AIX, if the *filename* contains embedded blanks, you must enclose it in double quotes (“ ”).

newName *name*

Optional argument. If you provide this argument, the configuration that is read is saved with the specified name. Otherwise, it is saved with the current name. In either case, the configuration that is read becomes the current configuration.

withoutLogging

Optional argument. Specifies that validation errors should not be logged.

-reboot *host name* **|** *ip address community* [*timeout*] [*date*] [*time*]

host name | ip address

Required argument. Specifies either the name of the device, which the program can resolve into an IP address, or the device’s actual IP address. The command sends a restart request to the device at the specified date and time, based on the date and time of the workstation on which the program is running. The program calculates the number of seconds between the date and time the program processes the command and the date and time specified in the command arguments.

community

Required argument. Specifies an SNMP community that is defined on the device with read access.

timeout

Optional argument unless the date or time arguments are specified, in which case it is required. Specifies the length of time (in seconds) before the SNMP request expires. The default value is 10.

date Optional argument unless the time argument is specified, in which case it is required. Specifies the date on which the device is to reboot. The default is the current date. The format is *dd/mm/yyyy* where *dd* is the 2-digit day of the month, *mm* is the 2-digit month of the year, and *yyyy* is the 4-digit year. You may enter the year as either two or four digits.

time Optional argument. Specifies the time at which the device will reboot. The default is the current time. The format for the time is HH:MM:SS (am or pm), or a time in 24-hour format, for example, 1800 (without the colons). The SS value, seconds, and am or pm fields are optional.

-retrieve *host name* **|** *ip address community* [*timeout*] [**configurationName** *name*]

host name | ip address
Required argument. Specifies either the name of the device, which the program can resolve into an IP address, or the device's actual IP address. The command retrieves the configuration from the device into the program's current configuration in memory.

community
Required argument. Specifies an SNMP community that is defined on the device with read access.

timeout
Optional argument. Specifies the length of time (in seconds) before the SNMP request expires. The default value is 10.

configurationName *name*
Optional argument. If you provide this argument, the configuration that is retrieved is saved with the specified name. Otherwise, it is saved with the current name. In either case, the configuration that is retrieved becomes the current configuration.

-save *directory config* [**configurationName** *name*]

directory
Required argument. Specifies the name of the directory into which you want to save the current configuration in the program's memory. If you specify an * instead of a name, the program uses the most recently named configuration directory. This command is functionally identical to the **-csfWrite** command.

Note: In AIX you can only specify an * instead of a name if you provide the **-save** command in a cfgargs or -commandsFrom file. It is not supported on the command line.

config Optional argument. Specifies the name of the configuration under which you want to save the current configuration. The default is the current configuration name. The physical file name under which the configuration is to be saved is an 8.3 version of the configuration name, with any invalid file name characters replaced with underscores and the file extension **.csf**.

configurationName *name*
Optional argument. If you provide this argument, the named configuration is written.

-selectConfiguration *config*

config Required argument. Specifies the name of the configuration to be selected. All **-asciiWrite**, **-csfWrite**, and **-send** commands use the selected configuration unless you explicitly name a configuration.

-send *host name* **|** *ip address community* [*timeout*] [**configurationName** *name*]

Note: If this command is issued before a valid configuration is read using either the **-asciiRead** or **-open** commands then the default configuration will be sent to the device. This default configuration will not contain an IP address for the device so if this configuration is activated then you will no longer be able to communicate with the device using IP.

host name | ip address
Specifies either the name of the device, which the program can resolve into an IP address, or the device's actual IP address. The command sends the program's current configuration in memory to the device.

community
Required argument. Specifies an SNMP community that is defined on the device with read access.

timeout
Optional argument. Specifies the length of time (in seconds) before the SNMP request expires. The default value is 10.

configurationName *name*
Optional argument. If you provide this argument, the named configuration is written.

-sendMultiple *send list*

send list Required argument. The send list contains the list of configurations and device addresses. You generate this list using the network configuration editor menu bar to select **Options** → **Communications** → **Send Multiple**, and completing the fields in the Send Multiple window. This command sends multiple device configurations to multiple devices.

-storeMessages **on** [*filename*] **off**

on Enables the logging of messages. If a log file exists, the system appends messages to it. If a log file does not exist, the program creates one.

filename
: Optional argument. Specifies the name of the file in which to put logged messages. The default is **cfg.log**.

off
: Disables the logging of messages.

Notes:

1. The **-storeMessages** command can appear anywhere on the command line, cfgargs file, or in any of the **-commandsFrom** files.
2. You can provide multiple **-storeMessages** commands, but the network configuration editor only honors the last one.
3. If you disable **-displayMessages** and enable **-storeMessages**, the program writes all messages to the log file.

Examples of Using the Command Line Facility With ASCII Files

In addition to being able to process ASCII files with the graphical interface, you can perform the following functions on them with the command line facility:

- Import one or more ASCII files
- Export one or more ASCII files

For details on doing these tasks, see “Importing from ASCII Format” on page 27 and “Exporting to ASCII Format” on page 29 as needed.

The following topics describe how to use the network configuration editor command line facility to process ASCII files. Examples are included to help you build cfgargs files so you can run certain processes in batch mode.

Importing an ASCII File

Use the **asciiRead** command to import (read) an ASCII file into the network configuration editor’s memory. The following example is provided to help you create a cfgargs file so you can run certain processes in batch mode. It imports the ASCII file named rtr656b.acf from directory asccfgs:

```
-displayMessages off
-storeMessages on cfgCmds.log
  -asciiRead d:\asccfgs\rtr656b.acf
-exit
```

Note: Validation errors are logged to file rtr656b.vlg.

Importing Multiple ASCII Files

You may want to import (read) multiple ASCII files and then export them to various devices. The following example assumes that you need to reload five devices with configurations, that you have valid ASCII files, and that you have already set up each of the devices to communicate with the network configuration editor. Each device has a unique configuration and you do not have to save the binary configuration.

```
-displayMessages off
-storeMessages on cfgCmds.log
  -asciiRead d:\asccfgs\rtr656f.acf
  -send mss656f mywritef
  -asciiRead d:\asccfgs\rtr656g.acf
  -send mss656g mywriteg
  -asciiRead d:\asccfgs\rtr656h.acf
  -send mss656h mywriteh
  -asciiRead d:\asccfgs\rtr656i.acf
  -send mss656i mywritei
  -asciiRead d:\asccfgs\rtr656j.acf
  -send mss656j mywritej
-exit
```

Exporting to ASCII from a Device

Use the **asciiWrite** command to export (write) an ASCII file. To export an ASCII file for a specific device, in the cfgargs file code a **retrieve** command (to read the configuration from the device) before you code the **-asciiWrite** command. The following example is provided to help you create a cfgargs file so you can run certain processes in batch mode. It exports an ASCII file containing the configuration for device CON656A (with comments) and then exits the network configuration editor:

```
-displayMessages off
-storeMessages on cfgCmds.log
  -retrieve CON656A myread
  -asciiWrite d:\asccfgs\con656a.acf withComments
-exit
```

Exporting to Network Device-Native Format from ASCII

Use the **createConfiguration** command to export (write) a network device-native format (**.cfg**) file (which you can use as backup for a device's configuration) from an ASCII file. The following example is provided to help you create a cfgargs file so you can run certain processes in batch mode. It imports an ASCII file, transmits it to a device, and then creates a **.cfg** file for backup.

```
-displayMessages off
-storeMessages on cfgCmds.log
  -asciiRead d:\asccfgs\mss656c.acf
  -send mss656c mywrite
  -createConfiguration d:\bincfgs\mss656c.cfg
-exit
```

Exporting to ASCII From a Retrieved Configuration

You may want to review the current configuration, for example, to debug or to plan for an upgrade. The following example of a cfgargs file shows how to run certain processes in batch mode. It retrieves the configuration from device RTR656D and exports the configuration to ASCII format for later use.

```
-displayMessages off
-storeMessages on cfgCmds.log
  -retrieve rtr656d readit
  -asciiWrite d:\asccfgs\rtr656d.acf
-exit
```

Exporting Multiple ASCII Files

The following example of a cfgargs file shows how to export (create) ASCII versions of configurations from multiple devices. It assumes that you have just received five devices that have been pre-configured. It also shows how to create documentation of the configuration for those devices.

```
-displayMessages off
-storeMessages on cfgCmds.log
  -retrieve rtr656a reada
  -asciiWrite d:\asccfgs\rtr656a.acf
  -retrieve rtr656b readb
  -asciiWrite d:\asccfgs\rtr656b.acf
  -retrieve rtr656c readc
  -asciiWrite d:\asccfgs\rtr656c.acf
  -retrieve rtr656d readit
  -asciiWrite d:\asccfgs\rtr656d.acf
  -retrieve rtr656e readit
  -asciiWrite d:\asccfgs\rtr656e.acf
-exit
```

Diagnosing Problems

If you experience problems with the network configuration editor, locate the symptom in Table 6 that most closely describes the problem and perform the actions described in the Recommended Actions column.

Table 6. Problem Symptoms

Symptom	Recommended Actions
Exit with error 34 message when attempting to start the network configuration editor in AIX	This message indicates that there is not enough space in the *\tmp* directory for the program to start. To avoid this problem, either delete unnecessary files in the directory or use SMIT to allocate more space to it.
Cannot read or make selections from a panel	Revert to the original initialization files by reinstalling the program.
The network configuration editor is running slowly or is hung	Stop other applications to see if that enables the program to operate normally. If it does not, revert to the original initialization files by reinstalling the program.
Internal Execution Error or Run Time Error (indicated by an error message in a pop-up window)	Do the following: 1. Record the version of the network configuration editor that you are using and any problem-related error messages. 2. Copy the following files in the network configuration editor's directory to a safe place: • The walkback.log file • The binary device configuration file (**.cfg**) that you were creating • The configuration storage file (**.csf**) that you were creating 3. Contact IBM Customer Service and provide the saved files.

Removing the Program

If you want to save your configurations for future use, before removing the network configuration editor, export them to a safe directory (outside of the program directory structure) in **.csf** format.

To remove the network configuration editor:

1. For Windows, use the Add/Remove Programs option from the Control Panel to uninstall the program.
2. For OS/2, use either the **del** or **erase** command to delete the files in the network configuration editor's directory, then delete the folder.
3. For AIX, use SMIT to uninstall the program.

Index

Readers' Comments — We'd Like to Hear from You

Network Configuration Editor User's Guide

Publication No. GC30-4031-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? □ Yes □ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name	Address
Company or Organization	
Phone No.	

Readers' Comments — We'd Like to Hear from You
GC30-4031-00

Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Design & Information Development
Department CGF/Bldg. 656
PO Box 12195
Research Triangle Park, NC
27709-9990

Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.