AT&T

386 UNIX® System V Release 3.1

Operations/System Administration Guide

NOTICE

The information in this document is subject to change without notice. AT&T assumes no responsibility for any errors that may appear in this document.

AT&T Products and Services

To order documents from the Customer Information Center:

- Within the continental United States, call 1-800-432-6600
- Outside the continental United States, call 1-317-352-8556
- Send mail orders to:

  AT&T Customer Information Center
  Customer Service Representative
  P.O. Box 19901
  Indianapolis, Indiana 46219

To sign up for UNIX system or AT&T computer courses:

- Within the continental United States, call 1-800-221-1647
- Outside the continental United States, call 1-609-639-4458

To contact marketing representatives about AT&T computer hardware products and UNIX software products:

- Within the continental United States, call 1-800-372-2447
- Outside the continental United States, call collect 1-215-266-2973 or 1-215-266-2975

To find out about UNIX system source licenses:

- Within the continental United States, except North Carolina, call 1-800-828-UNIX
- In North Carolina and outside the continental United States, call 1-919-279-3666
- Or write to:

  Software Licensing
  Guilford Center
  P.O. Box 25000
  Greensboro, NC 27420

Table of Contents

Table of Contents

Table of Contents

List of Figures

Introduction

Introduction

Chapter 1: Introduction

Purpose of This Guide

The purpose of this document is to provide you, the computer user, with the information you need to install and configure the UNIX system for your use, to operate the system on a day-to-day basis, and to successfully administer the system. It provides specific instructions for:

- Installing the Base System.
- Installing the optional Foundation Set software packages.
- Installing software applications.
- Changing the date and time.
- Adding, changing, displaying, and deleting user logins.
- Changing user passwords.
- Reporting system information.
- Formatting floppies and performing floppy-to-floppy copies.
- Creating, mounting, and unmounting UNIX system file systems.
- Setting up the computer to support parallel and serial devices.
- Setting up the electronic mail interface.
- Shutting down the system prior to turning off the power or rebooting.
- Backing up data to floppies or cartridge tape.
- Restoring data from floppies or cartridge tape.
- Determining the status of printers and their queues.
- Setting up Basic Networking.

Contents of This Guide

The material in this guide is organized into seven chapters, each separated by a tab. The following list summarizes the contents of each chapter:

- Chapter 1, *INTRODUCTION,* provides information necessary to understand the purpose and organization of this guide.
- Chapter 2, *SOFTWARE INSTALLATION,* tells you how to install the UNIX System V Operating System, Release 3.1, Version 1 Foundation Set on your system. In addition, this chapter tells you how to display software packages that are installed and how to remove add-on packages (if desired).
- Chapter 3, *USING THE **UNIX** SYSTEM SHELL,* describes the basics of the UNIX system shell such as entering commands, correcting mistakes, printing a file, running a program in the background, and a brief explanation of what manual pages are and how they are referenced.
- Chapter 4, *SYSTEM ADMINISTRATION,* describes all the administration tasks you can perform on your computer with the help of the AT&T Administration main menu.
- Chapter 5, *CUSTOMIZING YOUR COMPUTER,* describes how you can tailor the environment and system security on your computer.
- Chapter 6, *ALL ABOUT FILE SYSTEMS,* describes what file systems are and procedures for creating, checking, repairing, and using file systems.
- Chapter 7, *LINE PRINTER SPOOLER ADMINISTRATION,* describes how to administer Line Printer Spooling.
- Chapter 8, *BASIC NETWORKING ADMINISTRATION,* describes how to administer Basic Networking.
- Chapter 9, *REMOTE FILE SHARING ADMINISTRATION,* describes how to set up and administer the optional remote file sharing feature.
- Appendix A describes alternate control sequences if function keys are not available on your keyboard.
- Appendix B contains a list of commands that can be used in the interface when the computer keys do not exist on your keyboard.
- Appendix C contains a procedure to change passwords for other logins.

- Appendix D contains the procedures to network your computer with other computers such as how to physically connect direct links and modems, set up modems, and initialize modems. Some recommended switch settings are also provided.
- Appendix E contains a description of the error messages given using **fsck** (the file system check command.)
- The Glossary defines the menu, hardware, and software terminology used in this document.
- The Index gives an alphabetical listing of topics, together with the page numbers on which they appear in this guide.

Conventions Used in This Guide

Throughout this guide, there are certain conventions used to illustrate responses and how to react to them. The following is a list of the conventions you will see in this guide:

- Commands are in **bold** type.
- File names are *italicized.*
- `Light monospace` type indicates computer output, and **`bold monospace`** indicates computer input.
- Screen prompts and screen displays are in `light monospace`. Only one frame (menu, form, pop-up menu, or text frame) will be shown in screen layout figures. Partial screens are used and may not exactly duplicate what appears on your terminal screen.
- Keyboard references are in initial capital letters (to match the computer keyboard layout) and boxed. In this document the return key is represented by [Enter]. Also, the notation, [Ctrl] [d], means that you hold down the control key and strike the specified alphanumeric key, in this case, lowercase d.
- Highlighted menu items are in `reverse monospace`.
- Screen-labeled keys are all capital letters, with the key number inside parentheses [e.g., CANCEL, SAVE, CONT].
- References to other documents are in *italic* type.

Foundation Set Software Packages

The Foundation Set is the fundamental UNIX system software product supplied with your system. The Foundation Set provides you with the UNIX operating system kernel and a basic set of utilities. The Foundation Set consists of the following separately installable packages:

- Base System
- Editing
- Remote Terminal
- Security Administration
- 2K File System
- Network Support Utilities
- Remote File Sharing

The Base System package is the minimal required UNIX system. The other Foundation Set packages are optional, and you do not need to install them if you do not require the utilities they provide. Of course, you may install them later if you desire, and with the exception of the Remote Terminal Package, you may remove add-on packages if desired.

Base System

The Base System provides you with the UNIX operating system kernel, standard device drivers, and approximately ninety of the most essential UNIX system utilities. The Base System allows you to:

- Manage the hardware, software, and users on your computer system
- Perform Basic Networking (uucp) queued file transfers
- Share printers through the temporary storage of data and queuing of printer requests
- Tune system parameters to maximize performance under various load conditions, system configurations, and applications
- Modify the UNIX system to incorporate additional device drivers
- Run cooperating processes that share data and communicate with each other
- Perform mathematical calculations
- Check or change the executing environment of commands
- Schedule commands to be run at a later time
- Perform process accounting functions
- Install optional Foundation Set software packages and application software.

Editing Package

The Editing Package provides three related editors based on a consistent set of text editor commands: two line editors (**edit** and **ex**) and a screen editor (**vi**). The **edit** editor is mainly for novice users. The **ex** editor is an advanced version of **edit** and is for experienced users. The **vi** editor is a full screen editor that allows you to use your terminal as a window for viewing the text of a file. Within this window, you can add, delete, or change text in much the same way you would on a typewriter or with paper and pencil.

The Editing Package contains **spell** utilities that check for misspelled words in a file. It also contains other utilities that are helpful in developing shellscripts and examining files at the shell prompt.

Remote Terminal Package

The Remote Terminal Package consists of the *terminfo* data base that contains the descriptions and operating capabilities of over 150 popular terminal devices and terminal filters that allow a variety of terminals to print formatted output.

Security Administration Package

The Security Administration Package provides you with an encryption mechanism for protecting data that is being stored or transmitted. It gives your computer additional security protection beyond that obtained through login IDs, passwords, and permission modes. (This package has restricted distribution and is available only with systems intended for use within the United States.)

2 Kilobyte File System Utility Package

The 2 Kilobyte File System Utility Package provides an optional method of file system organization employing larger (2K) blocksizes to improve disk input/output (I/O) performance.

Network Support Utilities Package

The Network Support Utilities supplement the Base System by extending system capabilities to support networking applications. This includes standard STREAMS protocol modules (for use by applications), a network utility that monitors network service requests (the Listener), and the version of the sharable Network Services Library required to compile application programs.

Remote File Sharing Package

Remote File Sharing (RFS) allows you to share resources (directories containing files, subdirectories, devices, and named pipes) across a network with computers running UNIX System V, Release 3.0 or later releases. Administrators for computers on an RFS network can choose which directories on their systems they want to share and add them to a list of available resources on the network. From this list, they can choose resources from other computers on the network that they would like to use on their computers.

Each computer on an RFS network can be grouped with others in a "domain" or can operate as an independent domain. The domain can provide a central point for administering a group of computers. Unlike other distributed file systems used with the UNIX operating system, RFS is built into the operating system itself to provide compatibility, security, and flexibility.

Installing Software

The UNIX system is delivered ȯn diskettes organized in discrete packages as described in the previous section. Before you can use your computer, you have to install the minimum Base System.

The first floppy diskette in the Base System contains the UNIX operating system kernel and all commands necessary to install the rest of the Base System. After you have loaded the first diskettes, you then reboot the system from the hard disk.

After rebooting, you are ready to install the rest of the diskettes to complete the installation of the Base System.

With the Base System installed, you can then use the UNIX system automatic installation program to load the optional add-on packages that contain the rest of the UNIX system.

Getting Started

When you first turn on your computer (after all the appropriate software has been installed), it goes through a series of checks to make sure your computer is all right. The checks may take several minutes.

The following is a list of checks your computer makes when it is turned on:

- Runs a "resident diagnostic" program to check the hardware in your computer
- Determines whether the UNIX system was brought down correctly the last time it was used
- Boots the UNIX system and runs a file system check on system files if needed. The file system is checked for inconsistencies; if one should occur, the program will try to repair it.

You will then see the login prompt when the system checks are complete and everything is satisfactory.

Starting a Session With Your computer

A login name is one way of ensuring security on your computer system. Everybody using the computer has a different login name assigned to him/her. Refer to Chapter 4, System Administration, in "Add User Logins" to learn how to assign login names. When you turn on your computer, the following prompt will be displayed:

```
Console login:
```

If you are logging in from a remote terminal, your login screen will look like the following:

```
System name: <name>

Please login:
```

You can now enter your login name, followed by striking Enter. Your computer will now prompt you for your password. As you type in the password, it will not be displayed on the screen.

```
Password:
```

After you have entered the password and struck Enter, your computer will display the $ shell prompt and wait for the next command.

The Shell

The UNIX system shell is the starting point for your work with the computer. Each time you log in to the computer, you're in the shell. From here, you can perform user-oriented procedures or access the AT&T Administration main menu to do administration tasks.

NOTE Some logins (e.g., install) places the user directly into the AT&T Administration interface.

The Interface

Your computer gives you various administration features. System administration uses menus and forms to make the UNIX system administration easy. By using menus, you do not have to learn UNIX system commands to administer your computer.

When you initially log in to the computer, you are given a UNIX system shell prompt. To perform an administration task, you basically do the following:

- Access the AT&T Administration main menu
- Select the appropriate menus
- Supply the requested information
- Confirm that changes should be made.

Using Screen-Labeled Function Keys

The screen-labeled keys (SLKs) appear on the last line of your screen. SLKs are associated with the active frame and are executed by striking the corresponding function key on the keyboard (F1 through F8.) The SLKs change to provide different commands for different frames. The SLKs can be accessed from the following function keys:

F1	CANCEL
F2	CHOICES/MARK
F3	SAVE/CONT (continue)
F4	PREV-FRM (previous frame)
F5	NEXT-FRM (next frame)
F6	BLANK
F7	BLANK
F8	BLANK

If you do not have function keys, refer to Appendix A for alternative control sequences.

The screen-labeled keys are:

CANCEL is available for all frames. It cancels both the active frame and the task that the frame performs. If you did not confirm any activities, the active frame will close, ignoring changes. Once an activity begins (e.g., formatting a diskette) the CANCEL key will not interrupt the activity.

CHOICES is used to obtain a list of the options for the fields on a form. If choices are finite and the current field allows a selection, the options can be displayed. If the number of choices is two or three, the choices toggle in the field. When there are four or more possibilities, CHOICES opens a pop-up frame that is a menu of selectable items. When you select an item with [Enter], the menu disappears and the choice appears in the field.

CONT appears when the active frame is a form that requests a confirmation for an action or informs you that a task has been completed. When you strike CONT, the frame closes and the task is completed.

MARK is used to specify multiple items from a list on a pop-up menu. If you desire to only select one item, you can move the cursor to the appropriate item and strike [Enter]. If you desire to select several items, you can use the MARK function key. To select (mark) an item, move the cursor to the appropriate items and strike MARK. When you strike MARK, an asterisk (*) appears at the beginning of the items selected. If you change your mind about an item previosly selected, move the cursor to that item and strike MARK again. This unmarks the item and removes the asterisk.

SAVE appears when the active frame is a form that takes input on fields. When you have completed input to the form and are ready to save the changes in memory, strike SAVE. SAVE puts the changes into effect and closes the frame.

PREV-FRM is available for all frames to make the previously active frame the active frame again.

NEXT-FRM is available on all frames to move to the frame that was created after the active frame.

Command Line

All of the commands available on the function keys can be executed on the command line by typing Ctrl f followed by the function key number. For example, typing Ctrl f 3 performs the SAVE or CONT functions and typing Ctrl f 1 performs the CANCEL function.

UNIX system commands can be executed by typing a Ctrl z. When the --> prompt appears, enter the command. For example:

```
-->!date Enter
```

NOTE If your terminal becomes unreadable, you can type a Ctrl z, and when the --> prompt appears, type **refresh** to refresh the screen.

In addition, the command line can be used for navigation between frames by entering a frame number followed by an Enter. For example, type a Ctrl z, when the "-->" prompt appears, enter the following command:

```
-->3 Enter
```

will make frame 3 the active frame.

NOTE If you type an inappropriate command in the given context, use incorrect syntax or type the number of a non-existent frame; you will receive an error message.

You can also type Ctrl z <menu item>. For example, typing Ctrl z **exit** will exit the interface.

Function key commands can be abbreviated. However, the command must be uniquely identified by the letters that are entered. If the string is not unique, you will receive an error message that the string was not found. If this happens, try again using an abbreviation with more letters or the entire word. UNIX system commands can not be abbreviated. When a UNIX system command is executed, the screen clears for the display of the output. You are then prompted to strike Enter to return to the interface.

Editing on the Command Line

The following keys can be used for editing the command line using the computer keyboard. Additional keys are listed in Appendix B. Control key equivalents of all keys are available in Appendix A.

Del erases the character under the cursor. If there is no character under the cursor, it emits a beep.

Backspace erases the character under the cursor and moves the cursor to the left.

Spacebar moves the cursor to the right one character, replacing the current character with a blank space.

→ moves the cursor to the right one character and does not erase.

← moves the cursor to the left one character and does not erase.

Home moves the cursor to the beginning of the line and does not erase.

End moves the cursor to the last character typed on the line.

Enter executes the command typed on the command line.

Accessing the Interface

The following procedure will help you get started using menus. The procedure shows you how to login, access the AT&T Administration main menu, escape to a UNIX system shell and return to the AT&T Administration main menu, select various menus, and exit from the AT&T Administration main menu. If you have previous experience using menus, you should scan (rather than perform) this procedure.

NOTE: The following procedure uses "hotshot" as a typical example of a user login. Any valid user login can be used.

1. From the computer, log in as **hotshot**.

   ```
   Console login: hotshot [Enter]
   ```

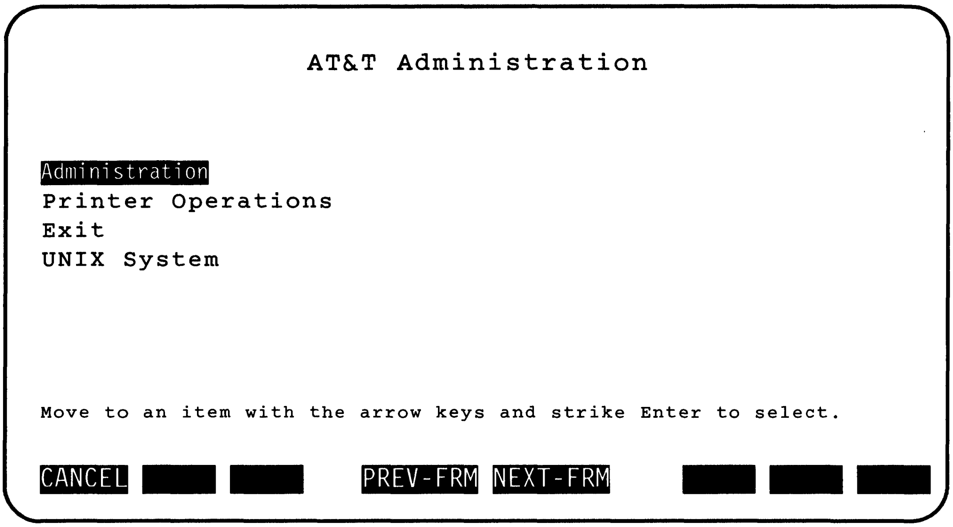
2. Enter your password. Note that the password is not echoed and thus does not appear on the terminal for security reasons.

   ```
   password: XXXXXXXX [Enter]
   ```

3. Access the AT&T Administration main menu by entering:

   ```
   $ adm [Enter]
   ```

   When this command is invoked, a AT&T Administration main menu appears as follows:



```
                    AT&T Administration


Administration
Printer Operations
Exit
UNIX System




Move to an item with the arrow keys and strike Enter to select.
```

4. On the AT&T Administration main menu, move to UNIX System with the arrow keys and strike Enter to select.

   The screen will clear and you will get the UNIX system shell prompt. You have invoked a UNIX system sub-shell without quiting the AT&T Administration main menu. This allows you to execute UNIX system commands and return to the AT&T Administration main menu without having to quit and start over.

5. When you desire to quit the sub-shell and return to the AT&T Administration main menu, enter:

   ```
   $ exit Enter
   ```

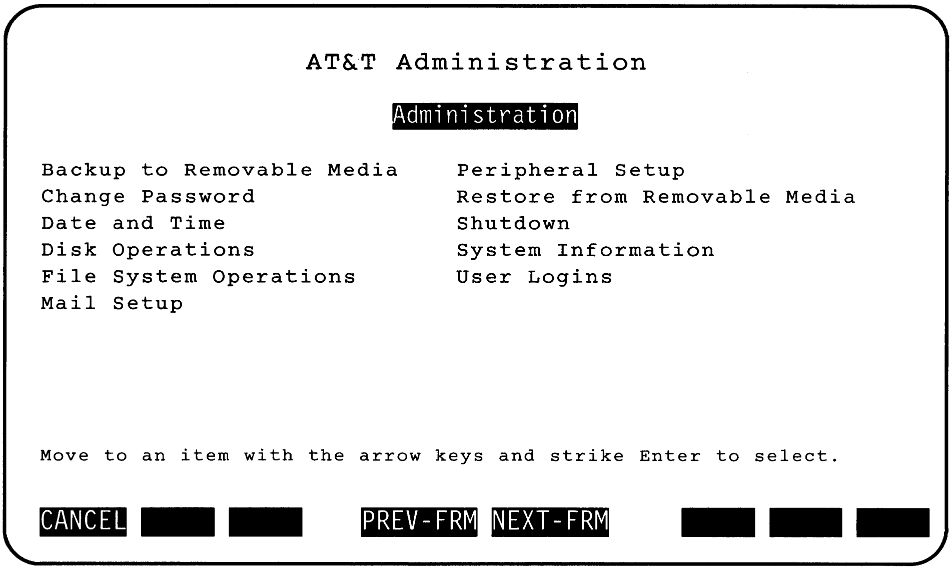
   or

   Ctrl d

   Then strike Enter to return to the AT&T Administration main menu.

6. From the AT&T Administration main menu, move to Administration with the arrow keys and strike Enter to select.

   The Administration menu appears as follows:



Move to an item with the arrow keys and strike Enter to select.

7. Return to the AT&T Administration main menu by striking the PREV-FRM or NEXT-FRM function key and select Exit.

   You will be informed that you are about to exit the interface and will be prompted to confirm with the CONT or CANCEL the exit.

   Confirm the exit with CONT, the screen will clear and you will get the UNIX system shell prompt.

   To return to the AT&T Administration main menu, you must enter the **adm** command.

Working With Frames

A frame is a bordered area that appears on your computer screen. Four types of frames are used in the interface: menus, forms, pop-up menus, and text frames. All frames (except for pop-up menus which are simply labeled CHOICES) have labels that include a frame number and a frame name. Frames are assigned according to the order that they are opened.

Several frames may be opened simultaneously on the display, and each frame is numbered as it is opened. Only one frame is active at a time. Labels on the screen-labeled function keys always refer to the active frame. The active frame may cover all or part of other frames on the screen display and will show the cursor resting on an item within the frame.

When you select an item from a menu, strike CANCEL, SAVE, or CONT; a new frame is created and that frame becomes the active frame.

Menus

Menus are frames with columnar lists of items. All lists are organized alphabetically with items beginning in uppercase appearing before lowercase items (with the exception of the AT&T Administration main menu.) All items are left justified with multi-column lists organized alphabetically down each column, wrapping to the next column.

Much of your administration work with the computer is done with menus. Menus make decision-making and problem-solving easier.

Navigating Within Menus

Moving around within menus and selecting an item is as easy as highlighting the item within your active menu and striking [Enter].

There are two distinct steps for navigating to a menu item.

- The following computer keyboard keys, the "key" method, can be used for navigation. The control command equivalents appear in Appendix A, and additional keys are listed in Appendix B.

The [↓] (down arrow key) moves the cursor down one item, stopping when it reaches the bottom of the list. In multi-column lists, the cursor moves to the top of the next column from the bottom of the previous column.

The [↑] (up arrow key) moves the cursor up one item, stopping when it reaches the top of the list. In multi-column lists, the cursor moves to the bottom of the previous column from the top of the next column.

The [→] (right arrow key) and [Spacebar] move the cursor down one item in a single-column menu and to the right one item in the multi-column menu.

The [←] (left arrow key) and [Backspace] move the cursor up one item in a single-column menu and to the left one item in a multi-column menu.

The (Home) key moves the cursor to the first item that is currently visible on the list.

The (Page/Down) key moves the cursor to the first item in the next frame of items and displays the entire next frame.

The (Page/Up) key moves the cursor to the first item in the previous frame and displays the entire previous frame.

The (End) key moves the cursor to the last object in the list, whether or not it is displayed.

On the console keyboard, there are two sets of arrow keys. One set is located between the numeric key pad and what is considered a normal typewriter keyboard, and the other set is located on the numeric key pad. For numeric key pad arrows to work, the Num Lock display light must be off.

- Another method of highlighting an entry is the "word search" or "character search." You can type the name of the item, the first character, or the first few letters of the item you want to highlight and the cursor will move to the item. Only as many letters as are necessary to make the entry unique are required. In many cases, only one letter is required. The character search is not uppercase or lowercase dependent. If the characters for a character search match more than one item on the menu, the cursor will stop on the first match. If no match is found, a beep will sound.

There are also two methods for selecting menu items:

- Enter
- Mark.

For example, if you are in the AT&T Administration main menu and want to select Printer Operations, you can strike the ↓ (down arrow key) enough times to bring the highlighting cursor down to the Printer Operations entry, and then strike Enter.

The Mark method is available on pop-up menus when multiple items can be selected for input. To select an item, you may use the arrow keys to move to that item and strike the MARK screen-labeled key when all selections have been MARKED, strike Enter.

Switching Frames

Navigation between frames can be done with simple moves when the frame is open, or non-simple moves which include opening the frame before moving to it. When navigating back to a frame, the cursor rests where it last rested before moving from the frame. When navigating to a frame that is newly opened, the cursor rests on the top item.

Simple Moves

You can have more than one menu displayed at a time, and you can switch between them with the PREV-FRM and NEXT-FRM screen-labeled keys. Just strike PREV-FRM to display the previous menu opened, or strike NEXT-FRM to display the next menu. You can loop between menus by striking the same PREV-FRM or NEXT-FRM function key.

You can also move to another frame by using Ctrl z [frame number] on the command line.

Non-Simple Moves

- By striking Enter while on a menu item that does not already exist in an open frame, a new frame opens with the cursor resting on the first item in the frame. If the frame was already open but not active, the cursor moves to that frame and rests on the item where it rested when the frame was last active.
- The SAVE function key saves the changes on the form and opens the next frame.
- The CANCEL function key closes the current frame and makes the previous frame the active frame. If you strike the CANCEL function key in the middle of a task, all the frames related to that task will be closed except for the first.
- Using Ctrl z **close** command on the command line without an argument will close the current frame and makes the previous frame active. If used with an argument, **close** cannot be used to navigate because it closes the frame specified by the argument and then returns to the frame where the command was initiated.
- Using Ctrl z [menu item name] will open a frame with a menu or form for that item. The cursor will rest at the top of the frame.

Interacting With Forms

Menus allow you to choose (by highlighting) certain tasks from a list of menu items. You strike Enter to select the menu item that you have highlighted.

A form, on the other hand, is displayed after a menu item is opened and input is required from you. Forms vary widely according to function. Forms are composed of items that prompt you for input. Some forms have fields that have an infinite number of valid inputs; others are limited to only a few inputs.

The Message Line provides directions on how to give input for the fields on a form. The CHOICES function key will list your choices when they are finite.

Once you have entered your data, follow the instructions on your screen. For example, when you have finished with the form, you may be required to use the SAVE function key to:

- Signal the interface to make the changes.
- Close the frame .

If you decide to discontinue the task, use the CANCEL function key.

Navigating Within Forms

You can move between the fields on a form using the following keys on the computer keyboard. Control key equivalents are listed in Appendix A, and additional keys are in Appendix B.

Enter moves the cursor to the next input field.

↓ moves the cursor to the next input field of the next item in the form and wraps back to the top when it reaches the bottom of the form.

↑ moves the cursor to the previous input field of the previous item in the form and wraps to the bottom when it reaches the top.

→ moves the cursor one position to the right within a field without erasing, until the end of the field.

← moves the cursor one position to the left within the field without erasing until the beginning of the field is reached.

Tab moves the cursor to the next field in the form, wrapping when the last field is reached.

Backtab moves the cursor to the previous field in a form, with wrapping from the first field to the last field when the top of the form is reached.

Home moves the cursor to the first character of the current field.

End moves the cursor to the first character of the last field in the form.

Input for Fields

Fields can do either or both:

- Accept input that you type from the keyboard
- Accept input from selections on a pop-up menu (menu mode).

If input is required from the insert mode only, using the CHOICES function key will cause a beep to be emitted.

Editing Fields

When you type the first character in the field, the field is automatically cleared. The following commands can be used to edit the input on a field. The keyboard equivalents are listed in Appendix A.

Backspace	erases the character under the cursor and moves the cursor one character to the left.
Spacebar	replaces the current character with a space, moving the cursor one position to the right.
→	moves the cursor one character to the right; it does not erase.
Del	deletes the character under the cursor and moves the remaining character to the left to fill the space. If there is no character under the cursor, it emits a beep.

Pop-up Menus

Pop-up menus are distinguished from menus because they disappear immediately after you respond to them. Pop-up menus appear when you strike the CHOICES function key for possible input to a field on a form. Navigation within a pop-up menu is the same as navigation within menus. Making a selection in a pop-up menu writes the value to the field on the form. The frame closes and the form becomes the active frame.

After you have used the CHOICES function key to get a pop-up menu, you can escape the pop-up menu without making any changes by using the CANCEL function key. The Enter key is also used to select an item from the pop-up menu and place it on a field on a form. When multiple items can be selected for input from the pop-up menu, MARK is used to mark the items followed by Enter to put the marked items on the form.

Text Frames

Text frames are used to give you information when no input is required. These frames provide information and do not have fields for navigation. To confirm or continue on a text frame, strike CONT.

Using the Exit Function

The AT&T Administration main menu lists the item Exit that allows you to exit the AT&T Administration main menu. When you select the Exit menu item from the AT&T Administration main menu, a Confirm Exit menu is displayed and you receive the following message:

```
Strike the CONT key to confirm, or CANCEL to cancel the exit.
```

If you desire to exit, strike CONT. Your screen clears and you are returned to the shell prompt. Other responses abort the exit request. To reenter the interface, you must type the **adm** command again.

Using the Escape Function

The AT&T Administration main menu also lists the item UNIX System that allows you to escape from the menu to the UNIX system shell prompt (i.e., invoke a UNIX system sub-shell) without quitting the AT&T Administration main menu. When you select UNIX System from the AT&T Administration main menu, your screen clears and you are returned to the special shell prompt $$. This escape differs from the "Exit" function in that you have not completely left the AT&T Administration main menu (i.e., only temporarily escaped to the UNIX system shell). You may return to the AT&T Administration main menu without having to invoke the **adm** command by typing **exit** or Ctrl d.

Shutting Down Your System

If you want to turn your computer off, always shut down the computer first or your files could be damaged. Shutting down your computer before turning off the power is necessary to ensure the integrity of your file system.

NOTE Shutdown is a selection in the AT&T Administration main menu.

In Case of Trouble

The UNIX system is designed to run your programs without difficulty. Occasionally, faulty software or hardware may cause problems. If you cannot get your computer to work correctly, try the following:

1. If your keyboard is locked up and will not respond to input, strike Ctrl q.

   If a Ctrl q does not work and you are at the console, do a Ctrl Break. If you are at a terminal, turn the terminal off.

2. Consult the documents provided with the computer.

3. Log off and then log back in again. If this does not help, go to the next item.

4. Shut down the computer according to the shutdown procedure in Chapter 4, System Administration. At the end of the shutdown procedure, strike the HARDWARE RESET button to reboot.

5. If this does not work, strike the HARDWARE RESET button.

   Do this step only if procedures 1 through 4 are unsuccessful.

Software Installation

Software Installation

Chapter 2: Software Installation

Introduction to Software Installation

This chapter tells you how to install the UNIX System V Operating System, Release 3.1, Version 1 on your system. First, you are provided with the detailed steps required to install the Base System, which is the set of UNIX system utilities required for most computing environments. Then, you are directed through the procedure for installing the optional add-on packages. In addition, this chapter tells you how to display installed software packages, as well as how to remove add-on packages.

Read Error Condition

If, when reading any diskette during installation of the Base System, a read error occurs (e.g., the diskette is not inserted, is misinserted, or is the wrong density), this shall be referred to in this document as a "Read Error" condition.

You will see the following message a maximum of three times, until the error is corrected or you strike Break (a break causes the installation process to abort, as discussed on the next page.)

```
A floppy disk read error has occurred. If necessary, please
consult your "Operations/System Administration Guide".

After correcting the error, strike ENTER to continue.
```

If you strike Enter, the attempt will be made again. If three attempts are made and the diskette still cannot be read, you will see the following termination message and you will be left in single-user mode with the hard disk unmounted:

```
Unable to read floppy disk. Installation terminated.
You may attempt to restart the installation process,
but if this problem recurs, please contact your support
representative immediately.
```

NOTE If the hard disk is unmounted, it is no longer accessible by the system. Refer to *mount*(1M) in the *User's/System Administrator's Reference Manual.*

Aborting the Installation Procedure

If at any point during the installation of diskettes you strike Break thereby aborting the installation, you will see the following "Abort Message":

```
You have aborted the installation of the UNIX
System. If you wish to rerun it, type INSTALL at
the prompt.  Please consult your "Operations/System
Administration Guide" for further information.
```

The installation process aborts but may be restarted at any time by typing **INSTALL**. You are left in the single-user mode with hard disk unmounted.

Hard Disk Formatting

The assumption is that your hard disk has been low-level formatted by you or the manufacturer. In the case of AT&T machines, the hard disk is customarily low-level formatted for your convenience.

NOTE

If the hard disk is not low-level formatted, strike Break and type:

```
/etc/mkpart -F3 /dev/rdsk/0s0
```

This formats the whole disk including any MS-DOS area.

Install Base System Diskette Number 1

Diskette number 1 contains the full Base System kernel and all commands necessary to install the remaining diskettes in the Base System. To install diskette number 1, do the following:

Boot System to Single-User Mode

1. Insert diskette number 1 into the diskette drive.
2. Press the hardware reset button or turn the system on to boot from the diskette.

   First, you will see the hardware diagnostics and then a message resembling the following:

```
total real mem = 3145728 (see note below)
total avail mem = 2031616

UNIX System V Release 3.1

Copyright (c) 1987 AT&T
ALL RIGHTS RESERVED

386/ix Drivers Copyright (C) 1986 Interactive Systems Corporation
ALL RIGHTS RESERVED
```

NOTE — These numbers will vary according to how much memory you have in your system.

3. When the Base System kernel has booted successfully into single-user mode, the system verifies that there is a minimum of 2 megabytes of system memory installed in the computer.

   If the 2 megabyte minimum is not installed, you will see the following message:

```
WARNING: Your system does not have the recommended minimum
2 Megabytes of memory. You may wish to power down the
machine, add memory, and begin the installation process again.
```

4. You will see the following prompt message (whether or not 2 megabytes are available):

```
Strike ENTER to install the UNIX System on your hard disk.
```

5. Strike Enter to install the UNIX system on your hard disk.

Partition the Hard Disk

1. Your screen should look similar to Screen A or Screen B shown on the next page.

2. Determine which screen layout is shown on your display.

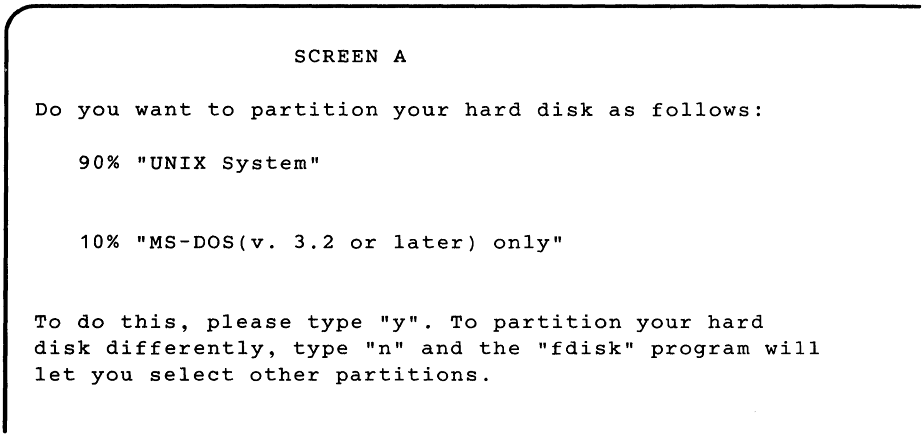
Screen A If you want to partition as shown on Screen A, type **y**, strike Enter, and go directly to the procedure to "Prepare Hard Disk For Surface Analysis."

If not, type **n** and strike Enter to tell the system that you do not want to set up your hard disk as shown in the message. Then go to Step 3.

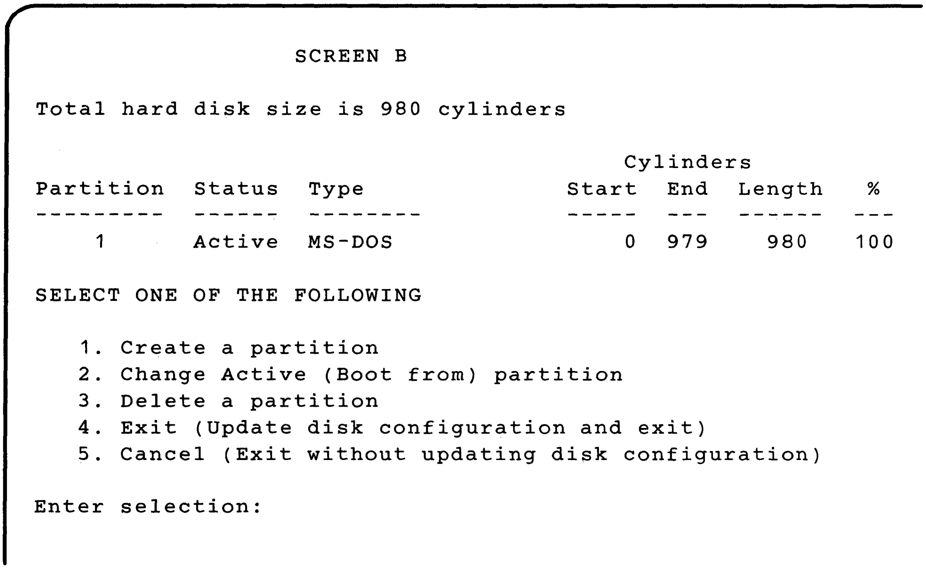
Screen B If the screen shows an active UNIX system partition, proceed to Step 8.

If the screen shows an MS-DOS partition, before continuing you need to delete the MS-DOS partition. Deleting a partition destroys all files in that partition. Before you delete the partition, copy any files you want to save to diskettes or optional tape. To delete the partition, type **3** and strike Enter and then type the number of the partition you want to delete and strike Enter again. After the partition has been deleted, go to Step 3.

NOTE: It is only necessary to delete the MS-DOS partition if you do not have space for the UNIX system partition.



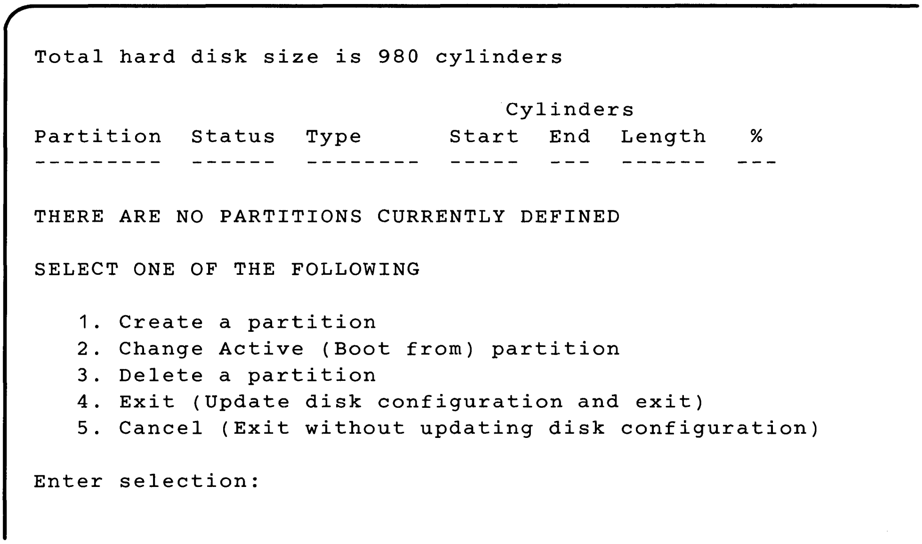
```
                    SCREEN A

Do you want to partition your hard disk as follows:

   90% "UNIX System"


   10% "MS-DOS(v. 3.2 or later) only"


To do this, please type "y". To partition your hard
disk differently, type "n" and the "fdisk" program will
let you select other partitions.
```



```
                    SCREEN B

Total hard disk size is 980 cylinders

                                                  Cylinders
Partition  Status  Type                  Start  End  Length   %
---------  ------  --------              -----  ---  ------  ---
    1      Active  MS-DOS                    0  979     980  100

SELECT ONE OF THE FOLLOWING

   1. Create a partition
   2. Change Active (Boot from) partition
   3. Delete a partition
   4. Exit (Update disk configuration and exit)
   5. Cancel (Exit without updating disk configuration)

Enter selection:
```

NOTE — The actual numbers shown in Screen B for hard disk size will depend on the size of your hard disk.

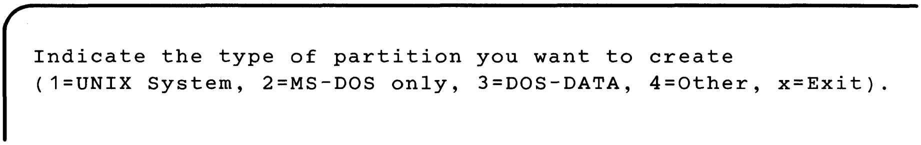
3. Make sure your screen looks like this with no partitions defined:



```
Total hard disk size is 980 cylinders

                                        Cylinders
Partition  Status  Type      Start  End  Length    %
---------  ------  --------  -----  ---  ------  ---

THERE ARE NO PARTITIONS CURRENTLY DEFINED

SELECT ONE OF THE FOLLOWING

   1. Create a partition
   2. Change Active (Boot from) partition
   3. Delete a partition
   4. Exit (Update disk configuration and exit)
   5. Cancel (Exit without updating disk configuration)

Enter selection:
```

4. Type **1** and strike Enter to select "Create a Partition."

   You will see the message:



```
Indicate the type of partition you want to create
(1=UNIX System, 2=MS-DOS only, 3=DOS-DATA, 4=Other, x=Exit).
```

5. Type **1** and strike Enter to select "UNIX System."

   You will see the message:

```
The UNIX System partition must use at least nnn% of the
hard disk.  Indicate the percentage (nnn-100) of
the hard disk you want this partition to use
(or enter "c" to specify in cylinders).
```

NOTE "nnn" depends on the size of your hard disk.

6. Type **100** and strike Enter.

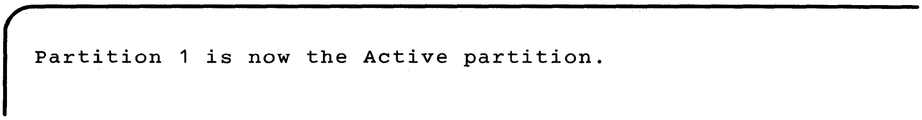
   You will see the message:

```
Do you want this to become the Active partition?
If so, it will be activated each time you reset
your computer or when you turn it on again.
Please type "y" or "n".
```

NOTE When your computer is turned on or reset, it looks for a diskette in the floppy disk drive. If it does not find one, it searches the hard disk for an active partition from which it can load an operating system.

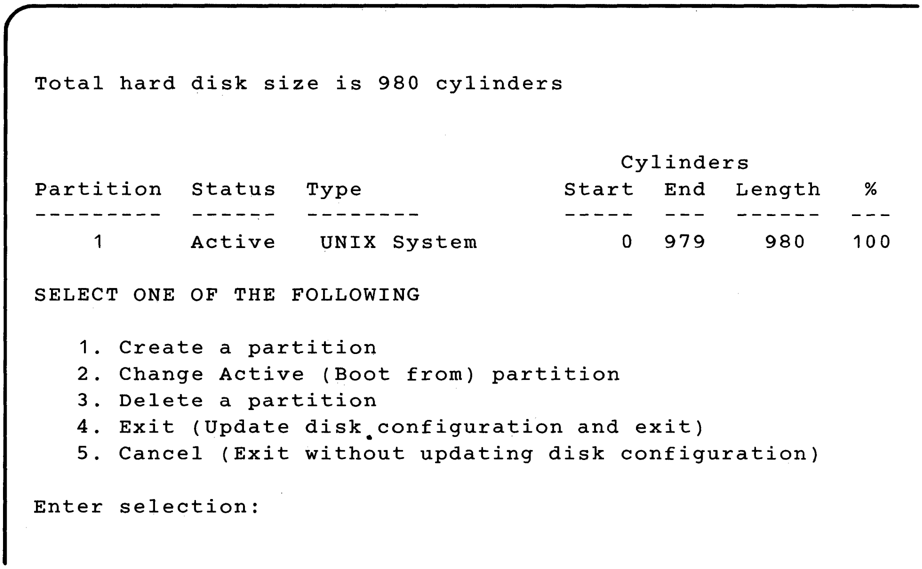
7. Type **y** and strike Enter to make the UNIX system partition active.

   At the bottom of your screen, you will see the message:



```
Partition 1 is now the Active partition.
```

   After the partition is created, your screen should resemble this:



```
Total hard disk size is 980 cylinders


                                               Cylinders
Partition  Status  Type                  Start  End  Length   %
---------  ------  ---------             -----  ---  ------  ---
    1      Active   UNIX System              0  979    980   100

SELECT ONE OF THE FOLLOWING

   1. Create a partition
   2. Change Active (Boot from) partition
   3. Delete a partition
   4. Exit (Update disk configuration and exit)
   5. Cancel (Exit without updating disk configuration)

Enter selection:
```

8. Type **4** and strike Enter to select "Exit."

Prepare Hard Disk for Surface Analysis

1. You will see the following message:

```
Hard disk partitioning complete.
```

2. The system tests to determine whether the active UNIX system partition has already been prepared for creation of a UNIX system file system.

   If the above tests pass, you will see the following message:

```
Your hard disk will now be set up for the UNIX System.

WARNING!  This procedure will destroy all data on
the active UNIX System partition of the hard disk! Do
you wish to continue (y or n)?
```

   If the active UNIX system partition had not been prepared, go directly to Step 4.

3. Set up your UNIX system partition as follows:

   - If you wish to set up a UNIX system partition, type **y** and strike Enter to continue.

     Go on to Step 4.

   - If you do not wish to set up the UNIX system partition, type **n** and strike Enter. You will see the following message:

```
UNIX System partition unchanged.
```

Go directly to the procedure to "Create UNIX System File Systems."

If you enter anything other than **y** or **n**, followed by Enter or Break (to be treated as **n**), you will see the following message until you enter a valid answer:

```
Set up UNIX System partition (y or n)?
```

4. You will see the following message:

```
Checking for bad sectors in the UNIX System partition...
```

This message tells you that the System is performing a surface analysis of the hard disk and generating a table of the defective blocks. While the UNIX system partition is being scanned, you will see the following message:

```
Checking cylinder: nnn
```

where "nnn" is updated for each cylinder scanned.

5. If, during the identification of defective blocks, the table overflows, you will see the following message, the installation aborts, and you will be left at the UNIX system prompt:

```
Error: Your UNIX System partition has too many bad
blocks. A UNIX System cannot be installed
on it. Please restart the installation procedure
by typing "INSTALL" at the UNIX System prompt and
repartition your hard disk so that the UNIX System
partition is located elsewhere.

Installation aborted.
```

If this occurs, restart the installation process by typing **INSTALL**. When you perform the fdisk procedure ("Partition the Hard Disk") on Page 2-6, manually place the UNIX system partition somewhere else. If this does not clear the problem, contact the manufacturer regarding the integrity of the hard disk.

Create UNIX System File Systems

1. When the surface analysis is completed, the system tests to see if a valid file system exists on the active UNIX system partition.

   - If a valid file system exists, you will see the following message:

```
Do you wish to create new UNIX System file system(s)
  on the hard disk? This will destroy all data
  on the UNIX System partition (y or n)?
```

     If you wish to create a new file system, type **y** and strike Enter. Then go directly to Step 2.

     If you do not wish to create a new file system, type **n** and strike Enter. Skip the rest of this section and go directly to the procedure to "Complete Installation of Diskette Number 1."

   - If you do not see the above message, go directly to Step 2.

2. When the surface analysis is completed, the system calculates the optimal amount of space on your hard disk for swap, user, and/or *root*(/) file systems. You will see a message that begins as follows:

```
The UNIX System partition has nnn cylinders assigned to it.
nn cylinders will be used for alternate sectors.
This leaves nnn cylinders (nnnnn bytes) available.

The following seems like a reasonable partitioning of
your UNIX System disk space:
```

NOTE

Throughout this procedure, "nnn" depends on the partition size.

3. You will see the file system size selections in the following messages:

   If no separate */usr* file system was called for, based on your partition size you will see the following message:

```
A combined root/user filesystem of nnn cylinders(nnnnnnn bytes),
```

Otherwise, you will see the following message:

```
A root filesystem of nnn cylinders (nnnnnnn bytes),
a user (usr) filesystem of nnn cylinders (nnnnnnn bytes),
```

In either case, if an additional */usr2* file system were called for, you will see the following message:

```
an extra user filesystem (/usr2) of nnn cylinders (nnnnnnn
bytes),
```

The message will be completed with:

```
and a swap/paging area of nnn cylinders (nnnnnnn bytes).
```

4. You will see the following prompt message:

```
Is this allocation acceptable to you (y/n)?
```

If this allocation is acceptable, type **y**, strike Enter and then proceed to Step 12.

If not acceptable, type **n** and strike Enter.

5. You will see the following message:

```
Do you wish to have separate root and user filesystems (y/n)?
```

If you wish to have separate root and user file systems, type **y** and strike Enter.

If you wish that root and user be combined, type **n** and strike Enter.

NOTE With many users and limited hard disk space (less than 68 megabytes), it is advisable to create separate root and user file systems.

6. You will see the following message:

```
Do you want an additional /usr2 filesystem (y/n)?
```

Generally, an additional */usr2* file system is not necessary. However, if you wish to have an additional */usr2* file system, type **y** and strike Enter.

If you do not want the above, type **n** and strike Enter.

7. You will see the following message:

```
You will now be given the opportunity to specify the
size, in cylinders, of each filesystem. (One megabyte
of disk space is approximately nn.n cylinders).
```

followed by the prompt message:

```
How many cylinders would you like for swap/paging (1-nnn)?
```

NOTE

In the above message, nnn is the maximum legal size of swap space, calculated as the total space (UNIX system partition less cylinders reserved for alternates) minus 20 megabytes.

Enter the desired parameter and strike Enter. If your answer was not in the given range, you will see the following message and Step 7 will be repeated:

```
Illegal value: nnn; try again.
```

If only one file system was selected, you will see the following message:

```
The remaining nnn cylinders will be assigned to root/usr.
```

NOTE In the above message, nnn is the remaining space after swap space is subtracted.

If only one file system was selected, then proceed to Step 10.

8. You will see the following message:

```
How many cylinders would you like for root (1-nnn)?
```

NOTE In the above message, nnn is the maximum legal size for *root* (/), which is the amount of space remaining after swap space has been subtracted.

Enter the desired parameter and strike Enter.

If your answer is not in the given range, you will see the following message and Step 8 will be repeated:

```
Illegal value: nnn; try again.
```

If there is no space left after subtracting the *root(/)* file system, only one file system shall be used, you will see the following message, then proceed to Step 10.

```
No space remaining for a user filesystem.
Assuming single root/usr filesystem.
```

If an additional */usr2* file system was not selected, you will see the following message, then proceed to Step 10.

```
The remaining nnn cylinders will be assigned to /usr.
```

If an additional */usr2* file system was selected, you will see the following message:

```
The remaining nnn cylinders will be assigned to /usr2.
```

9. You will see the following message:

```
How many cylinders would you like for /usr (1-nnn)?
```

NOTE In the above message, nnn is the maximum legal size for */usr,* which is the amount of space remaining after *root*(/) has been subtracted.

Enter the desired parameter and strike Enter.

If the answer is not in the given range, you will see the following message and this step will be repeated:

```
Illegal value: nnn, try again.
```

If there is no space after subtracting the */usr* file system, only two file systems shall be used, you will see the following message, and the process continued with Step 10:

```
No space remaining for a /usr2 filesystem.
Assuming just root and /usr filesystems.
```

Otherwise, you will see the following message:

```
The remaining nnn cylinders will be assigned to /usr2.
```

10. You will see the following message:

```
You have specified the following disk allocation:
```

Return to Step 3.

11. For each selected file system, you will see the following message:

```
A xxx filesystem will now be created on your hard disk...
```

NOTE

In the above message, xxx is either "*root(/)*", "*/usr*", or "*/usr2.*"

Skip Step 12 and go directly to the procedure to "Verify Successful File System Creation."

12. You will see the following message:

```
UNIX System file system(s) will now be created
  on your hard disk...
```

Verify Successful File System Creation

1. If an error was encountered in the creation of any of the UNIX system file system(s), you will see the following message:

```
An error has occurred while setting up your hard disk.
Strike ENTER to install again.
```

If you see this message, strike Enter to install again and go back to the procedure to "Partition Your Hard Disk."

NOTE If you strike Break, the Abort Message and procedure on Page 2-2 will be deployed.

2. If the mkfs commands were performed successfully, you will see the following message:

```
UNIX System file system(s) have been created
in your active UNIX System partition.

A UNIX System will now be installed on your
hard disk.....
```

Go on to the next step.

Complete Installation of Diskette Number 1

1. The *root(/)* file system is mounted. If this fails, you will see the following message (up to three times):

```
Mounting root file system failed, trying again ...
```

   If the mount attempt fails three times, you will see the following message:

```
Cannot mount the root file system.
Please notify your AT&T services representative for further
assistance.
```

   If the above occurs, the installation will abort and you will be left in single-user mode with the hard disk unmounted.

2. All files on diskette number 1 are copied to their respective directories on the hard disk.

3. You will see the following message:

```
Please remove the floppy disk from the
drive and strike CTRL-ALT-DEL to reboot
from the hard disk.

Reboot the system now.
```

NOTE If you strike any key other than Ctrl, Alt, and Del simultaneously, it will be ignored by the system and not echoed.

4. Strike Ctrl, Alt, and Del simultaneously to reboot the system.

Install the Remainder of the Base System

When the system has rebooted from the hard disk, a procedure is initiated automatically that:

- Copies all files from the remaining diskettes to the hard disk
- Sets the date and time
- Sets the system name
- Sets special login passwords for root and install
- Presents the login prompt.

If, at any time, you strike Break, you will not receive a message and will be left in single-user mode at the prompt with the hard disk mounted, and in the *root*(/) directory on the hard disk. You may restart this procedure by typing **INSTALL.**

The procedure to install the remainder of the Base System is as follows:

1. You will see the following message for each diskette remaining, beginning with diskette number 2:

```
Please insert the UNIX System "Base System Package"
Floppy Disk <n> of 8 and then strike ENTER.
```

Strike Enter and the following "in-progress" message will appear:

```
Installation is in progress -- do not remove the floppy disk.
```

2. The sequence number of the diskette shall be verified. If a floppy read error is detected, the Read Error procedure on Page 2-1 shall be used. If the error is corrected and you strike Enter, the "in-progress" message will reappear.

3. If the sequence number of the diskette is not correct, you will see the following message until you insert the correct diskette or strike Break.

NOTE A break leaves you in single-user mode at the prompt with no message and the hard disk mounted.

```
The inserted floppy disk is incorrect.  Please insert
the floppy disk labeled <n> of 8 and strike ENTER.
```

When you strike Enter, the "in-progress" message appears. Again, if a read error occurs, the Read Error procedure on Page 2-2 is used.

4. Once the correct diskette has been inserted, its contents will be copied to the hard disk.

5. Repeat this procedure (beginning at Step 1) until all Base System diskettes are read into the system.

Wrapup Base System Installation

1. When you have completed the installation of the Base System diskettes, you will see the following message:

```
UNIX System files have been copied to the hard disk. It is now
safe to remove the floppy disk. Additional system files
will now be set up. Please stand by ...
```

   The system is checked. This will take a few minutes.

2. You are presented with the administration procedure to enter the date and time.

   Enter the date and time using the procedures in Chapter 4, System Administration. Strike the DONE function key when you finish.

   When you enter the information, the system clock is updated.

3. You are presented with the administration procedure for Mail Setup to set the UNIX system name.

   Set the system name using the procedures for "Mail Setup" in Chapter 4, System Administration. Strike the DONE function key when finished.

4. Canceling the setting of the system name is not allowed. If you canceled it, you will see the following message:

```
You must enter a system name
before proceeding.

Strike ENTER to continue
```

   Strike Enter to reinvoke the administration procedure. Return to Step 3 to set the UNIX system name.

5. You will see the following prompt message:

```
Enter a password for the "root" or
super-user.

(Note: This password must be kept
EXTREMELY secure):
```

   Enter the "root" password and strike Enter.

   You will see the following message:

```
Re-enter new password.
```

   Reenter the "root" password and strike Enter.

6. You will see the following message:

```
Enter a password for the "install" user.

(Note: This password must be kept
EXTREMELY secure and should be different
from the root password):
```

Enter the "install" password and strike Enter.

You will see the following message:

```
Re-enter new password.
```

Reenter the "install" password and strike Enter.

7. You will then see the following message:

```
The UNIX System installation process is now complete.

To install the Foundation Set Add-on packages, use
the "installpkg" command from the UNIX System prompt.

Be sure the floppy drive is empty and
strike CTRL-ALT-DEL to reboot your newly
configured UNIX System.

Reboot the system now.
```

Reboot the UNIX System

1. Strike Ctrl, Alt, and Del at the same time to reboot into the UNIX System from the hard disk.
2. You will see the following message:

```
Welcome to the AT&T 386 UNIX System
System name: <name>

To log in and begin customizing your system, type "install" at

the prompt.

Console login:
```

   When you log in as "install" and enter the correct password, you will be presented with the AT&T Administration main menu.

3. After you have logged in, you should make a copy of the first diskette for backup.

NOTE	The first diskette contains necessary files in the event your UNIX system becomes corrupted.

Install Optional Add-On Packages

After you have completed the installation of the Base System, you are ready to install the add-on packages that make up the rest of the Foundation Set, as well as other third-party software packages.

This section, "Install Optional Add-On Packages" provides an overview of the installation process and a general installation procedure that applies to installing all add-on packages except the Remote Terminal Package.

NOTE The procedures to install the Remote Terminal Package are contained in a separate section in this chapter.

Overview of Installation Software

The Base System Package contains a System Add-On Installation Procedure program which installs add-on software packages. The first diskette in each add-on software package contains an Add-On Install Script that is used by the System Add-On Installation Procedure to physically install the package.

General Instructions

Use of the Enter and Esc Keys

Throughout the procedure to install Optional Add-On packages, the following will apply for error conditions (except as noted otherwise) whenever you are prompted to strike Enter or Esc:

- If you strike Esc, you will see the following message:

```
The Installation is canceled.
```

- If you strike any keys other than Enter or Esc, the system will beep.
- If you strike Enter, the installation procedure will be restarted with the procedure to "Install First Add-On Package Diskette".

Use of the Break Key

If you strike Break at any point during the software installation, up to the point where the application's Install script begins executing, you will see the following message:

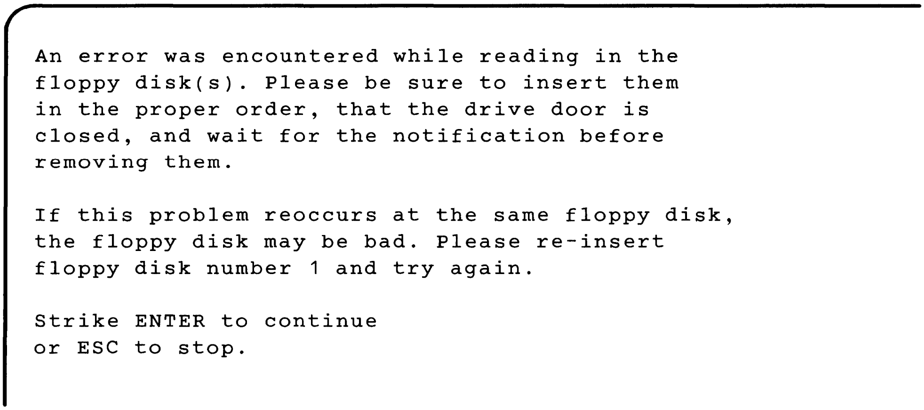
```
You have canceled the installation. If you wish
to try it again strike ENTER, otherwise strike ESC.
```

Your response to the above prompt will be as described in the preceding paragraph.

If you strike Break while the application's Install script is executing, the break will be ignored and no message displayed in order to leave the system in a sane, if not necessarily a desirable state.

Diskette Handling

If the diskette drive door is open ,the diskette has an error, or the diskette is incorrectly inserted, you will see the following message:



```
An error was encountered while reading in the
floppy disk(s). Please be sure to insert them
in the proper order, that the drive door is
closed, and wait for the notification before
removing them.

If this problem reoccurs at the same floppy disk,
the floppy disk may be bad. Please re-insert
floppy disk number 1 and try again.

Strike ENTER to continue
or ESC to stop.
```

You will be given the opportunity to correct the problem and start over.

When add-on packages are installed, files are copied to a temporary "holding" directory in order that checks may be performed as the installation process continues.

CAUTION Do not open the diskette drive door while files are being installed.

If, during the copying of files from the diskette to the hard disk, the hard disk runs out of space in the temporary directory, the temporary directory and any/all of its contents will be removed, the installation procedure will abort, and you will see the following message:

```
Your user (/usr) filesystem is out of space.
Please remove some files and try again.

Installation aborted.
```

If the system finds that the diskette's file header information is invalid, you will see the following message:

```
The floppy disk you inserted is either not the correct floppy disk
or you inserted it in the wrong order. If this problem reoccurs at
the same floppy disk, the floppy disk may be bad. Please re-insert
the first floppy disk and try again.

Strike ENTER to continue
or ESC to stop.
```

If you wish to continue, remove the diskette and start again at Step 1 of the procedure to "Install the First Add-On Package Diskette."

If you wish to stop here, strike [Esc].

Check for Install Permission

To install a software package, do the following:

1. Type **installpkg** at the UNIX system prompt, and strike Enter.
2. The system checks to see if you have permission to execute privileged operations. If you do not, you will see the following message:

```
You (<logname>) do not have permission to
perform software installation.
Please consult your Operations/System
Administration Guide for more
information on assigning permissions
to privileged operations.
```

   Refer to Chapter 4, System Administration.

3. The system also checks to see if you are at the console. If not, you will see the following message:

```
You must be on the console to run installpkg.
```

   The procedure terminates.

Install First Add-On Package Diskette

1. You will see the following message:

```
Please insert the floppy disk.

If the program installation requires more than
one floppy disk, be sure to insert the disks in
the proper order, starting with disk number 1.

After the first floppy disk, instructions will
be provided for inserting the remaining floppy
disks.

Strike ENTER when ready
or ESC to stop.
```

2. Insert diskette number 1 into the drive and strike Enter.

   You will see the following message:

```
Installation is in progress -- do not remove the floppy disk.
```

3. If the system detects that the diskette is in the "sysadm" (AT&T 3B2 Computer simple administration) format, the install script provided on the diskette by the manufacturer will be executed. You may skip the rest of this installation procedure and instead follow the prompt messages that will appear on your screen to complete the installation of the package.

4. If the diskette is not in the "sysadm" format, your screen will be cleared and the system will attempt to extract the file called "Size" from the first diskette. While this attempt is in progress, you will see the following message:

```
Searching for the Size file
```

5. If the Size indicates that there is enough room on the hard disk to install the package, you will see the following message:

```
           Install in progress


Transfer in progress - Do not remove the floppy disk.
```

   When you see the above message, go directly to Step 8.

6. If the Size file is not found, the package is presumed to be an old style (e.g., AT&T PC 6300 PLUS) package and you will see the following message:

```
Please enter the number of floppies in
the package followed by ENTER:
```

   Enter the number of diskettes and strike Enter.

   NOTE: Until you enter a numeric value greater than or equal to 1 followed by Enter, the system will beep for each invalid response.

You will see the following message:

```
Please enter 1 (for 360 KB) or 2 (for 1.2 MB)
for disk density followed by ENTER:
```

Enter **1** or **2** and strike Enter.

NOTE: Until you enter either a **1** or **2** followed by Enter, the system will beep for each invalid response.

Go back to Step 1 of the procedure to "Install the First Add-On Package Diskette." After Step 1, go directly to Step 8.

7. If the Size file is found but the system determines that it is invalid, you will see the following message:

```
Invalid Size file found. Cannot determine disk requirements.
```

Enter the number of floppies and density as directed in the preceding step.

8. If the package's hard disk requirement exceeds the space available on either the *root(/)* or *usr* file system, you will see the following message:

```
There is not enough room on the hard
disk to install the package. Please
remove some files from the <filesystems>
filesystem(s) and try again.
```

NOTE

In the above message, "<filesystems>" may be replaced with "*root(/)*" to indicate the *root(/)* file system or "user (*/usr*)" for the *usr* file system. If both file systems lack adequate space, "*root(/)* and user(*/usr*)" shall be used.

9. When loading of the diskette is completed and the package has more than one diskette, you will see the following message:

```
Reached end of medium for input.
You may remove the floppy disk.
To QUIT - strike <q> followed by ENTER.
To continue - insert floppy disk number <n+1>
and strike the ENTER key.
```

NOTE

Where n is the current diskette number. Example: If this is diskette number 1, the n+1 diskette will be diskette number 2.

When you see this message, remove the diskette from the drive.

10. If the package consists of only one diskette, go directly to the procedure to "Complete Installation of Add-On Package."

Install Additional Add-On Package Diskettes

1. Insert the next diskette in the drive, and strike [Enter].

   You will see the following message:

```
Transfer in progress - Do not remove the floppy disk.
```

NOTE If you type anything other than [Enter], the system will beep.

2. When loading of the diskette has completed and there are more diskettes in the package, you will see the following message:

```
Reached end of medium for input.
You may remove the floppy disk.
To QUIT - strike <q> followed by ENTER.
To continue - insert floppy disk number <n+1>
and strike the ENTER key.
```

If you have additional diskettes to install for this package, return to Step 1.

If this was the last diskette in the package, go directly to the procedure to "Complete Installation of Add-On Package."

Complete Installation of Add-On Package

1. The system tests for the presence of the required installation files. If any are missing, you will see the following message:

```
The software package is missing the
necessary installation programs. Please
check to make sure you have the right
floppy disk(s).

Strike ENTER to restart installation
or ESC to stop.
```

   If you receive the above message and wish to restart the installation, strike Enter. If you wish to cancel the installation, strike Esc.

2. If the system determines that an identical software package has already been installed, you will see the following message:

```
The <package name> package has already
been installed. The new installation
will now replace the original <package name>
files.

Strike ENTER to continue
or ESC to stop.
```

   You may strike Enter to replace the previous "instance" of the package with the new one and continue the installation, or you may strike Esc to cancel the installation.

   If you strike Esc, you will see the following message:

```
The Installation is canceled.
```

and you will be left at the command line prompt.

NOTE If you type anything other than Enter or Esc, the system will beep.

3. Your screen will clear and the package's "Install" script will begin execution.

NOTE If the package just installed contained a UNIX system driver,you will see messages informing you that the UNIX system kernel is being reconfigured. This takes a few minutes. When the reconfiguration of the kernel has completed, you will be asked if you want to shut your system down and reboot your UNIX System with the newly configured kernel. If you choose not to, shut down your system and reboot as soon as possible. When you have rebooted, the installation process is complete.

4. If no driver was installed and the installation is successful, you will see the following message:

```
The installation of the <package name>
package is now complete.
```

You will be left at the command line prompt.

5. If the installation is not successful, you will be left at the UNIX system prompt with the installation terminated.

Install the Remote Terminal Package

To install the "Remote Terminal" package in the Foundation Set (i.e., to add a terminal definition or definitions to your system for use with applications), do the following:

Check for Install Permission

1. Type **installpkg** at the UNIX system prompt and strike Enter.
2. The system checks to see if you have permission to execute privileged operations. If you do not, you will see the following message:

```
You (<logname>) do not have permission to
perform software installation.
Please consult your Operations/System
Administration Guide for more
information on assigning permissions
to privileged operations.
```

   Refer to Chapter 4, System Administration.

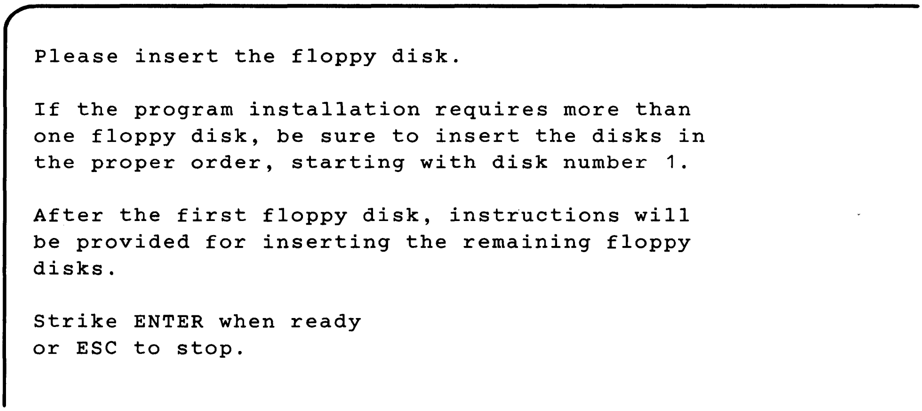
3. The system also checks to see if you are at the console. If not, you will see the following message:

```
You must be on the console to run installpkg.
```

   The procedure terminates.

Install Remote Terminal Package Diskette

1. You will see the following message:

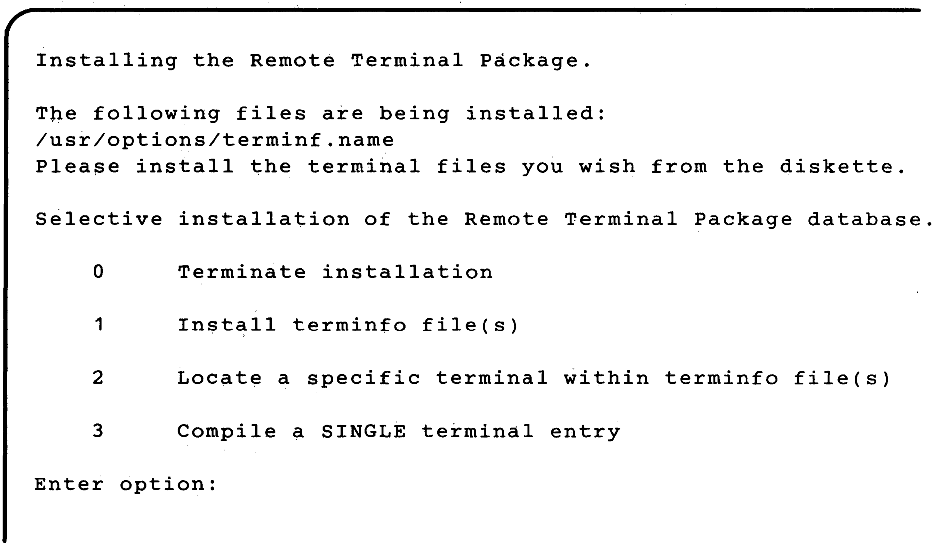


```
Please insert the floppy disk.

If the program installation requires more than
one floppy disk, be sure to insert the disks in
the proper order, starting with disk number 1.

After the first floppy disk, instructions will
be provided for inserting the remaining floppy
disks.

Strike ENTER when ready
or ESC to stop.
```

2. Insert diskette number 1 of 1, containing the Remote Terminal Package, into the drive and strike Enter.

Install Terminal Files

You will see the following message:



```
Installing the Remote Terminal Package.

The following files are being installed:
/usr/options/terminf.name
Please install the terminal files you wish from the diskette.

Selective installation of the Remote Terminal Package database.

    0     Terminate installation

    1     Install terminfo file(s)

    2     Locate a specific terminal within terminfo file(s)

    3     Compile a SINGLE terminal entry

Enter option:
```

Locate a Terminal File

If you wish to locate the terminal information file for a specific terminal within the *terminfo* data base on the diskette, type **2** and strike Enter.

You will see the following message:

```
Enter terminal name to be located:
```

Suppose you wish to locate the file for an AT&T 5425 terminal. In that case, you would type the terminal name **5425** and strike Enter.

You will see the following message:

```
Terminal 5425 is located within terminfo file "att.ti"
```

Select File(s)

You may wish to display the *terminfo* data base file listing and select a file(s) to install. In that case, you will type **1** and strike Enter.

You will see the following message:

```
The following terminfo files may be selected for installation:

adds.ti          annarbor.ti    ansi.ti         att.ti
beehive.ti       cdc.ti         colorscan.ti    contel.ti
datamedia.ti     dec.ti         diablo.ti       fortune.ti
general.ti       hardcopy.ti    hazeltine.ti    hds.ti
heath.ti         homebrew.ti    hp.ti           lsi.ti
microterm.ti     misc.ti        pc.ti           perkinelmer.ti
print.ti         special.ti     sperry.ti       tektronix.ti
teleray.ti       televideo.ti   ti.ti           tymshare.ti
visual.ti

Enter a file name, "all", "done", or "files":
```

You may enter a file name to install an entire file or type one of the following and strike Enter.

- "**all**" to install all files in the *terminfo* data base.
- "**done**" to return to the main menu.
- "**files**" to relist the files (see above).

Compile a Single Terminal Entry

You may wish to compile a SINGLE terminal entry. To do that, type **3** and strike Enter.

You will see the following message:

```
Enter terminal name:
```

If, for example, you wish to enter an AT&T 5425 terminal, type the terminal name **5425** and strike Enter.

You will see a message similar to the following:

```
Working in /usr/lib/terminfo
Created 5/5425
Linked 4/4425
Linked A/ATT4425
Linked A/ATT5425
Linked a/att4425
Linked a/att5425
Linked t/tty5425
```

Complete Installation of Remote Terminal Package

When you are ready to complete the installation of the Remote Terminal Package, type **0** and strike Enter.

You will see the following message:

```
The installation of Remote Terminal package is now complete.
```

You will be left at the UNIX system prompt.

Display Installed Software Packages

To get a sorted list of the software packages currently installed on your system, do the following:

1. Type **displaypkg** at the UNIX system prompt and strike Enter.

   A sorted listing of the currently installed software packages will be displayed.

2. Use:

   - "/"
   - "?"
   - "-"

   and all documented features of the "pg(1)" command to view the list. Refer to the User Reference Manual for more information on the "pg(1)" command.

   NOTE "pg(1)" is normally installed as part of the Editing Package. You do not need to install the Editing Package to use **displaypkg**.

3. When you are finished, enter **q** at the ":" prompt. This will leave you at the UNIX system prompt.

Remove Add-On Software Package

If your hard disk is low on space, you may wish to remove some software packages which are not commonly used. The **removepkg** command lets you do this.

You may type **removepkg** followed by a "package name" (in double quotes) to remove a package or you may type **removepkg** without arguments and select the package that you wish to remove from a list of installed packages.

1. Enter the **removepkg** command as follows:
   - To remove a package on the command line, type **removepkg** <**"package name"**> and strike Enter.

NOTE	The package names are identified in the */usr/options* directory.

   - To select the package from a list, type **removepkg** (without arguments) and strike Enter.
2. The system checks to see if you have permission to execute privileged operations. If you do not, you will see the following message:

```
You (<logname>) do not have permission to
perform software removal.
Please consult your Operations/System
Administration Guide for more
information on assigning permissions
to privileged operations.
```

Refer to Chapter 4, System Administration.

3. The system also checks to see if you are at the console. If not, you will see the following message:

```
You must be on the console to run removepkg.
```

The procedure terminates.

4. If you entered the **removepkg** command without arguments, you will see a sorted listing of installed packages.

NOTE — You may halt and continue the display using Ctrl s and Ctrl q respectively.

You will see the following message:

```
Select a number (1-n) from this list to remove:
```

NOTE — n is the number of packages currently installed.

Type the number of the package that you wish to remove and strike Enter.

5. If you correctly entered the **removepkg** command, and the system located the package that you specified for removal, you will see the following message:

```
Do you really want to remove
<package name>?

Strike ENTER to continue
or ESC to stop.
```

If the above message did not occur, go directly ahead to Step 9 to resolve the problem.

Strike Enter to remove the package.

If you decided not to remove this package or specified the wrong one, then strike the Esc key, you will see the following message:

```
<package name> not removed.
```

NOTE: If you strike anything other than Enter or Esc, the system will beep.

6. If the system detects that the package is in the "sysadm" (AT&T 3B2 Computer simple administration) format, you should follow the prompt messages that will appear on your screen to complete the removal of the package.

7. If the system did not find a valid removal script for the package, you will see the following message:

```
Cannot find removal script for
<package name>. You will have
to remove this package manually
using UNIX System tools from
the UNIX System Shell.

The file /usr/lib/installed/Files/<installed filename>
contains a list of the files and
directories installed or created
by the package. You may wish to
use this file to help in removing
the package.
```

NOTE

The <installed filename> will be some filename of the form: "<package>.name".

8. If the package is removed successfully, you will see the following message:

```
The <package name> package is now removed.
```

If the removal is unsuccessful, the removal process is halted.

The remaining Steps 9 and 10 do not apply.

9. If the system found no optional add-on packages installed on your system, you will see the following message:

```
There are currently no software
applications installed that can be
removed.
```

10. If the system did not find the package name you supplied as an argument to **removepkg**, you will see the following message:

```
There is no software package
currently installed resembling:
<argument(s)>.
```

Using the UNIX System Shell

Using the UNIX System Shell

Chapter 3: Using the *UNIX* System Shell

Commands

Since the target user is an experienced UNIX system user who is a programmer, you can skip this chapter. However, if you happen to be a novice UNIX system user, read this chapter and Chapter 7, Shell Tutorial, in the *UNIX System User's Guide*. This guide is not provided with your documentation set. See the *Documentation Roadmap* for ordering information.

The UNIX system shell serves as the interface between you and the computer. The shell accepts commands from you. In the UNIX system, a command (executable program) is a program that can be executed by the computer without a need for translation. Commands or requests to the shell are usually entered as a single line typed on the keyboard. This single line is called a command line. A command line is divided into two major parts:

- The program name
- The arguments.

The first word of the command line is the name of the program to be executed. The other words on the line are referred to as arguments. Arguments are used to provide information required by the program. The command line looks like this:

command *argument argument argument* ... ... [Enter].

Entering Commands

After receiving the prompt #, you can type in your command line and strike Enter. When the shell prompt returns, the program is finished running, and you can enter another command line.

UNIX system commands are specified in lowercase letters. Enter all these commands in lowercase letters. If a command is entered in uppercase letters, the computer will not recognize it.

The files you create to execute at a later time are called executable files. These filenames can have uppercase or lowercase letters. When you invoke an executable file that includes uppercase letters, you should always type the filename just the way it appears.

If you make a mistake while typing in a command, just backspace (strike the Backspace key) to the beginning of the mistake and retype.

If you log in using uppercase letters, the system assumes that your terminal (console included) is not capable of handling lowercase. All input and output for the remainder of the session are expected to be uppercase. If this happens, log off and make sure your Caps/Lock key is not engaged. Then log in again.

Using Control Characters

The control key is used in combination with other keyboard characters. These keys are used to initiate a controlling action such as backspacing or tabbing across a line. In addition, some control characters define UNIX system specific commands, such as temporarily halting output from displaying on your screen.

The control key on your keyboard is labeled [Ctrl].

You can type a control character by holding down the [Ctrl] key and then striking the appropriate alphabetic key. Control characters do not print on the screen when typed. In this guide, if you're instructed to strike Control s, the text will read "strike [Ctrl] [s].

Let's take a look at some of the control character combinations you'll be using regularly when working with the UNIX system.

Stopping a Command

If you want to stop a command from executing on your console, simply strike Del or Rubout. You will receive the UNIX system prompt, indicating that the UNIX system is ready to accept your next command. You can do this before or after a command has started to execute.

Temporarily Stopping Output

At times, you may wish to temporarily stop the UNIX system from displaying output on your display screen. This could be the case when information is scrolling freely across your screen. If you type Ctrl and s simultaneously, the information on the screen will stop scrolling. When you type Ctrl and q simultaneously, the information will again start to scroll freely across the screen.

Running a Command in the Background

Within the UNIX system, you can run two or more programs or commands at one time. You can run commands or programs in the foreground (where you can see it on the screen) and in the background (where you can't see it on the screen). When a command is running, it is known as a process. Therefore, a foreground command is called a foreground process. An example of a foreground process is as follows:

```
spell filename [Enter]
```

The **spell** command will display on your screen any misspelled words in the file. If there are no misspellings, there will be no output and the prompt will be returned.

You can run a command in the background (background process) by adding an ampersand (&) to the end of a command line before striking [Enter].

When the shell reads the **&**, the shell starts running the command in the background, displays an identification number, and displays the # prompt so you can continue to work in the foreground.

To save the output of the process you are running in the background, you must redirect the output into another file. You can look at the file later. Redirecting command output puts the results into a specified file so you can look at the file later. For example, if you want to run the **spell** command on a large file, but want to look at it later, type the following:

```
spell filename > newfile & [Enter]
```

The shell first gives you an identification number and then the UNIX system window prompt. Then you can go ahead and run another command. When your process in the background is finished, the output will be contained in the specified file.

When a process is running in the background, it cannot be temporarily stopped with [Ctrl] [s] or halted with [Del] or [Rubout] like foreground processes. Only when you log off or terminate the background process will it stop. To stop the background process while you're still logged in, type `kill` *id_number* [Enter], where *id_number* is the identification number of the process.

To see if your process is running in the background, use the **ps** (report process status) command. After you type a **ps**, a list of the processes that are currently running is displayed. Look through these processes and find the identification number that was assigned to your background process. If you find it, the process is still running. If you do not find it, then the process has completed running.

For example, suppose you run **spell** in the background and your id number is 29570. Type the **ps** command to see if the process is running. The output will look something like the following:

```
  PID TTY      TIME COMMAND

29047 sxt002   0:03 sh
29570 sxt002   0:00 spell
29572 sxt002   0:00 tee
29573 sxt002   0:00 sh
29575 sxt002   0:03 sort
29577 sxt002   0:00 spellpro
29578 sxt002   0:02 sed
29579 sxt002   0:00 comm
29580 sxt002   0:01 ps
```

Notice the second line from the top matches your id number. That means your background process is still running. See the **ps**(1) command in the *User's/System Administrator's Reference Manual* for additional information.

Printing the Console Screen

Did you ever want a printout of the display on your terminal screen, but did not want to go to the trouble of printing out your whole file? Well, you do not have to print out the entire file. By striking two keys on the keyboard, you can have only the information on your computer screen printed for you. Strike (Shift) and the (Print/Scrn) key on your keyboard, and the display will be printed for you.

If the printer is busy, your print request will be queued like any other print request. To see if the printer is clear, use the **lpstat** command. This will give you a status of the print jobs that are queued to be printed. Type the following:

```
lpstat (Enter)
```

See the **lpstat**(1) command in the *User's/System Administrator's Reference Manual* for additional information.

Printing a File

To print a file, you first need a printer connected to your computer. The *User's Guide* (999-300-395) provides directions and configuration information (switches and cables) for connecting several printers.

```
lp filename [Enter]
```

The *filename* is the file you want to print. As mentioned before, use the **lpstat** command to see the print jobs queued on the printer. Refer to Chapter 4, System Administration, for information on setting up your printer, and Chapter 7, Line Printer Spooler Administration, for information on printer administration.

Stopping a Session With Your Computer

To complete a session, log off: strike Ctrl and d at the same time or type **`exit`** and strike Enter. Either one of these commands will log you off the UNIX system.

An Introduction to UNIX System Commands

The UNIX system software you receive with your computer is called the Foundation Set. This software set contains the files necessary to provide the UNIX system on your computer, including many useful utilities (programs) and commands.

A program is a set of instructions that the computer follows to do a specific job. In the UNIX system, programs that can be executed by the computer without the need for translation are called executable programs or commands.

These commands allow you to:

- process text,
- manage information,
- communicate electronically, and
- use a productive programming and software development environment.

When you first log in to your computer, you will receive a shell prompt. The shell prompt is a signal from the shell command interpreter that it is ready to accept your request. The command you wish to execute is typed in on the keyboard followed by striking the Enter key. When the shell receives the Enter, it considers the input (whatever it is) as a command. The shell searches one or more directories to locate the program you specified. When the program is found, the shell brings your request to the attention of the kernel. The kernel then follows the program's instructions and executes your request. After the program runs, the shell asks you for more information or tells you it is ready for your next command.

For an explanation of all the commands included with your computer, refer to the *User's/System Administrator's Reference Manual* and the *Programmer's Reference Manual.* Also, refer to the *UNIX System V User Guide* for more information about UNIX system commands. The User Guide in not included in your documentation set. To obtain a copy, consult your Documentation Roadmap for ordering information.

Basic File Operations

Your Foundation Set includes many commands. Some of the more common commands used for file and directory manipulation are included here.

- **cat** (concatenate): Will display the contents of one or more files on your screen.
- **cd** (change directory): Will change your current working directory to the directory you specify. If a directory is not specified, your current working directory will be your home directory.
- **chmod** (change mode): Will change the permission modes of a file or directory. See "Access Permissions" in Chapter 5, Customizing Your Computer, for a discussion of permission modes.
- **cp** (copy): Will copy the contents of a file to another specified file.
- **ls** (list): Will list the contents of one or more files or directories.
- **mkdir** (make directory): Will create a directory of a specified name.
- **mv** (move): Will move (rename) a file or directory.
- **rm** (remove file): Will remove a specified file or files.
- **rmdir** (remove directory): Will remove a specified directory or directories.

System Maintenance Commands

This group of commands is used for system administration. The following is a list of some of the more common system maintenance commands.

- **df** (report number of free disk blocks): Will display the number of free disk blocks and free i-nodes available for on-line file systems.
- **du** (summarize disk usage): Will display the number of blocks contained in all files and directories.
- **fsck** (file system check): Will run a consistency check on a specified file system.
- **mount**: Will mount a file system.
- **ps** (report process status): Will display all active processes that are running on the UNIX system.
- **umount**: Will unmount a file system.
- **uname** (display current UNIX system): Will display the node name of the current UNIX system when used with the *-n* option. Refer to Chapter 4, System Administration, in the "Setting Up Mail" section for information on naming your computer.

Text and Text Editing Commands

This group of commands is used to edit and manipulate text. The following is a list of these text and text editing commands:

- **vi**: Visual screen editor.
- **ed**: Text editor.
- **diff** (file comparator): Will run a file comparison on two files and display the lines of text that are different.
- **grep** (search for a pattern): Will search for a pattern of text in one or more files.
- **spell**: Will run a spelling check on the specified files.

File Encryption Commands

This group of commands is used in protecting files. These encryption commands are packaged separately for use in the United States only. You'll have to install them after the initial UNIX system installation with the procedures given in Chapter 4, System Administration, under the "Installing UNIX System Applications Software From Floppy Disk" section. Refer to Chapter 5, Customizing Your Computer, to learn how to use the **crypt** command.

- **crypt**: Is used to encrypt a file.
- **vi -x**: Is used to edit a file (with the visual screen-editor), where that file has already been encrypted. This is also used to create an encrypted file.
- **ed -x**: Is used to edit a file (with the text editor), where that file has already been encrypted. This is also used to create an encrypted file.

Basic Networking Commands

This group of commands allows you to use Basic Networking on your computer. Refer to Chapter 8, Basic Networking Administration, for a more detailed description of Basic Networking. The following is a list of some of the more common Basic Networking commands.

- **cu** (call another UNIX system): Will call up another UNIX system or terminal.
- **mail** (send or read mail): Will send mail to someone on your UNIX system or another UNIX system and allow you to read mail sent by someone else.
- **uuto** (send files): Will send files to a remote computer and place the files in **PUBDIR/receive/***user***/***mysystem*, where **PUBDIR** is defined as **/usr/spool/uucppublic**.
- **uupick** (retrieve files): Will "pick up" files that are sent to a computer using **uuto**.
- **uux** (UNIX system-to-UNIX system command execution): Will gather necessary files from a UNIX system, execute the command on a specified system, and send the output to a file on a specified system.

Line Printer Spooler Commands

This group of commands is used for printer spooling on your computer. Refer to Chapter 7, Line Printer Spooler Administration, for a more detailed description of printer spooling. The following is a list of some of the more common printer spooling commands.

- **lp** (line printer): Will arrange your file to be printed on a specified printer.
- **cancel**: Will stop a print job from printing and remove that job from the printer queue when the print job id is entered.
- **enable**: Will restart a printer if it was stopped with the **disable** command. All print jobs in the printer queue will start printing again, in turn.
- **lpstat** (lp status information): Will display the information about the status of the LP line printer system.

These functions are also available in the AT&T Administration main menu.

What is a Manual Page

Manual pages describe computer commands. Each command has independently numbered manual pages describing that command in detail. The manual pages are grouped into sections; each section contains the commands for a specific function as follows:

- Section 1 — System Commands
- Section 1M — System Maintenance commands
- Section 2 — System Calls
- Section 3 — Subroutines
- Section 4 — File Formats
- Section 5 — Miscellaneous Facilities
- Section 7 — Special Files
- Section 8 — System Maintenance Procedures.

Sections 1, 1M, 7, and 8 are in the *User's/System Administrator's Reference Manual.* The *Programmer's Reference Manual* contains sections 2, 3, 4, and 5.

NOTE

For convienience, several manual pages appear in more than one manual.

System Administration

System Administration

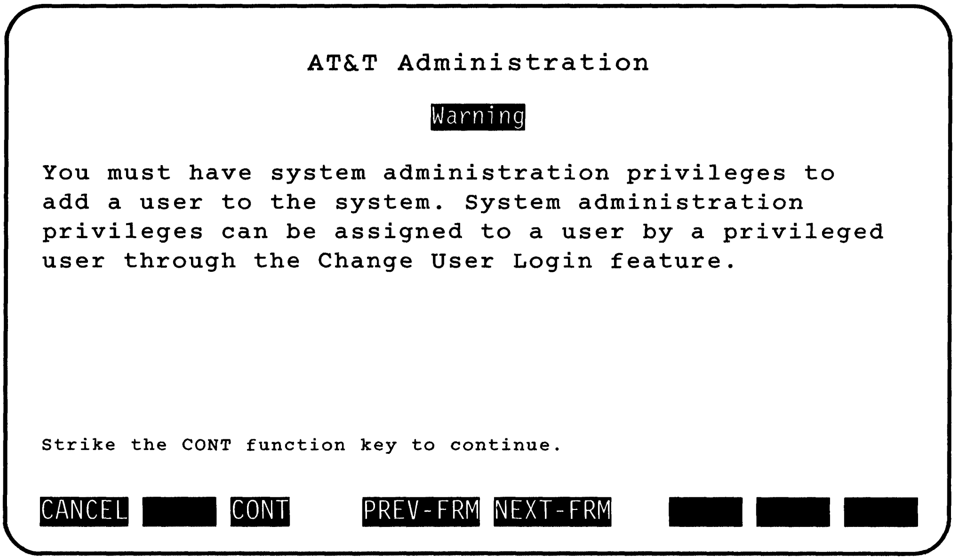
Chapter 4: System Administration

Introduction to System Administration

System Administration Privileges

Some system functions require that you have special system administration privileges. If you try to use a function that requires special system administration privileges, a validation check occurs that compares your login id against a file of "allowed" users. If you have system administration privileges, the function you are doing continues. If you do not have system administration privileges, you will receive a message. This type of message is not shown in this chapter to reduce redundancy. Your computer will display the appropriate message.

A typical warning message follows:



```
                    AT&T Administration

                          Warning

You must have system administration privileges to
add a user to the system. System administration
privileges can be assigned to a user by a privileged
user through the Change User Login feature.




Strike the CONT function key to continue.
```

Accessing The Interface

The tasks described in this chapter assume that you know how to log on, access AT&T Administration, make selections, and navigate within the interface. If you do not know how to do this, read Chapter 1, Introduction. The tasks begin at some point within AT&T Administration to prevent redundancy and assume that you know how to get there.

Backup to Removable Media

Backing up your system means making a floppy disk or cartridge tape copy of the files you have on the computer Hard Disk. Backing up system software on a regular basis is a good safety precaution against an unexpected system failure or operator error.

Backup is an important part of regular maintenance. Each time you update the software, back up the files. If you accidentally delete a file or if a system failure occurs that corrupts important files, you can restore your files using the latest system or incremental backup set of floppy disks.

You can back up all files (system and user), or only the files updated since the last backup, or an individual user's files, or selected files/directories.

When you back up to floppy, the system tells you the approximate number of formatted floppy disks you need and the approximate time it will take to do the backup. You can use 1.2 MB or 360 KB, double-sided, floppy disks.

The Backup History function can be used to determine when the last system backup and the last incremental backups were done or if they were ever done.

The ways to back up your computer are listed below:

- **Personal Backup**
  All files in the user's HOME directory tree are copied to the removable media.
- **Selective Personal Backup**
  You can use this function to specify file names (directories or regular files) that are located in the user's HOME directory to back up.
- **System Backup**
  The system backup requires system administration privileges. All system and user files that have been modified or created since the system was installed are backed up to the removable medium. The installation date is the date of the last file that was installed at installation time. The search for files to back up starts from "/" (the root file system) and includes all mounted file systems.

  Currently, the contents of certain system directories (like */usr/bin*) are not normally backed up. If you desire to back up one of these directories, use the **touch** command on the directory contents, then they will be backed up to the removable medium even though they have not been modified or created since the system was installed. If you avoid using **touch** on these directories, then only system files that are modified after system installation, such as */etc/passwd,* are backed up and later restored.

When copying to floppy, several formatted floppy disks may be required, depending on the amount of information you have on your system.

You should do a system backup on a regular basis, such as every month. You should also do a system backup to preserve large file system changes, such as any time you update the system software.

WARNING

The System Backup function does not back up a complete image of your entire file system. Only the system and user files that have been modified or created since the system was last installed are backed up.

NOTE

Notice that only copying the files that have been modified or created since installation prevents overwriting the contents of system directories like */usr/bin* or */etc/bin* because the contents of these directories (especially the commands) are not normally modified after system installation. This means that if the contents of these system directories are lost, the lost files will have to be recovered from the system installation floppies.

- **Incremental System Backup**
  The incremental system backup requires system administration privileges. All the system and user files that have been modified or created since the last system or incremental backup will be copied to the removable media.

- **System Backup of Users**
  User files will be backed up to the removable media. You have the option to back up all users' files or back up one or more selected users' files.

- **Selective System Backup**
  Regular files or the contents of directories may be backed up. The path name specifiled may be anywhere on a mounted file system.

NOTE

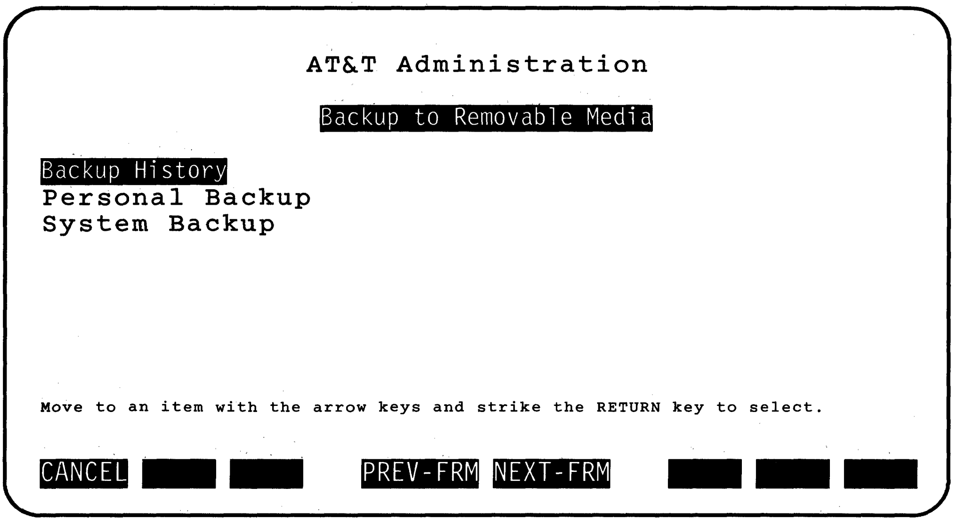
Since any file(s) may be specified (including system directories like */bin*), on restore, you will be asked if you want to overwrite the existing file. The default is no because, by default, a file on disk with a date newer than the restore file will not be overwritten. Again, this is a safeguard to prevent you from getting in trouble while doing a restore after a system upgrade.

Backup History

You should follow a regular schedule for performing backup. When you back up your computer, a record of the date and time is made. The Backup History form will display the last date and time you performed a system or incremental backup.

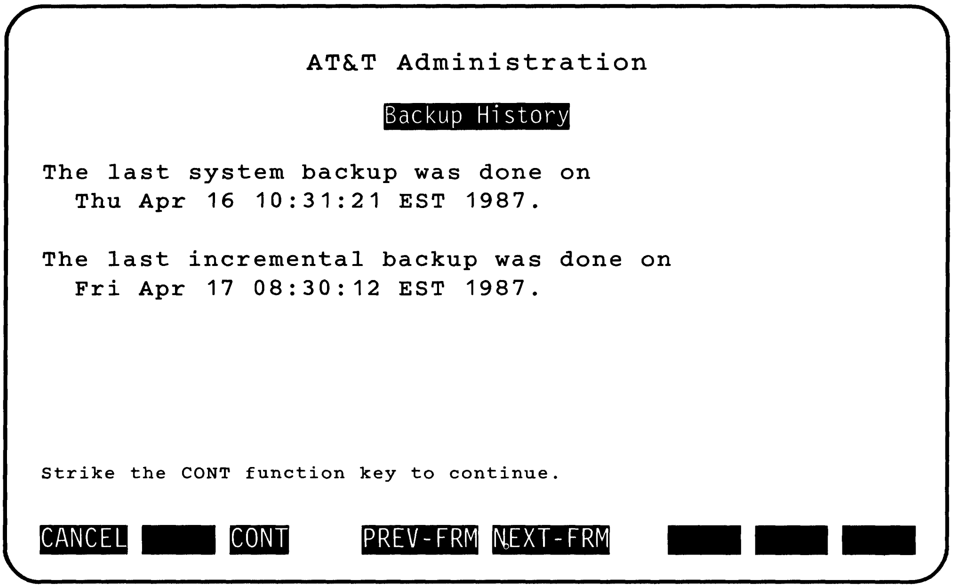
To check the Backup History, use the following procedure.

1. From the Administration menu, highlight `Backup to Removable Media` and strike Enter. The Backup to Removable Media menu appears as follows:



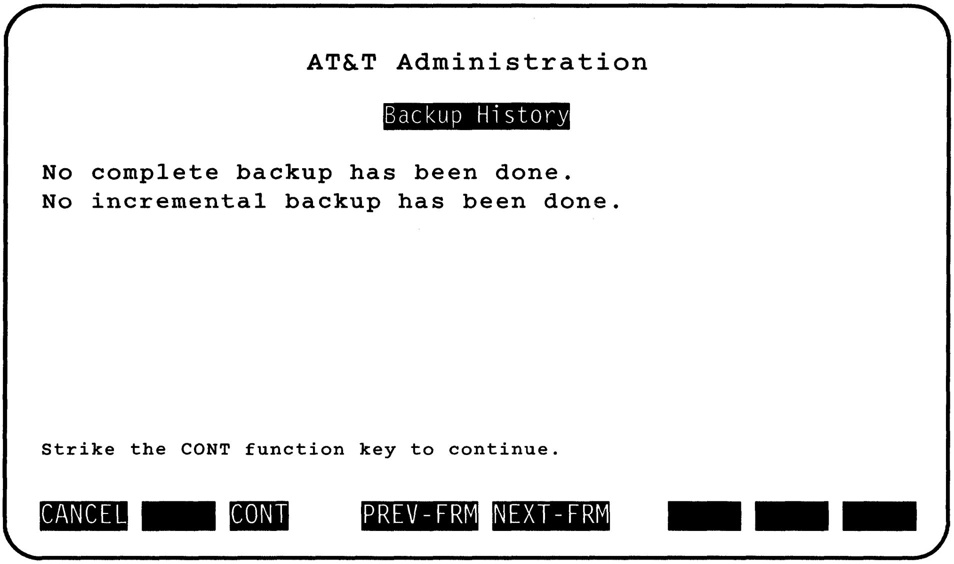
```
                    AT&T Administration

                 Backup to Removable Media

Backup History
Personal Backup
System Backup




Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL                  PREV-FRM NEXT-FRM
```

2. Highlight Backup History from the Backup to Removable Media menu and strike Enter. The Backup History form appears as follows:



```
                AT&T Administration

                   Backup History

The last system backup was done on
  Thu Apr 16 10:31:21 EST 1987.

The last incremental backup was done on
  Fri Apr 17 08:30:12 EST 1987.




Strike the CONT function key to continue.
```

The date and time of the last system and/or incremental backups were done are displayed in the Backup History form. If you never did a system or incremetal backup, the Backup History form would appear as follows:



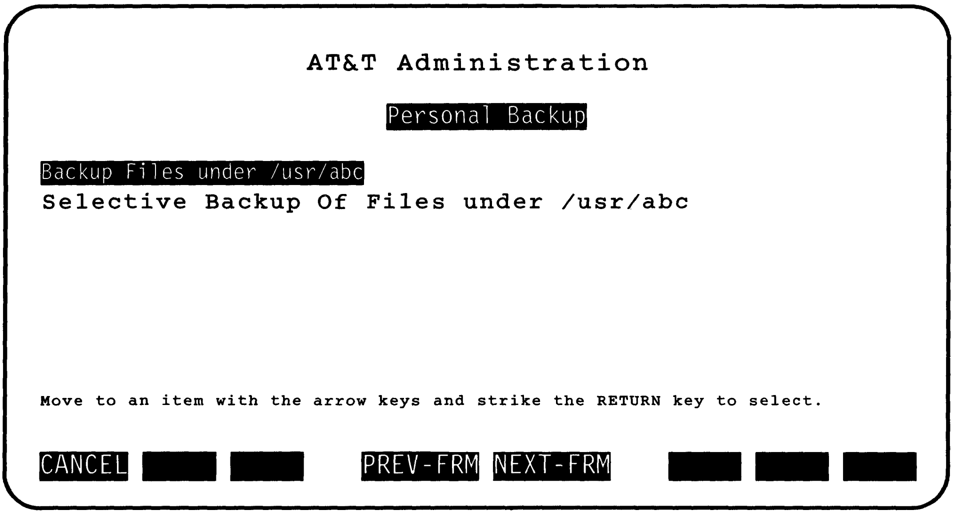
```
                    AT&T Administration

                       Backup History

No complete backup has been done.
No incremental backup has been done.







Strike the CONT function key to continue.
```

3. Strike CANCEL to close this frame and make the Backup to Removable Media menu active.

Personal Backup

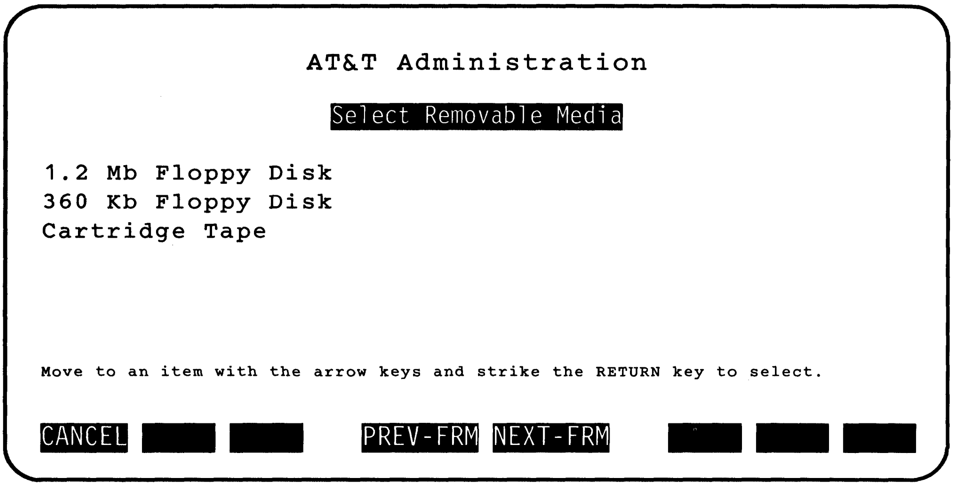
To copy all files in your HOME directory to the removable media, use the Backup Files under<user's home directory> function where <user's home directory> is the name of your HOME directory. You cannot backup other user's files using Personal Backup. The following example assumes your HOME directory is */usr/abc*.

1. From the Administration menu, highlight Backup to Removable Media and strike Enter. The Backup to Removable Media menu appears as shown in the previous screen.
2. From the Backup to Removable Media menu, highlight Personal Backup and strike Enter. The Personal Backup menu appears as follows:



```
                    AT&T Administration

                      Personal Backup

Backup Files under /usr/abc
Selective Backup Of Files under /usr/abc




Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL                  PREV-FRM NEXT-FRM
```

3. From the Personal Backup menu, highlight Backup Files under /usr/abc and strike Enter. The Select Removable Media menu appears as follows:



```
                    AT&T Administration

                    Select Removable Media

1.2 Mb Floppy Disk
360 Kb Floppy Disk
Cartridge Tape




Move to an item with the arrow keys and strike the RETURN key to select.
```

4. Depending on what options are available on your computer and what your needs are, select the appropriate media for storing your data by highlighting one of the three choices (1.2 MB floppy disk, 360 KB floppy disk, or cartridge tape) and striking Enter. Once you have selected the media, you will receive the following message:

   Computing the number of files to be backed up. Please wait.

   Your computer will estimate the number of floppies (1.2MB and 360 KB) or tapes needed to fit all the files to be backed up and how much time the backup will take. Then, you will follow the instructions from your computer to insert and remove the previously formatted floppy disks or tapes and how to number them in sequence. The following is an example of this.

```
The backup will need:

    Approximately 3 formatted 1.2BM floppy disks.

The backup will take 2 minutes, approximately.

The floppy disks used for the backup MUST be
formatted.  Be sure to number the floppies
consecutively in the order they will be inserted.

Please insert the first floppy and
strike RETURN to continue.
```

5. Insert a blank, formatted floppy disk or a cartridge tape and strike Enter.

NOTE

Tapes do not have to be formatted.

The floppy disks used to back up your computer must be formatted in UNIX system format. (See "Format 1.2 MB UNIX System Floppy Disk" in this chapter.)

Once the backup is in progress, you will receive the following instruction:

```
Backup in progress. Do not remove the floppy/tape.
```

6. In addition, if the backup spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

```
You may remove floppy number 1.
To exit, please press 'q' followed by RETURN.

To continue, insert floppy number 2
and strike the RETURN key.
```

If you press 'q' to exit, the following message is displayed:

```
You have canceled the Backup to Removable Media.
```

NOTE

As you remove each floppy disk or tape, attach a label containing subject, date, and the number of the floppy disk or tape. File the floppy disk in its envelope. If you write on a label already attached to the floppy disk, only use a felt-tip or nylon-tip pen. Do not use a ball-point pen to write on a label already attached; this can cause damage to the floppy disk.

7. Repeat inserting, removing, and labeling until a message appears indicating the backup is complete as follows:

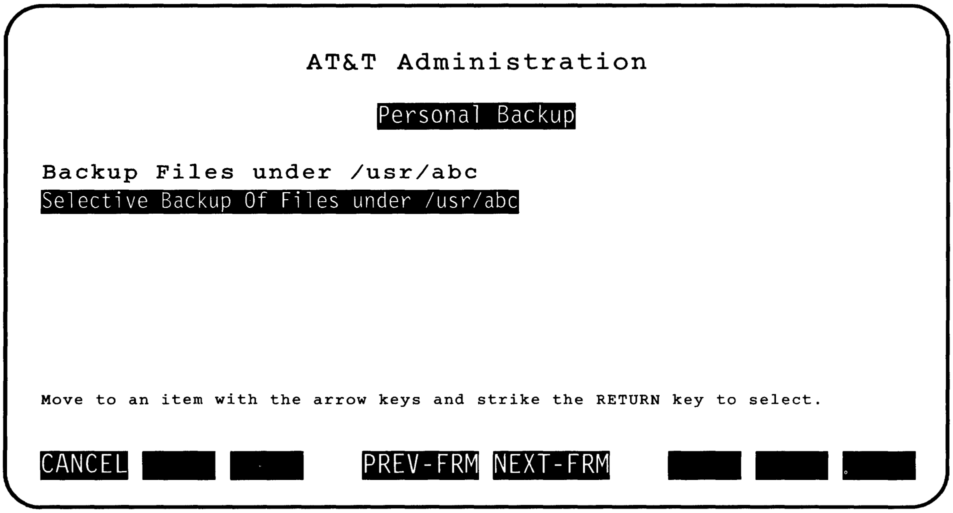
   ```
   Backup is now done. You may remove the floppy.
   ```

8. Remove the last floppy disk when the system informs you that it has completed the backup.

Selective Personal Backup

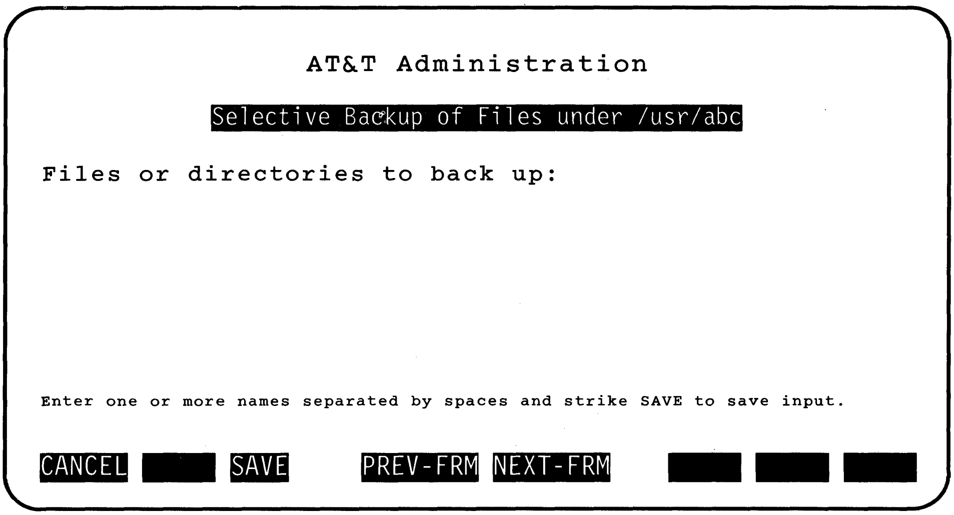
You may back up selective files and directories under your HOME directory. The following procedure assumes your HOME directory is */usr/abc.* You cannot back up files located outside of your HOME directory using this function.

1. From the Backup to Removable Media, highlight Personal Backup and strike Enter. The Personal Backup menu appears as follows:



```
                    AT&T Administration

                      Personal Backup

Backup Files under /usr/abc
Selective Backup Of Files under /usr/abc




Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL                PREV-FRM NEXT-FRM
```

2. From the Personal Backup menu, highlight Selective Backup of Files under /usr/abc and strike Enter. The following frame appears.



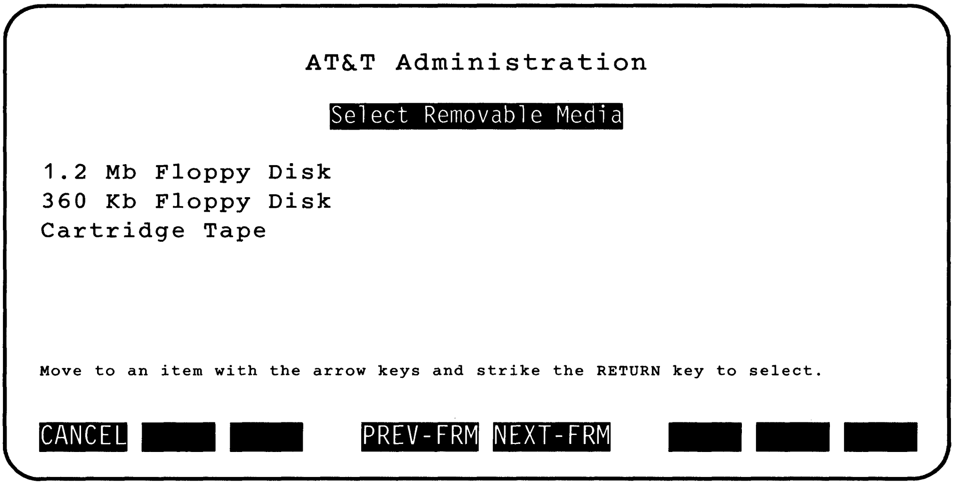
AT&T Administration

Selective Backup of Files under /usr/abc

Files or directories to back up:

Enter one or more names separated by spaces and strike SAVE to save input.

3. Enter the file names or directories to be copied (Shell metacharacters can be used for the file or directory names.) and strike SAVE. The Select Removable Media menu appears as follows:



AT&T Administration

Select Removable Media

Move to an item with the arrow keys and strike the RETURN key to select.

4. Depending, on what options are available on your computer and what your needs are, select the appropriate media for storing your file(s) by highlighting one of the three choices (1.2MB floppy disk, 360 KB floppy disk, or Cartridge Tape) and striking Enter. Once you have selected the media, you will receive the following message:

   ```
   Computing the number of files to be backed up. Please wait.
   ```

   Your computer will estimate the number of floppies (1.2MB and 360 KB) or tapes needed to fit all the files to be backed up and how much time the backup will take. Then, you will follow the instructions from your computer to insert and remove the previously formatted floppy disks or tapes and how to number them in sequence.

5. Insert a blank, formatted floppy disk or a cartridge tape and strike Enter.

   NOTE

   The floppy disks used to back up your computer must be formatted in UNIX system format. (See "Format 1.2 MB UNIX System Floppy Disk" in this chapter.)

   Once the backup is in progress, you will receive the following instruction:

   ```
   Backup in progress. Do not remove the floppy/tape.
   ```

6. In addition, if the backup spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

   ```
   You may remove floppy number 1.
   To exit, please press 'q' followed by RETURN.

   To continue, insert floppy number 2
   and strike the RETURN key.
   ```

   If you press 'q' to exit, the following message is displayed:

   ```
   You have canceled the Backup to Removable Media.
   ```

NOTE As you remove each floppy disk or tape, attach a label containing subject, date, and the number of the floppy disk or tape. File the floppy disk in its envelope. If you write on a label already attached to the floppy disk, only use a felt-tip or nylon-tip pen. Do not use a ball-point pen to write on a label already attached; this can cause damage to the floppy disk.

7. Repeat inserting, removing, and labeling until a message appears indicating the backup is complete as follows:

   ```
   Backup is now done. You may remove the floppy.
   ```

8. Remove the last floppy disk when the system informs you that it has completed the backup.

Backup Users

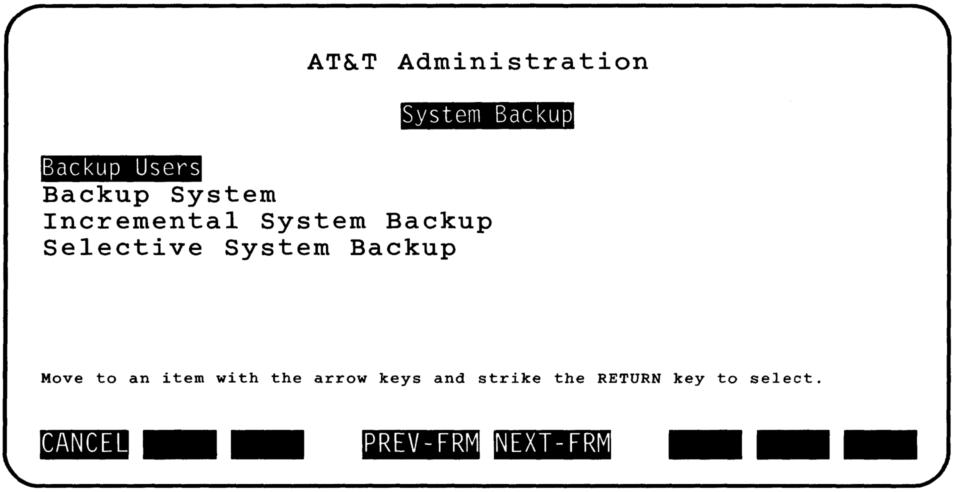
The backup users function allows you to back up the users' files in the user's HOME directory to the removable media. You have the option to:

- Back up all users
- Back up one or more selected users.

You must have system administration privileges to use this function. Only user logins are allowed to be backed up.

Use the following procedure to back up user's files.

1. Highlight System Backup from the Backup to Removable Media menu and strike Enter. The System Backup form appears as follows:

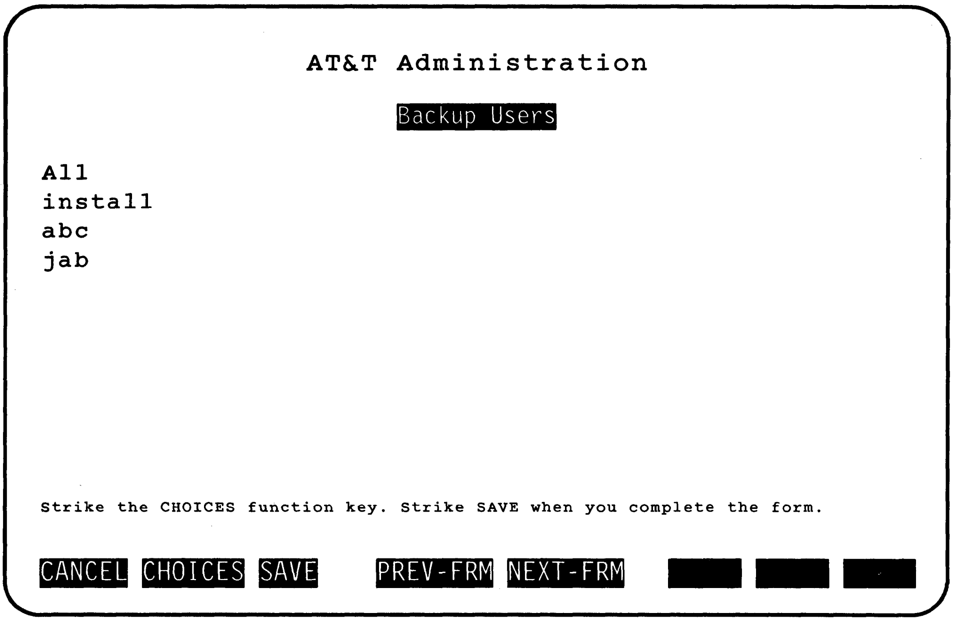


AT&T Administration

System Backup

Move to an item with the arrow keys and strike the RETURN key to select.

2. From the System Backup menu, highlight Backup Users. Assume the user logins **abc** and **jab** have been added to your system. An example Backup Users form would appear as follows:



AT&T Administration

Backup Users

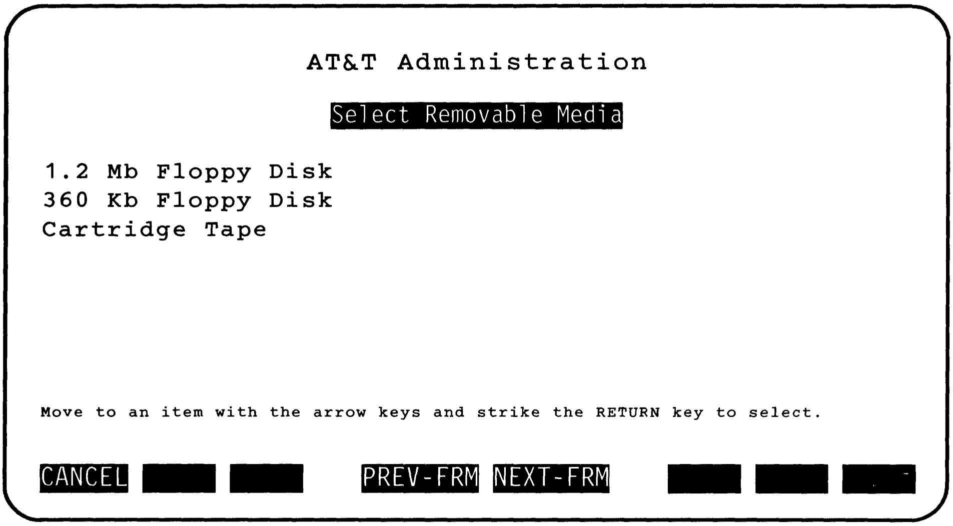
```
All
install
abc
jab
```

Strike the CHOICES function key. Strike SAVE when you complete the form.

3. You may need to strike CHOICES to display the pop-up menu of login names. If you do not, you will receive a list of user login names automatically. Choose a login name from the list of login names. Move to an item with the arrow keys and strike the MARK function key to select login names. When you are finished marking the user login names, strike Enter. You can also select by typing the login name with the cursor resting on the "User's login name:" field. If you enter an invalid login name, you will receive the following error message:

   ```
   <User's login name> is not a valid login name.
   ```

4. When you have finished selecting the user's login names, strike SAVE. The Select Removable Media menu appears as follows:



```
                    AT&T Administration

                   Select Removable Media

1.2 Mb Floppy Disk
360 Kb Floppy Disk
Cartridge Tape






Move to an item with the arrow keys and strike the RETURN key to select.
```

5. Depending on what options are available on your computer and what your needs are, select the appropriate media for storing your file(s) by highlighting one of the three choices (1.2MB floppy disk, 360 KB floppy disk, or Cartridge Tape) and striking Enter. Once you have selected the media, you will receive the following message:

   Computing the number of files to be backed up. Please wait.

   Your computer will estimate the number of floppies (1.2MB and 360 KB) or tapes needed to fit all the files to be backed up and how much time the backup will take. Then, you will follow the instructions from your computer to insert and remove the previously formatted floppy disks or tapes and how to number them in sequence.

6. Insert a blank, formatted floppy disk or a cartridge tape and strike Enter.

NOTE	The floppy disks used to back up your computer must be formatted in UNIX system format. (See "Format 1.2 MB UNIX System Floppy Disk" in this chapter.)

Once the backup is in progress, you will receive the following instruction:

```
Backup in progress. Do not remove the floppy/tape.
```

7. In addition, if the backup spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

```
You may remove floppy number 1.
To exit, please press 'q' followed by RETURN.

To continue, insert floppy number 2
and strike the RETURN key.
```

If you press 'q' to exit, the following message is displayed:

```
You have canceled the Backup to Removable Media.
```

NOTE

As you remove each floppy disk or tape, attach a label containing subject, date, and the number of the floppy disk or tape. File the floppy disk in its envelope. If you write on a label already attached to the floppy disk, only use a felt-tip or nylon-tip pen. Do not use a ball-point pen to write on a label already attached; this can cause damage to the floppy disk.

8. Repeat inserting, removing, and labeling until a message appears indicating the backup is complete as follows:

```
Backup is now done. You may remove the floppy.
```

9. Remove the last floppy disk when the system informs you that it has completed the backup.

Backup System

The system backup function backs up all system and user files (in all mounted file systems) that have been modified or created since the system was installed.

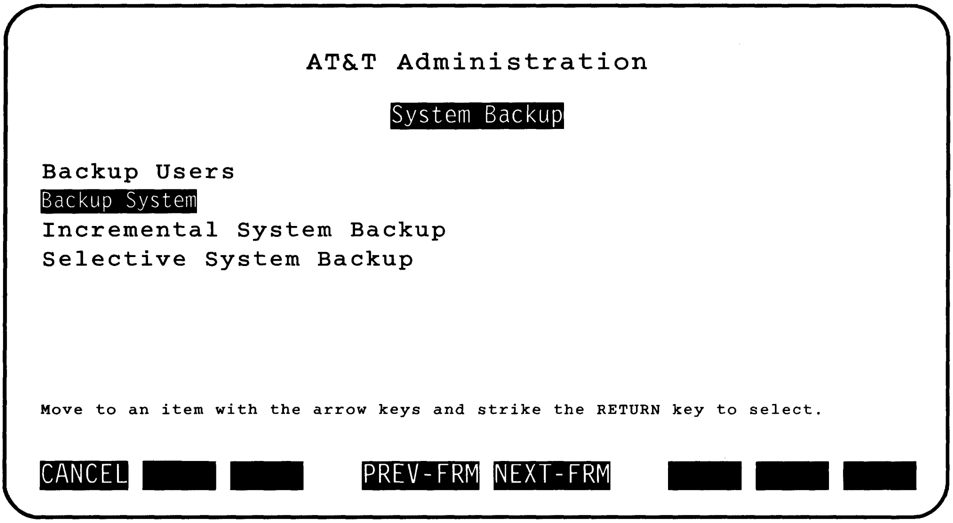
WARNING

The System Backup function does not back up a complete image of your entire file system. Only the system and user files that have been modified or created since the system was last installed are backed up.

You must have system administration privileges to use this function.

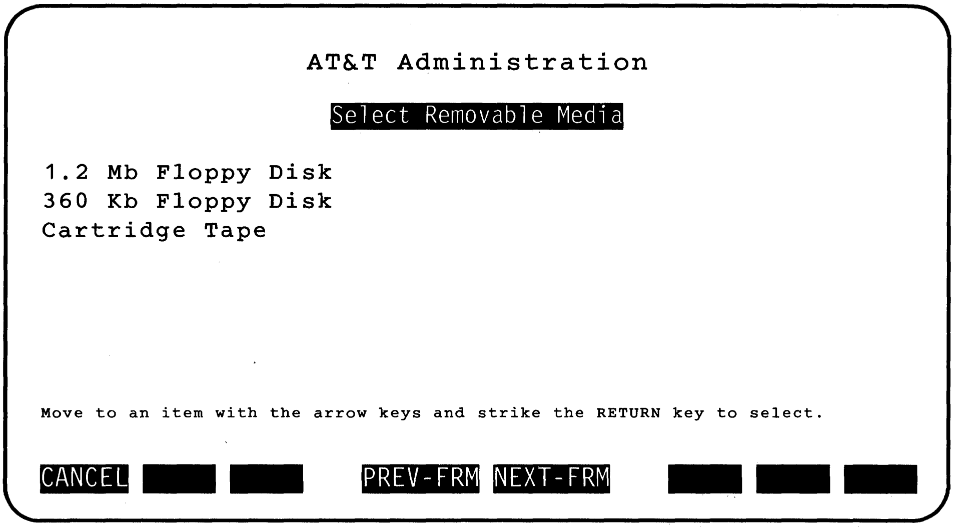
Use the following procedure to do a system backup.

1. Highlight System Backup from the Backup to Removable Media menu and strike Enter. The System Backup form appears as follows:



```
                    AT&T Administration

                       System Backup

Backup Users
Backup System
Incremental System Backup
Selective System Backup




Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL                PREV-FRM NEXT-FRM
```

2. From the System Backup menu, highlight Backup System The Select Removable Media menu appears as follows:



```
                    AT&T Administration

                   Select Removable Media

1.2 Mb Floppy Disk
360 Kb Floppy Disk
Cartridge Tape




Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL                 PREV-FRM NEXT-FRM
```

3. Depending on what options are available on your computer and what your needs are, select the appropriate media for storing your file(s) by highlighting one of the three choices (1.2MB floppy disk, 360 KB floppy disk, or Cartridge Tape) and striking Enter. Once you have selected the media, you will receive the following message:

   Computing the number of files to be backed up. Please wait.

   Your computer will estimate the number of floppies (1.2MB and 360 KB) or tapes needed to fit all the files to be backed up and how much time the backup will take. Then, you will follow the instructions from your computer to insert and remove the previously formatted floppy disks or tapes and how to number them in sequence.

4. Once the backup is in progress, you will receive the following instruction:

   ```
   Backup in progress. Do not remove the floppy/tape.
   ```

5. In addition, if the backup spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

```
You may remove floppy number 1.
To exit, please press 'q' followed by RETURN.

To continue, insert floppy number 2
and strike the RETURN key.
```

If you press 'q' to exit, the following message is displayed:

```
You have canceled the Backup to Removable Media.
```

6. When the backup is complete you will receive the following instructions:

```
Backup is now done. You may remove the floppy.
```

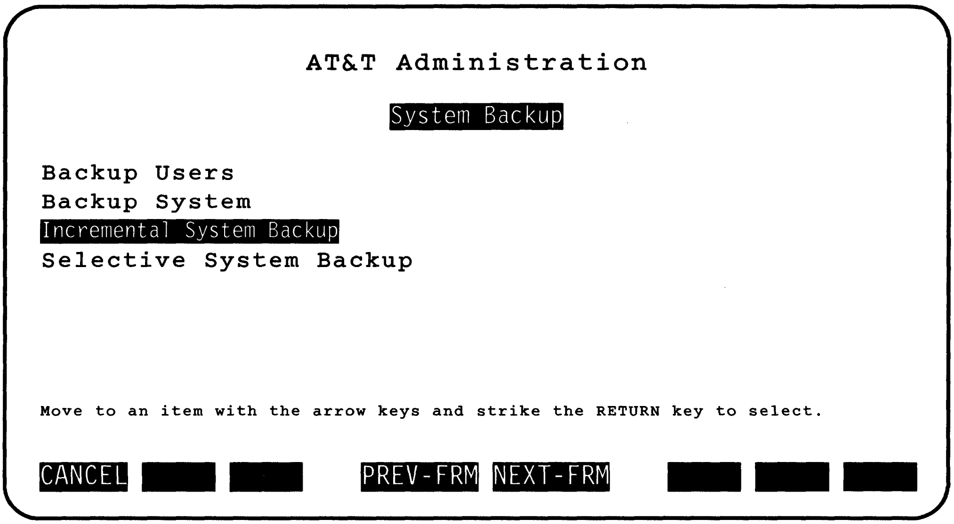
Incremental System Backup

The incremental system backup function backs up all files in all mounted file systems that have been modified or created since the last system or incremental backup. An incremental system backup differs from a system backup because a system backup backs up all files in all mounted file systems that have been modified or created since the system was installed.

You must have system administration privileges to use this function.

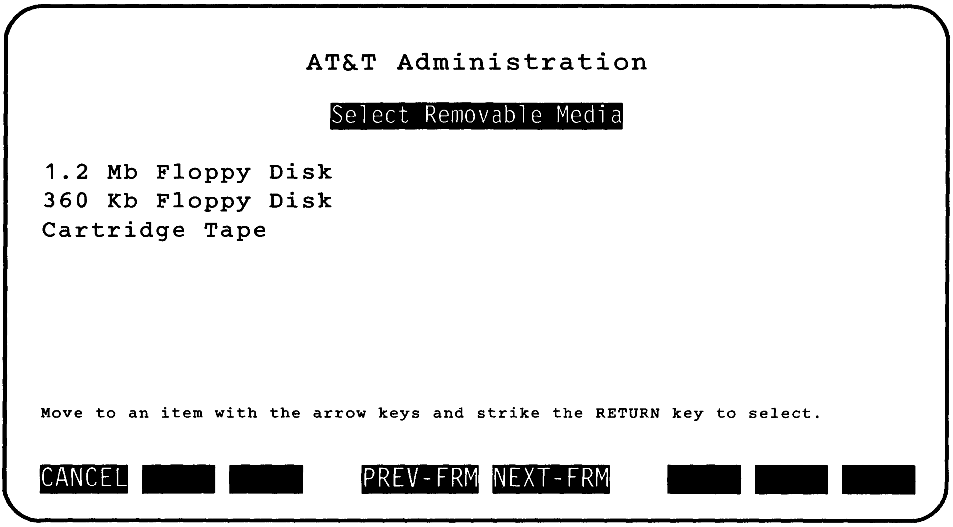
Use the following procedure to do an incremental system backup.

1. Highlight System Backup from the Backup to Removable Media menu and strike Enter. The System Backup form appears as follows:



```
                    AT&T Administration

                        System Backup

Backup Users
Backup System
Incremental System Backup
Selective System Backup




Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL                  PREV-FRM NEXT-FRM
```

2. From the System Backup menu, highlight Incremental System Backup and strike Enter. The Select Removable Media menu appears as follows:



```
                    AT&T Administration

                   Select Removable Media

1.2 Mb Floppy Disk
360 Kb Floppy Disk
Cartridge Tape






Move to an item with the arrow keys and strike the RETURN key to select.
```

3. Depending on what options are available on your computer and what your needs are, select the appropriate media for storing your file(s) by highlighting one of the three choices (1.2MB floppy disk, 360 KB floppy disk, or Cartridge Tape) and striking [Enter]. Once you have selected the media, you will receive the following message:

   Computing the number of files to be backed up. Please wait.

   Your computer will estimate the number of floppies (1.2MB and 360 KB) or tapes needed to fit all the files to be backed up and how much time the backup will take. Then, you will follow the instructions from your computer to insert and remove the previously formatted floppy disks or tapes and how to number them in sequence.

4. Insert a blank, formatted floppy disk or a cartridge tape and strike [Enter].

NOTE	The floppy disks used to back up your computer must be formatted in UNIX system format. (See "Format 1.2 MB UNIX System Floppy Disk" in this chapter.)

Once the backup is in progress, you will receive the following instruction:

```
Backup in progress. Do not remove the floppy/tape.
```

5. In addition, if the backup spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

   ```
   You may remove floppy number 1.
   To exit, please press 'q' followed by RETURN.

   To continue, insert floppy number 2
   and strike the RETURN key.
   ```

   If you press 'q' to exit, the following message is displayed:

   ```
   You have canceled the Backup to Removable Media.
   ```

   NOTE: As you remove each floppy disk or tape, attach a label containing subject, date, and the number of the floppy disk or tape. File the floppy disk in its envelope. If you write on a label already attached to the floppy disk, only use a felt-tip or nylon-tip pen. Do not use a ball-point pen to write on a label already attached; this can cause damage to the floppy disk.

6. Repeat inserting, removing, and labeling until a message appears indicating the backup is complete as follows:

   ```
   Backup is now done. You may remove the floppy.
   ```

7. Remove the last floppy disk when the system informs you that it has completed the backup.

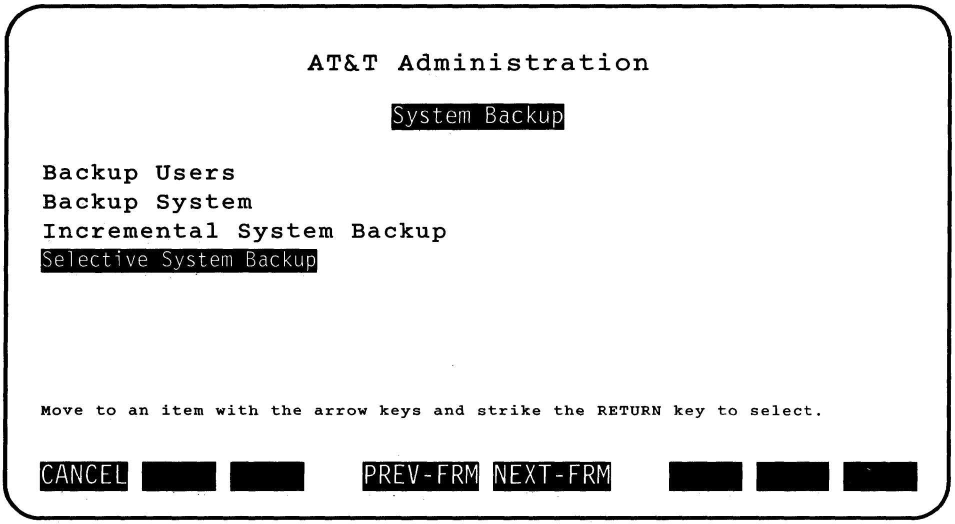
Selective System Backup

The selective system backup function backs up files or contents of directories in mounted file systems that you specify. The full path name must be specified.

You must have system administration privileges to use this function.

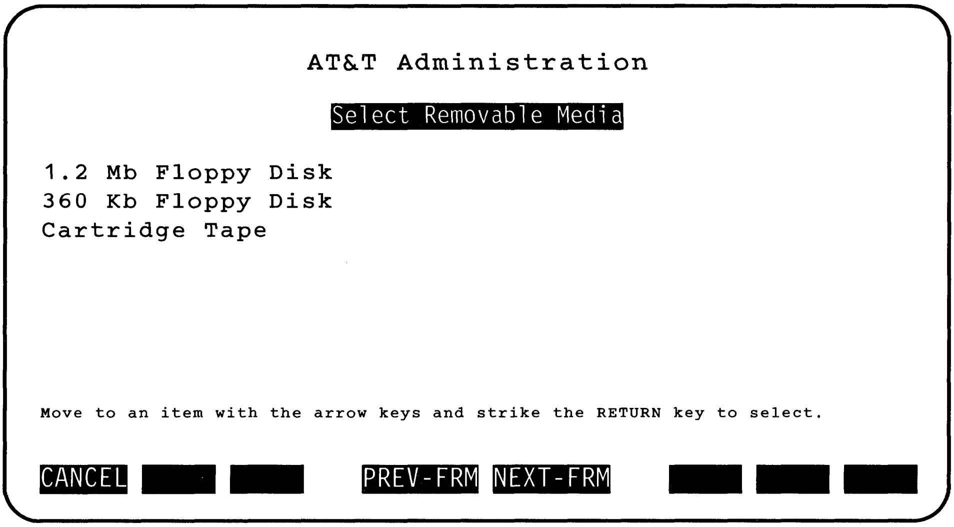
Use the following procedure to do a selective system backup.

1. Highlight System Backup from the Backup to Removable Media menu and strike (Enter). The System Backup form appears as follows:



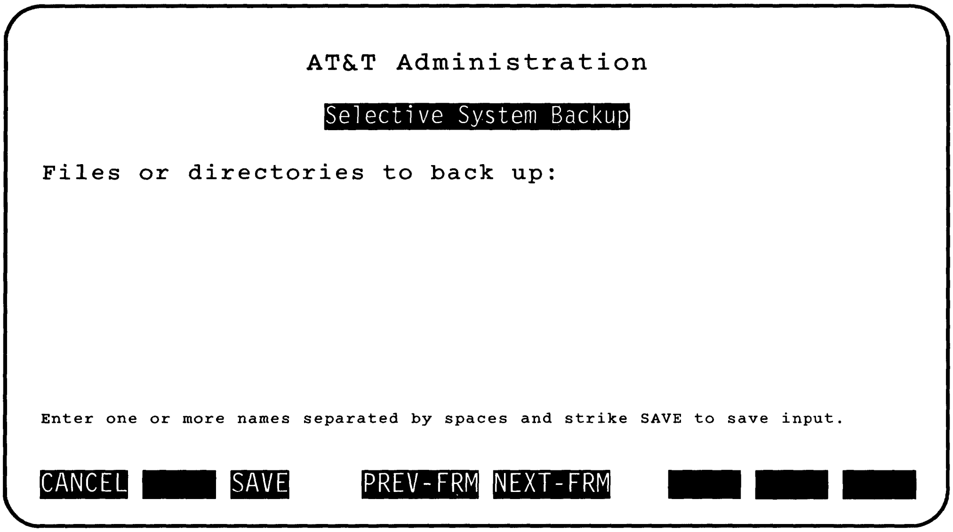
```
                    AT&T Administration

                       System Backup

Backup Users
Backup System
Incremental System Backup
Selective System Backup




Move to an item with the arrow keys and strike the RETURN key to select.
```

2. From the System Backup menu, highlight Selective System Backup and strike (Enter). The Select Removable Media menu appears as follows:



```
                    AT&T Administration

                   Select Removable Media

1.2 Mb Floppy Disk
360 Kb Floppy Disk
Cartridge Tape





Move to an item with the arrow keys and strike the RETURN key to select.
```

3. Depending on what options are available on your computer and what your needs are, select the appropriate media for storing your file(s) by highlighting one of the three choices (1.2MB floppy disk, 360 KB floppy disk, or Cartridge Tape) and striking Enter. The selective system backup form appears as follows:



```
                    AT&T Administration

                  Selective System Backup

Files or directories to back up:




Enter one or more names separated by spaces and strike SAVE to save input.
```

4. Enter one or more files or directories separated by spaces and strike SAVE. If the file or directory cannot be found, you will receive the following message:

   `<File name> cannot be found.`

   Once you have selected the files or directories, you will receive the following message:

   Computing the number of files to be backed up. Please wait.

   Your computer will estimate the number of floppies (1.2MB and 360 KB) or tapes needed to fit all the files to be backed up and how much time the backup will take. Then, you will follow the instructions from your computer to insert and remove the previously formatted floppy disks or tapes and how to number them in sequence.

5. Insert a blank, formatted floppy disk or a cartridge tape and strike Enter.

NOTE The floppy disks used to back up your computer must be formatted in UNIX system format. (See "Format 1.2 MB UNIX System Floppy Disk" in this chapter.)

Once the backup is in progress, you will receive the following instruction:

```
Backup in progress. Do not remove the floppy/tape.
```

6. In addition, if the backup spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

```
You may remove floppy number 1.
To exit, please press 'q' followed by RETURN.

To continue, insert floppy number 2
and strike the RETURN key.
```

If you press 'q' to exit, the following message is displayed:

```
You have canceled the Backup to Removable Media.
```

NOTE As you remove each floppy disk or tape, attach a label containing subject, date, and the number of the floppy disk or tape. File the floppy disk in its envelope. If you write on a label already attached to the floppy disk, only use a felt-tip or nylon-tip pen. Do not use a ball-point pen to write on a label already attached; this can cause damage to the floppy disk.

7. Repeat inserting, removing, and labeling until a message appears indicating the backup is complete as follows:

```
Backup is now done. You may remove the floppy.
```

8. Remove the last floppy disk when the system informs you that it has completed the backup.

Change Password

A password is a code word that should be known only by its creator. The password secures your login so no unauthorized person can enter the computer and have access to your files. Once a password has been assigned, it must be entered with your login when you want to use the computer.

After you have responded to the `login:` prompt, the password prompt appears on the screen. When you type in your password, it will not appear on the screen. This protects your password from being seen by someone else.

Each password is required to be at least six characters or longer. The password must have two alphabetic characters and at least one numeric character in the first eight characters.

Choose a password that is not common and is hard to guess. Your password should be changed from time to time to safeguard its secrecy.

You can only change passwords for your own login name. See Appendix C for a procedure to change the password for other users.

Changing Your Own Password

To change the password that is associated with your login, use the following procedure.

1. Log in using the login name associated with the password you want to change.

2. From the Administration menu, highlight Change Password and strike Enter. The screen clears and the UNIX system **passwd** command is executed. At the top of the screen, the following message is printed.

   ```
   Strike Break or Del to return to AT&T Administration
   without changing your password.
   ```

3. When prompted for your current password (Old password:) type the password you used when you logged in.

4. When prompted for the new password (`New password:`), enter the new password you want.

   The password you enter will not be displayed on the screen.

   You will receive an error message in the following circumstances:

   - If you enter the old password incorrectly
   - If the new password is not six character long
   - If the new password does not have two alphabetic characters and at least one special character in the first eight
   - If the password resembles the login name by being a reverse or circular shift
   - If the new password does not differ from the old password by three or more characters
   - If the new password includes a space or a `":"`
   - If you enter the new password incorrectly the second time.

5. When prompted to repeat the new password (`Reenter new password:`), type your new password again.

   If the two password entries are the same, the password is assigned. If the two password entries do not match, the message

```
     They don't match; try again.
     New password:
```

appears. If this message appears, type the new password again and then reenter the new password again.

6. After you reenter the new password, you will be prompted to:

   ```
   Strike RETURN to continue with AT&T Administration.
   ```

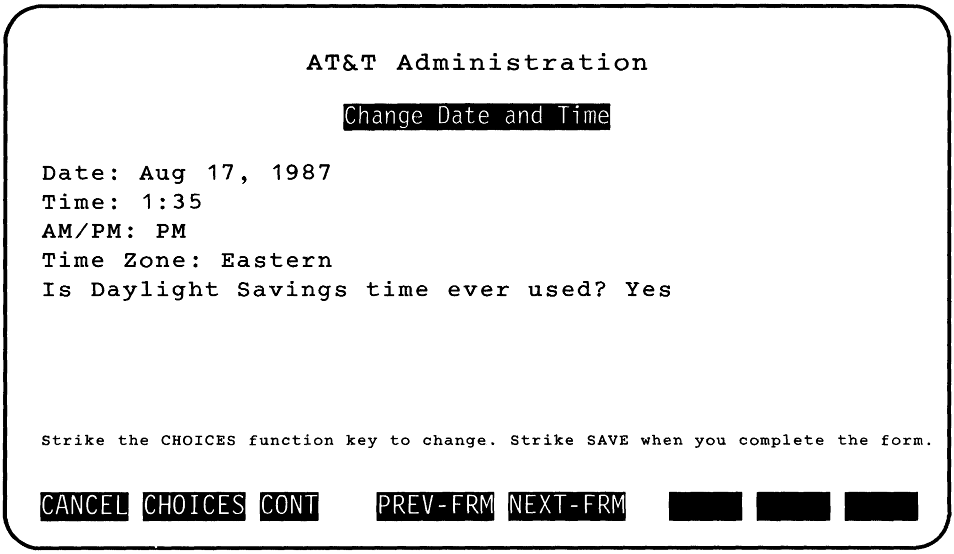
   Strike Enter to return to the Administration menu.

Date and Time

The system clock can be changed using the Date and Time feature. You must have system administration privileges to change the date and time. You can use the Date and Time form to display the current setting, change a value as desired, and verify that the appropriate changes have been made.

Before re-setting the date or time, notify all users that the date is being reset. Changing the date and/or time may disrupt **make**, **cron**, a compile, or any applications that rely on the current date thus disturbing other users' work. To set the UNIX system clock, use the following procedure.

1. From the Administration menu, highlight Date and Time and strike Enter. The Date and Time form appears as follows:



```
AT&T Administration

Change Date and Time

Date: Aug 17, 1987
Time: 1:35
AM/PM: PM
Time Zone: Eastern
Is Daylight Savings time ever used? Yes

Strike the CHOICES function key to change. Strike SAVE when you complete the form.
```

2. Use the arrow keys to move the cursor to the field to be changed.

3. Once the cursor is resting on the field to be changed, strike the CHOICES key. If only two or three choices are available, the choice will toggle when the CHOICES key is striked. If more than three choices are available for you to enter into each field a pop-up menu will appear. Use the arrow keys to move to the appropriate entry and select by striking Enter.

4. The date fields are: months of the year, date of the month, and year. Valid values are as follows:

   ```
   months: January - December
     days: 1 - 31
    years: 1987 - 1999
   ```

   The time fields are: hours of the day and minutes of the hour. Valid values are as follows:

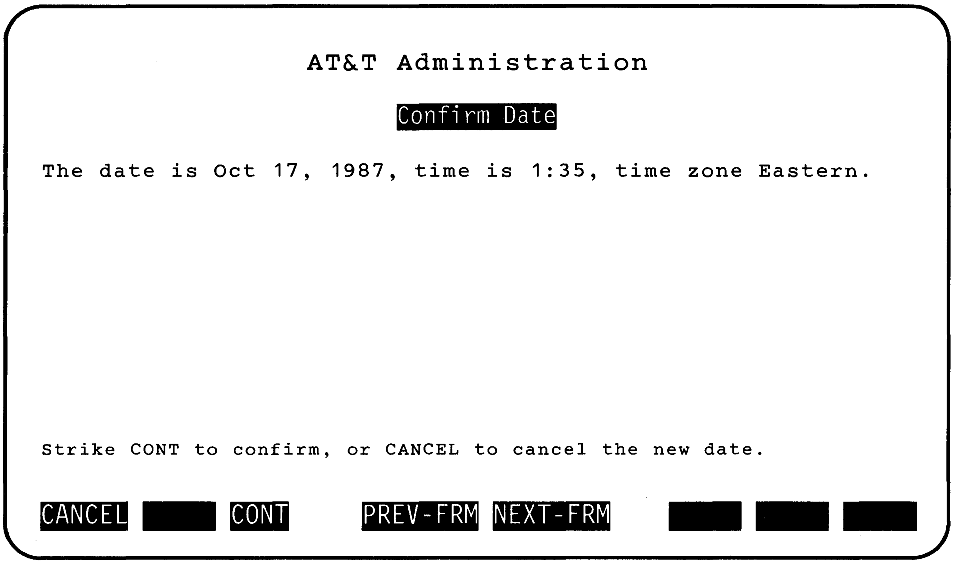
   ```
     Hours: 1 - 12
   Minutes: 00 - 59
   ```

   The remaining fields on the form are AM/PM, Time Zone (which can be Eastern, Central, Mountain, Pacific, or GMT), and Daylight Savings Time.

   If you type in an invalid value for these fields, an error message appears. An error will also occur when month, date, and year are in an impossible combination.

5. Strike SAVE after you've changed all required fields on the Date and Time form.

   The confirm change message lets you know that the computer set the new date and time. A sample confirm change message appears as follows:



```
                    AT&T Administration

                        Confirm Date

The date is Oct 17, 1987, time is 1:35, time zone Eastern.




Strike CONT to confirm, or CANCEL to cancel the new date.
```

6. Strike CONT to change the date and time on the system clock, or strike CANCEL to cancel without changing the date and time.

NOTE	You must log off and log back on again to see the effects of changing either your time zone, or the use of Daylight Savings Time.

Disk Operations

The Disk operations function allows the user to:

- Copy the contents of one floppy to another floppy
- Format floppies.

Floppy-To-Floppy Copy

For safekeeping you can store important information on more than one floppy disk by copying the information from one floppy disk to another.

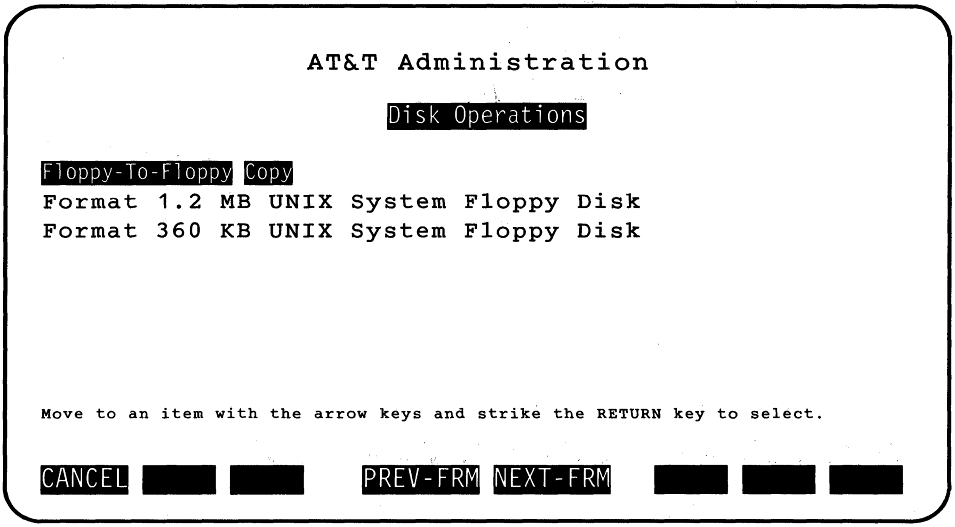
The floppy disk containing the files you want to copy is called the *source floppy disk,* and the floppy disk you want to copy the information to is called the *destination floppy disk.* Remember to format the destination floppy disk before you start.

CAUTION It is recommended that the source and destination floppies be the same density, i.e., both should be 1.2 MB floppies or both should be 360 KB floppies. It is possible to copy 360 KB floppies to 1.2 MB floppies without any problems, but if you try to copy a 1.2 MB floppy to a 360 KB floppy you will risk running out of space on the 360 KB floppy.

While a floppy is being copied or being copied to, never remove the floppy disk from the disk drive. Wait for the message to insert or remove the appropriate floppy disk.

To copy the contents from a source floppy disk to a destination floppy disk, use the following procedure.

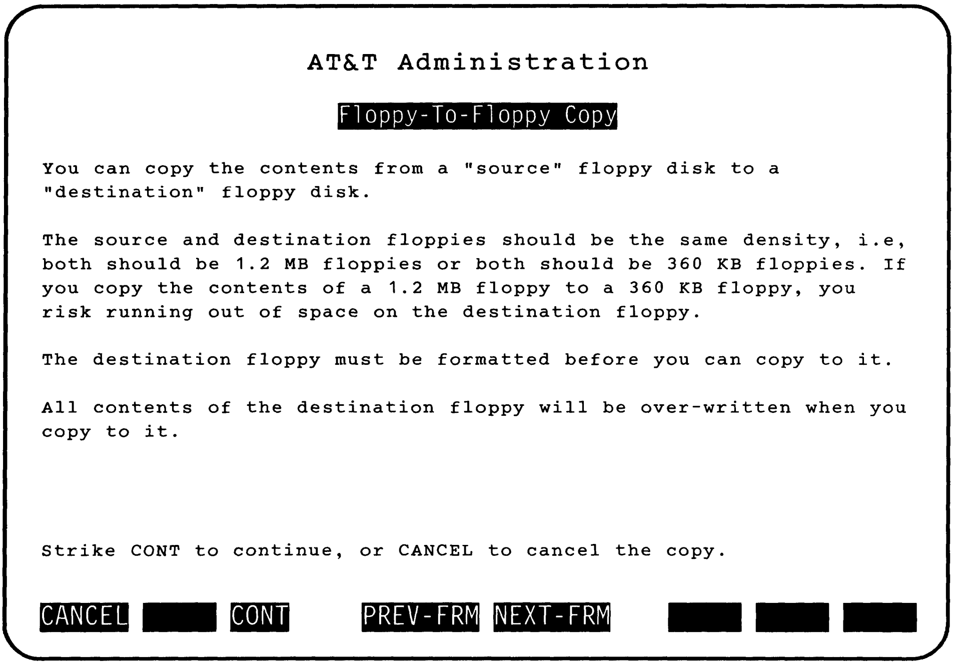
1. From the Administration menu, highlight `Disk Operations` and strike Enter. The Disk Operations menu appears as follows:



```
                    AT&T Administration

                      Disk Operations

Floppy-To-Floppy Copy
Format 1.2 MB UNIX System Floppy Disk
Format 360 KB UNIX System Floppy Disk







Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL                 PREV-FRM NEXT-FRM
```

2. Highlight Floppy-To-Floppy Copy from the Disk Operations menu and strike Enter.

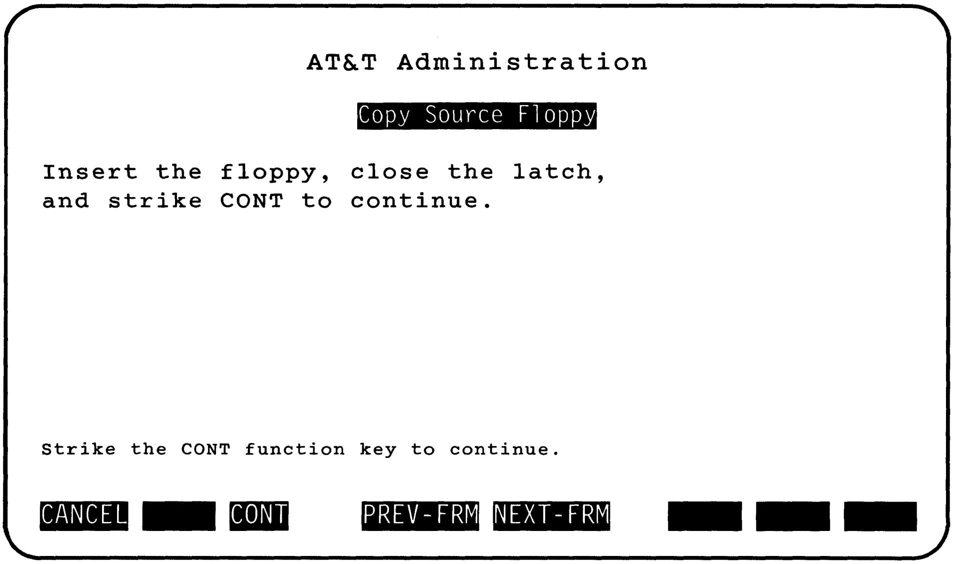
3. The Floppy-To-Floppy Copy form appears as follows:



```
                    AT&T Administration

                  Floppy-To-Floppy Copy

You can copy the contents from a "source" floppy disk to a
"destination" floppy disk.

The source and destination floppies should be the same density, i.e,
both should be 1.2 MB floppies or both should be 360 KB floppies. If
you copy the contents of a 1.2 MB floppy to a 360 KB floppy, you
risk running out of space on the destination floppy.

The destination floppy must be formatted before you can copy to it.

All contents of the destination floppy will be over-written when you
copy to it.



Strike CONT to continue, or CANCEL to cancel the copy.
```

4. Ensure the following:
   - that the source and destination floppies are the same density,
   - the destination floppy is formatted, and
   - it is alright to overwrite the destination floppy.

   Then strike CONT.
5. The Copy Source Floppy form appears as follows:



```
                    AT&T Administration

                    Copy Source Floppy

Insert the floppy, close the latch,
and strike CONT to continue.




Strike the CONT function key to continue.
```

6. Insert the source floppy disk and close the latch.
7. Strike CONT.

   If you get an error message, the source floppy can not be copied. The possible reasons include the following:

   - There is no floppy inserted.
   - The floppy is inserted improperly.
   - The latch is not turned down.
   - The floppy is not readable because it is unformatted or formatted incorrectly.

   If you get an error message, check to make sure that you put the floppy in the drive and that the latch is turned down. Try re-inserting the floppy. Also, verify that the floppy is the floppy you intended to copy.

8. When the floppy is inserted properly and being read, the message line appears as follows:

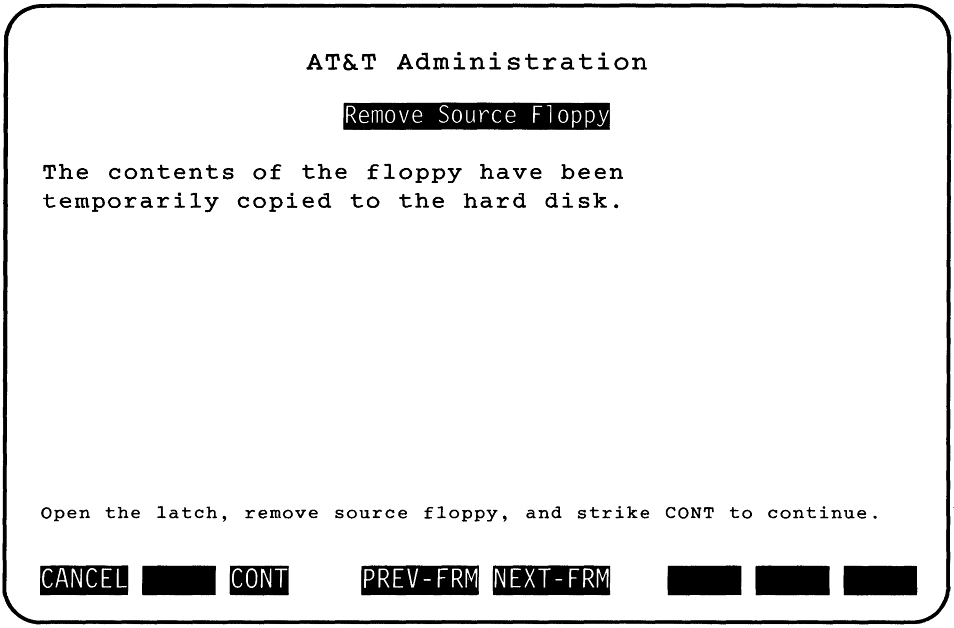
```
Please wait while source floppy is being copied.
```

The CANCEL key will do nothing if struck at this time. The computer is now copying the files from the source floppy disk onto the hard disk. If your computer does not have enough space to copy this floppy, you will receive an error message. If you receive this error message, you need to delete some files on the hard disk before trying to copy the floppy again.

NOTE

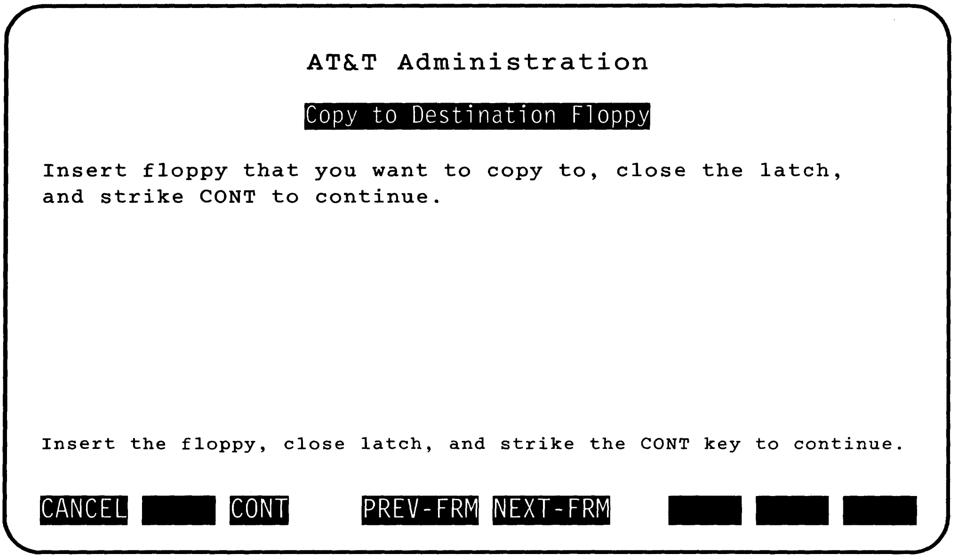
If you strike CANCEL while the source floppy is being copied, the copy will not be effected. However, the rest of the task (i.e., copy to the destination copy) will be canceled. You will return to the Disk Operations menu.

When copying is complete, the Remove Source Floppy form appears as follows:



```
                    AT&T Administration

                    Remove Source Floppy

The contents of the floppy have been
temporarily copied to the hard disk.









Open the latch, remove source floppy, and strike CONT to continue.
```

9. Remove the source floppy disk and strike CONT.
10. The Copy to Destination Floppy form appears as follows:



```
                    AT&T Administration

                 Copy to Destination Floppy

Insert floppy that you want to copy to, close the latch,
and strike CONT to continue.







Insert the floppy, close latch, and strike the CONT key to continue.
```

11. Insert the destination floppy disk and strike CONT.

    If you receive the following error message,

    > The destination floppy can not be copied to. The possible reasons include: there is no floppy inserted; the floppy is inserted improperly; the latch is not turned down; or the floppy is not writable because it has a write protect tab or it is unformatted or formatted incorrectly.

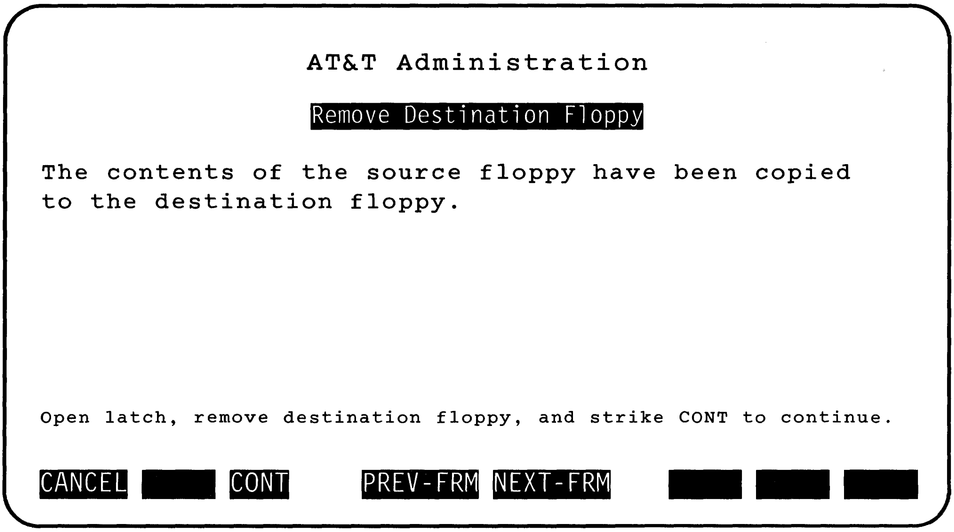
    Check that you inserted a floppy and closed the latch. Take the floppy out and check for a write protect tab. If it has one, remove it or insert a different formatted floppy. If you suspect the floppy may not be formattted, insert one that you know is formatted.

    You might also mistakenly insert a 360 KB floppy when you should have inserted a 1.2 MB floppy. In this case, you should remove the floppy and insert a 1.2 MB floppy.

    If no errors are encountered, the message line appears as follows:

    ```
    Please wait while floppy is being written.
    ```

When the write from hard disk to destination floppy is finished, the Remove Destination Floppy form appears as follows:



AT&T Administration

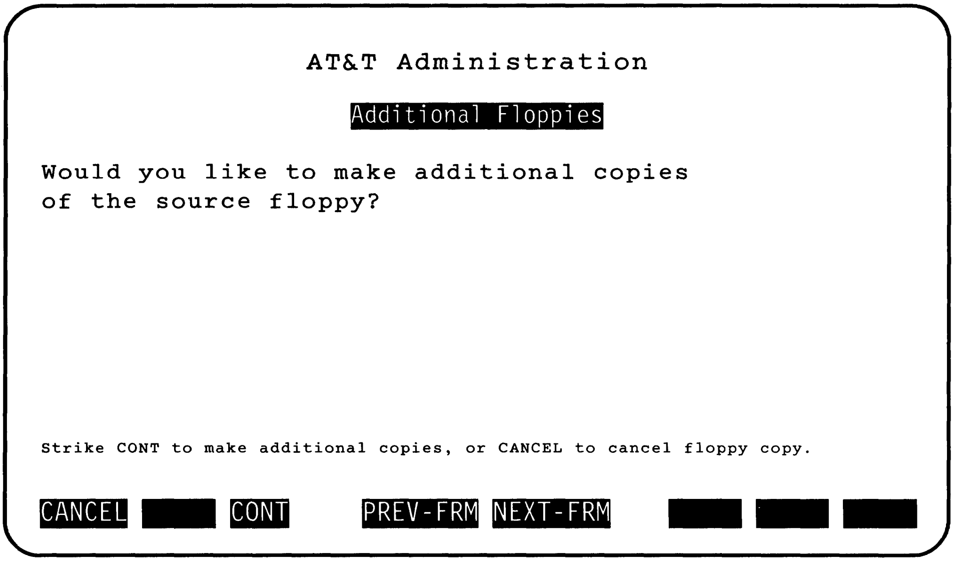
Remove Destination Floppy

```
The contents of the source floppy have been copied
to the destination floppy.
```

```
Open latch, remove destination floppy, and strike CONT to continue.
```

12. Remove the destination floppy disk and strike CONT.

After you strike CONT, the Additional Floppies form appears as follows:



```
                    AT&T Administration

                     Additional Floppies

Would you like to make additional copies
of the source floppy?




Strike CONT to make additional copies, or CANCEL to cancel floppy copy.
```

13. If you do not need another copy, strike CANCEL to exit this procedure.

14. If you want to make another copy, strike CONT and follow the menu instuctions that repeat this procedure.

15. When you're finally finished, strike CANCEL to return to the Disk Operations menu.

Format 1.2 MB UNIX System Floppy Disk

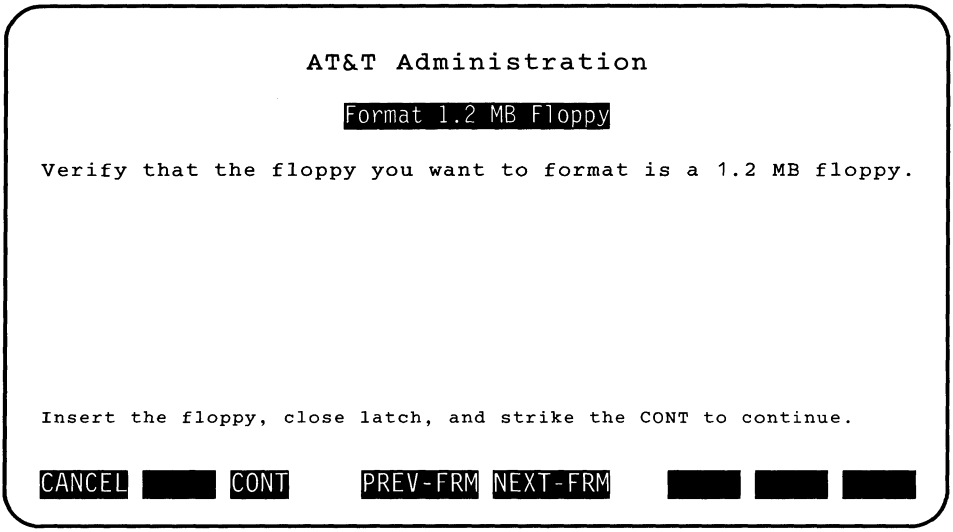
Format all new floppy disks before you use them with the computer. Formatting a floppy disk prepares it to accept the directories and files you want to store.

You can also reformat an old floppy disk. Be sure you do not need the information on it any longer.

CAUTION **DO NOT FORMAT A FLOPPY DISK THAT CONTAINS INFORMATION YOU WANT TO KEEP.** Formatting a floppy disk destroys any existing information.

To format a 1.2 MB floppy disk, use the following procedure.

1. Highlight `Format 1.2 MB UNIX System Floppy Disk` from the Disk Operations menu and strike Enter. The Format 1.2 MB Floppy form appears as follows:



AT&T Administration

Format 1.2 MB Floppy

Verify that the floppy you want to format is a 1.2 MB floppy.

Insert the floppy, close latch, and strike the CONT to continue.

2. Insert the floppy disk you want to format and strike CONT. If you see the following message:

   > The floppy can not be formatted. The possible reasons include: there is no floppy inserted; the floppy is inserted improperly; the latch is not turned down; the floppy is not writable because it has a write protect tab on it.

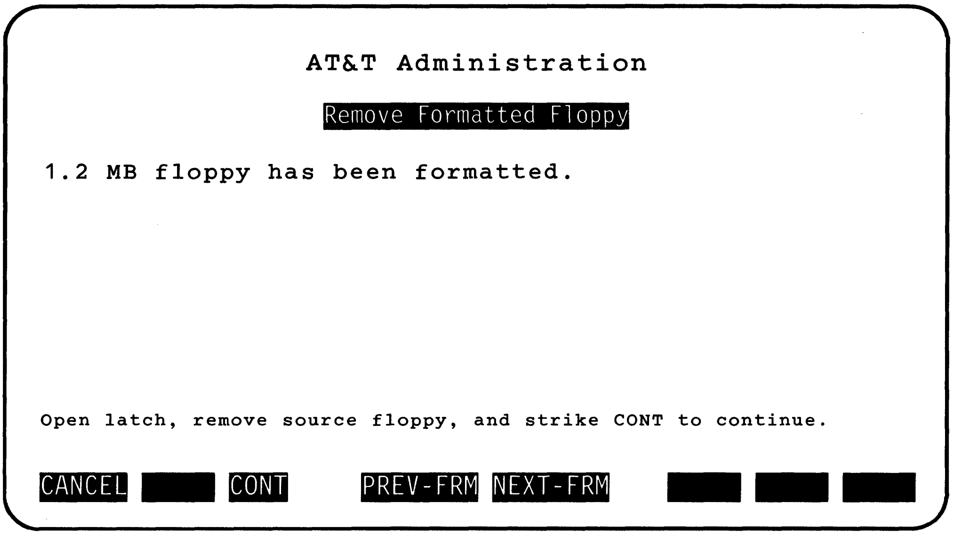
   Check that you inserted a floppy and closed the latch. Take the floppy out and check for a write protect tab. If it has one, remove it and reinsert the floppy. If the floppy is the incorrect density, insert a floppy of the correct density.

   You might also get a message that tells you explicitly that the floppy is the wrong density. Remove the floppy and insert one of the correct density.

   If no error conditions are encountered, the message line appears as follows:

   ```
   Formatting of 1.2 MB floppy is in progress.
   ```

After the floppy disk is formatted, the Remove Formatted Floppy form appears as follows:



```
                    AT&T Administration

                  Remove Formatted Floppy

1.2 MB floppy has been formatted.




Open latch, remove source floppy, and strike CONT to continue.
```

3. Remove the floppy disk from the disk drive and strike CONT.

   Repeat the procedure by following the same procedure for each floppy disk you need to format.

4. Strike CANCEL to exit when finished.

Format 360 KB UNIX System Floppy Disk

The procedure for formatting a 360 KB floppy disk is identical to formatting a 1.2 MB floppy disk except you select `Format 360 KB UNIX System Floppy Disk` from the Disk Operations menu. See section "Format 1.2 MB UNIX System Floppy Disk" for procedural details.

File System Operations

The file systems you create are an independent collection of files and directories. The File System Operation function allows you to

- Create a file system on floppy disk
- Mount a file system on both hard disk and floppy disk
- Mount a file system on a second hard disk if one is installed
- Unmount a previously mounted file system.

Before a floppy disk file system is accessible through the UNIX system, it must be attached to a directory that is already a part of the file system on the integral hard disk. This is referred to as mounting a file system. The location in the file system where the **mount** command attaches the mounted file system is called the mount point.

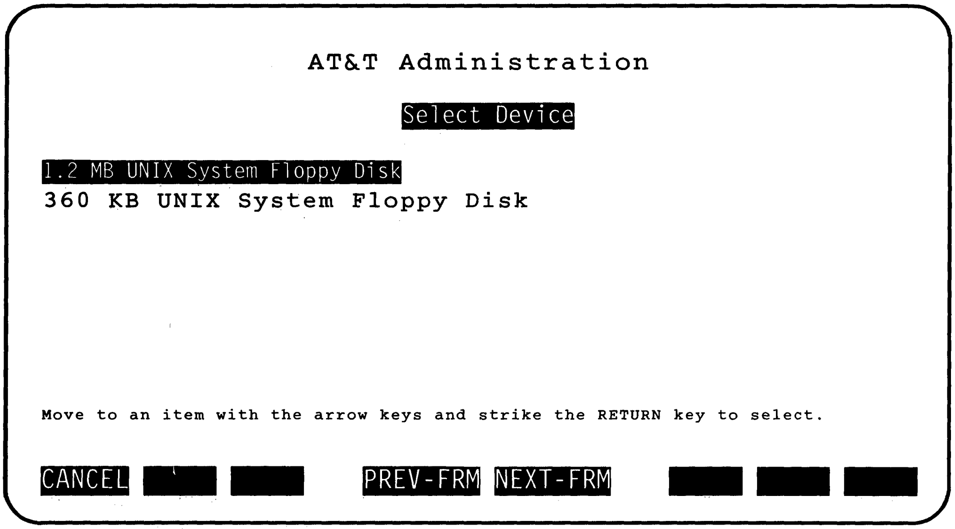
Therefore, a file system is brought under UNIX system control by mounting the file system. To release the file system so that it can be removed from the UNIX system, the file system must be unmounted.

Create File System

A 1K UNIX system file system is created on a floppy disk. After the file system is created, it is marked with a file system name and optional file system label that can be mounted at a later time.

Use the following procedure to create a file system on a floppy diskette.

1. Highlight Create File System from the File System Operations menu and strike Enter. The Select Device Menu appears as follows:



AT&T Administration

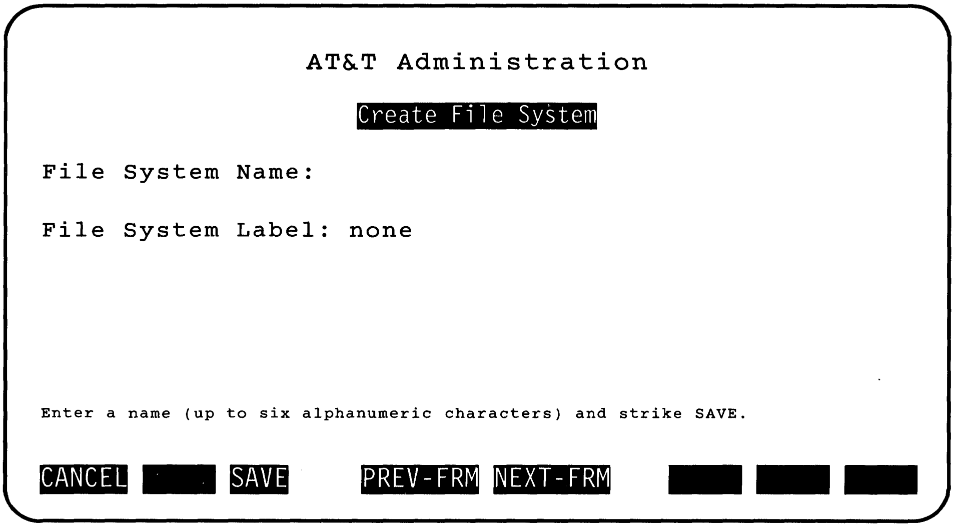
Select Device

1.2 MB UNIX System Floppy Disk
360 KB UNIX System Floppy Disk

Move to an item with the arrow keys and strike the RETURN key to select.

2. For the density desired, select 1.2 MB UNIX System Floppy Disk or 360 KB UNIX System Floppy Disk and strike Enter.

   Now the Create File System Form appears as follows:



AT&T Administration

Create File System

File System Name:

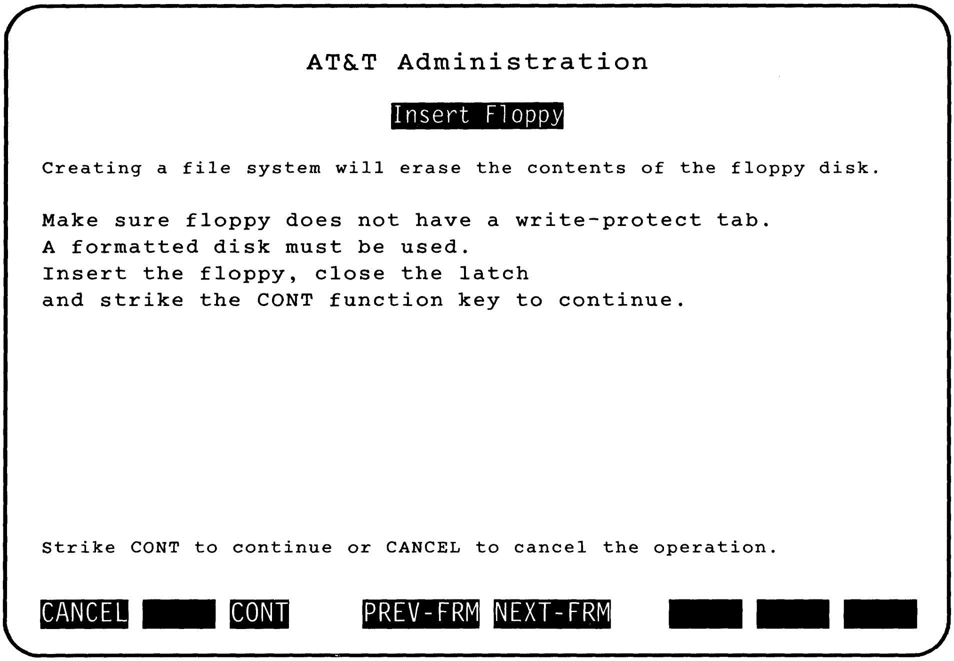
File System Label: none

Enter a name (up to six alphanumeric characters) and strike SAVE.

3. You must enter the file system name (the label is optional) and strike SAVE. By default, label is "none". You may change this label to any other name (up to six alphanumberic characters).

   You will receive an error message if your file system or label name is longer than six characters.

   The Insert Floppy form appears as follows:

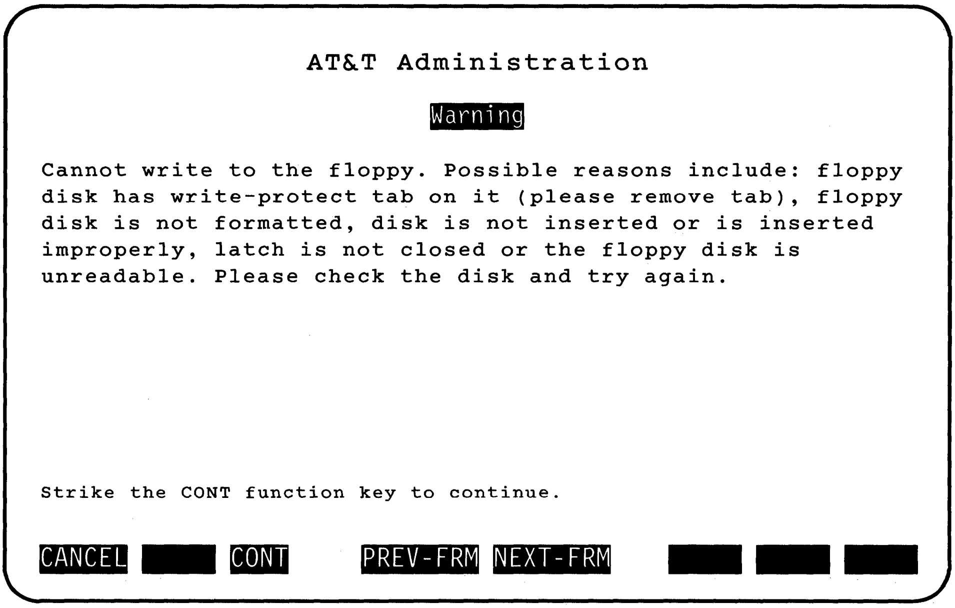


```
                    AT&T Administration

                       Insert Floppy

Creating a file system will erase the contents of the floppy disk.

Make sure floppy does not have a write-protect tab.
A formatted disk must be used.
Insert the floppy, close the latch
and strike the CONT function key to continue.








Strike CONT to continue or CANCEL to cancel the operation.
```

4. Next, insert the floppy diskette, close the latch, and strike CONT. You are asked to wait until the file system is created.

Create File System Warning Messages

If a write error to the floppy occurs, the following Warning message appears:



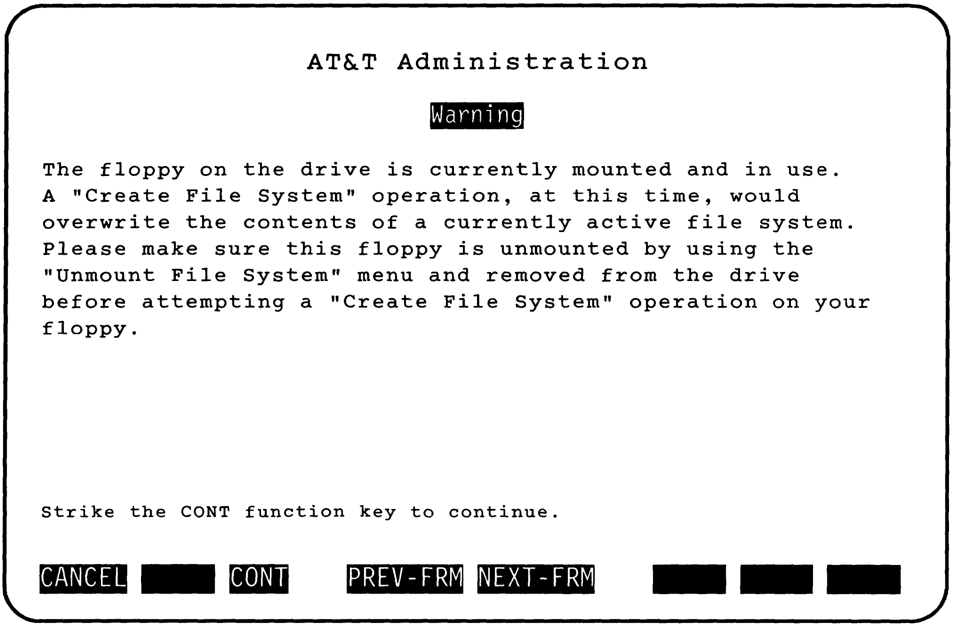
```
                    AT&T Administration

                          Warning

Cannot write to the floppy. Possible reasons include: floppy
disk has write-protect tab on it (please remove tab), floppy
disk is not formatted, disk is not inserted or is inserted
improperly, latch is not closed or the floppy disk is
unreadable. Please check the disk and try again.




Strike the CONT function key to continue.
```

5. If the Warning message appears, do the following:

   - Remove the floppy from the drive.
   - Check for a write-protect tab.
   - Reinsert the floppy and close the latch and then, strike CONT.

   If the error persists, try another formatted disk.

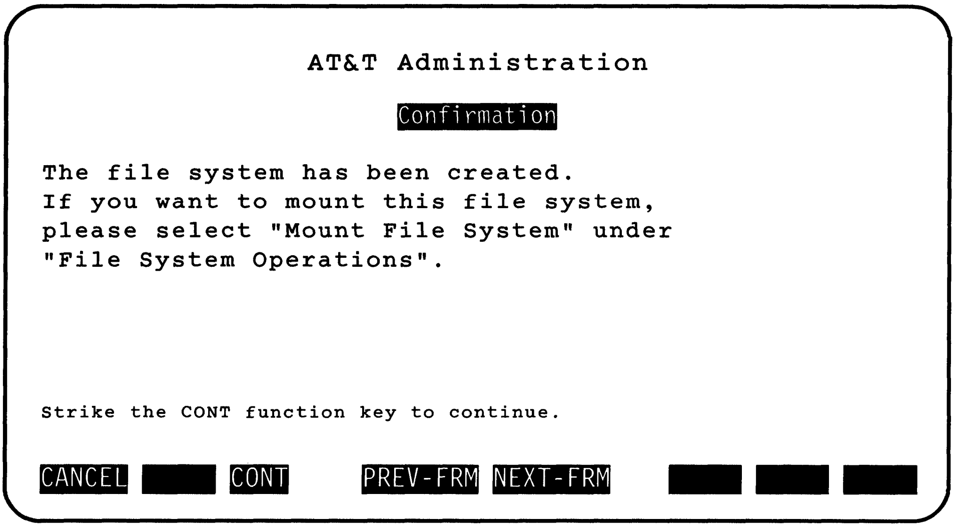
Another possible error is trying to create a file system on a mounted floppy. If the floppy on the device has a mounted file system, you will not be allowed to create a new file system on this floppy. This error is detected and you are notified as follows:



```
                    AT&T Administration

                          Warning

The floppy on the drive is currently mounted and in use.
A "Create File System" operation, at this time, would
overwrite the contents of a currently active file system.
Please make sure this floppy is unmounted by using the
"Unmount File System" menu and removed from the drive
before attempting a "Create File System" operation on your
floppy.




Strike the CONT function key to continue.
```

6. If the previous Warning message appears, unmount the floppy and restart the operation by striking CONT or CANCEL.

7. If no errors are encountered, the Confirmation form appears as follows:



AT&T Administration

Confirmation

```
The file system has been created.
If you want to mount this file system,
please select "Mount File System" under
"File System Operations".
```

Strike the CONT function key to continue.

8. Strike the CONT function key.

   The floppy file system is created and left unmounted.

NOTE	Remember to mount the file system using the Mount File System function before trying to use it.

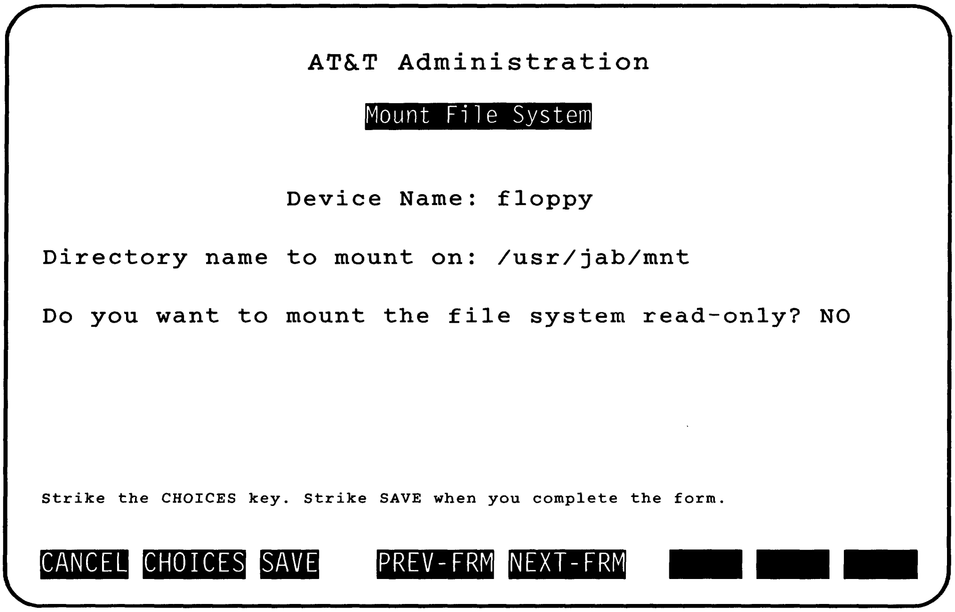
Mount File System

Both floppy file systems, as well as, file systems on a second hard disk can be mounted. You must specify the mount directory name where you want to mount the file system. A write protected floppy can only be mounted as read-only. When the operation is completed, the system will give you a confirmation message that the file system has been mounted.

You must have system administration privileges to use the Mount File System function.

Use the following procedure to mount a file system on a floppy diskette.

1. Highlight Mount File System from the File System Operations menu and strike Enter. The Mount File System form appears as follows:



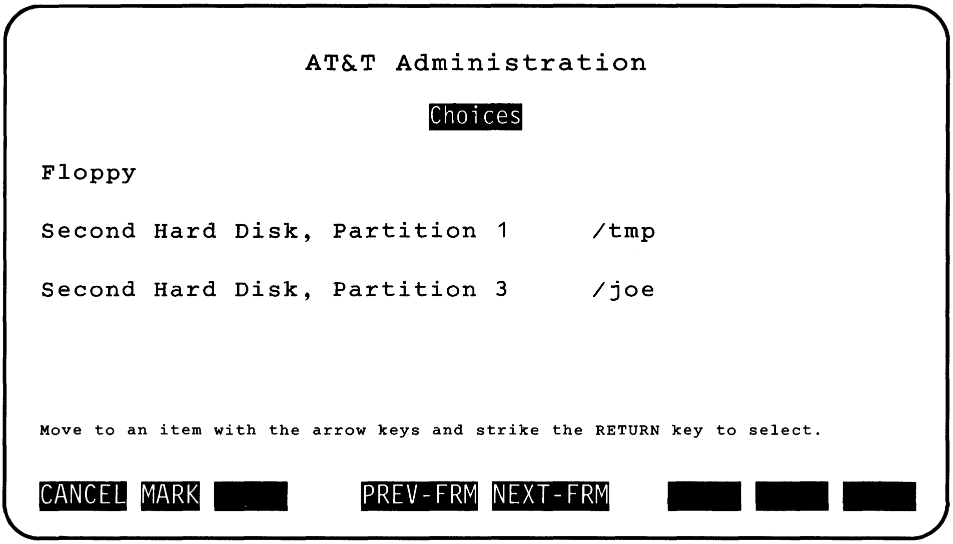
```
                    AT&T Administration

                     Mount File System


                  Device Name: floppy

Directory name to mount on: /usr/jab/mnt

Do you want to mount the file system read-only? NO




Strike the CHOICES key. Strike SAVE when you complete the form.
```

2. If your computer is not set up with a second hard disk, then the entire "Device Name:" field will not appear on your screen. Floppy disk is assumed. Skip to Step 4.

   If your computer does have a second hard disk set up, floppy will appear in the "Device Name:" by default. If you want to mount a file system that is on the floppy, skip this step and move to the next field.

If you want to mount the file system that is on the second hard disk, strike CHOICES while the cursor is resting on the "Device Name:" field.

The following example pop-up menu appears:



```
                    AT&T Administration

                          Choices

Floppy

Second Hard Disk, Partition 1     /tmp

Second Hard Disk, Partition 3     /joe



Move to an item with the arrow keys and strike the RETURN key to select.
```

NOTE

Release 1 supports hard disks connected only to a single controller.

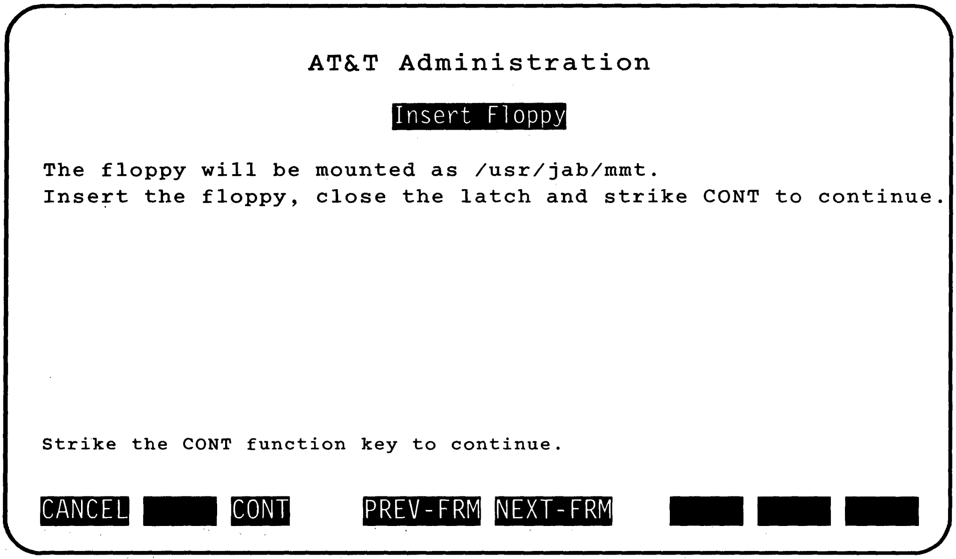
3. Highlight the device name in the Choices pop-up menu and strike Enter.

4. If you are satisfied with using the default *<your home dir>/mnt* as a mount directory for the file system, move to the "Do you want to mount the file system read-only?" field.

Otherwise, with your cursor resting on the "Directory name to mount on:" field, enter the mount point desired (*<path name>/mnt*). You must specify the directory you select for the mount point as a full path name. If you select a directory that does not exist, it will be created. If you select a directory that exists and has contents, the current contents of the directory will not be accessible while the file system is mounted.

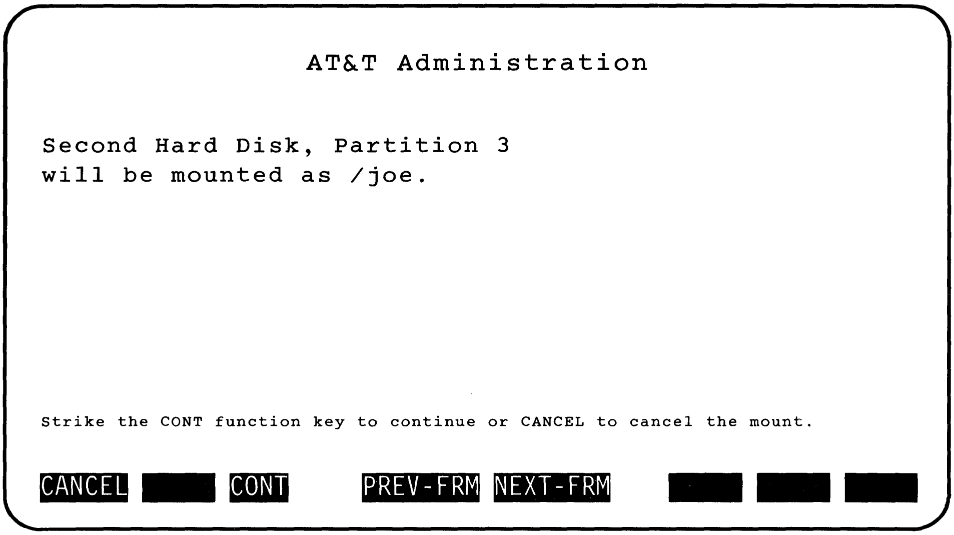
5. With the cursor resting on the "Do you want to mount the file system read-only?" field, strike CHOICES and toggle (YES or NO) until your choice appears in the field.
6. When the form is complete, strike SAVE.

If you are mounting a floppy, the Insert Floppy form appears as follows:



```
                    AT&T Administration

                       Insert Floppy

The floppy will be mounted as /usr/jab/mmt.
Insert the floppy, close the latch and strike CONT to continue.







Strike the CONT function key to continue.
```

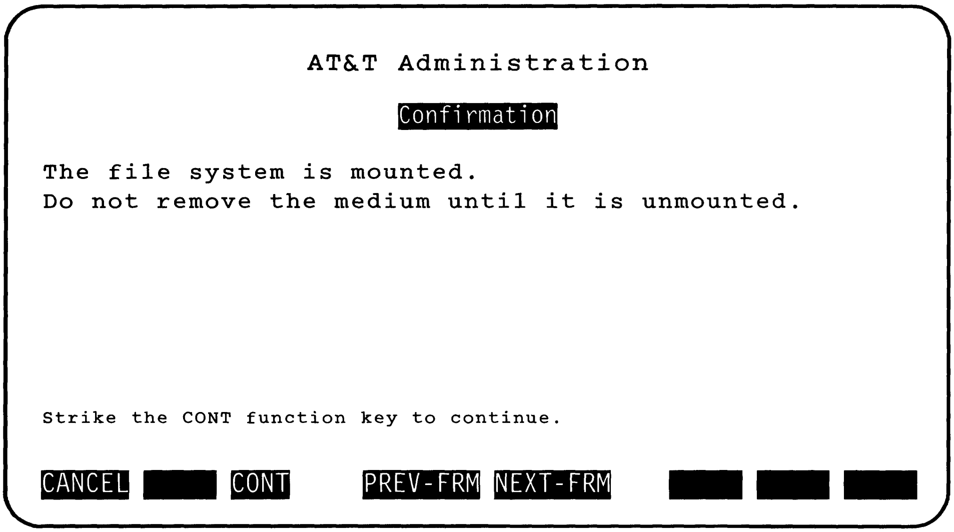
If you are mounting to the second hard disk, the following screen appears:



```
                    AT&T Administration

Second Hard Disk, Partition 3
will be mounted as /joe.




Strike the CONT function key to continue or CANCEL to cancel the mount.
```

7. If you are mounting a floppy, insert the floppy, close the latch, and strike CONT.

When the file system is mounted, you will receive a Confirmation pop-up form as follows:

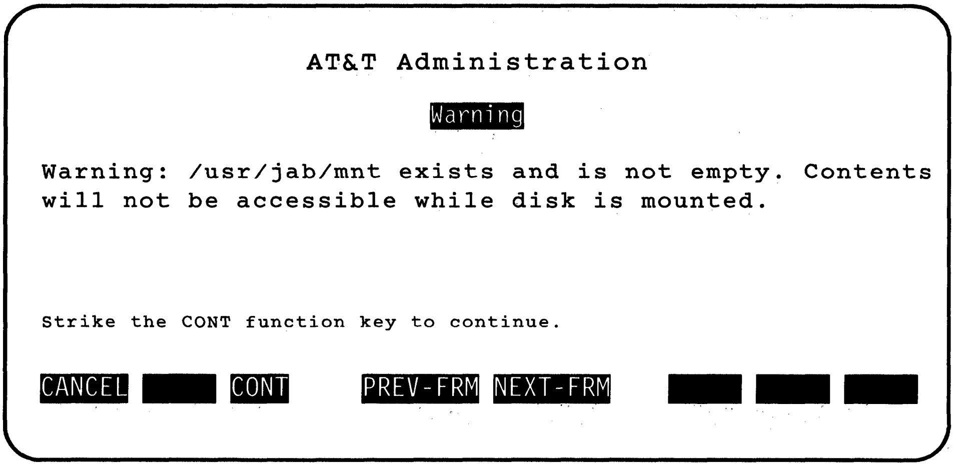


```
                    AT&T Administration

                        Confirmation

The file system is mounted.
Do not remove the medium until it is unmounted.




Strike the CONT function key to continue.
```

Mount File System Warning Messages

Directory Not Empty

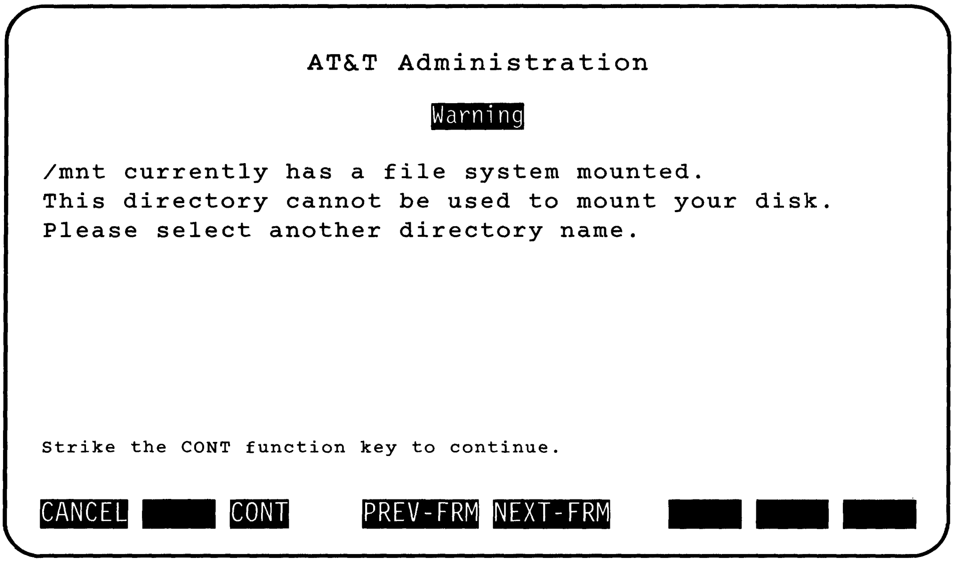
If the mount point is not empty, a warning message appears as follows:



```
AT&T Administration

Warning

Warning: /usr/jab/mnt exists and is not empty. Contents
will not be accessible while disk is mounted.



Strike the CONT function key to continue.
```

Mount Directory in Use

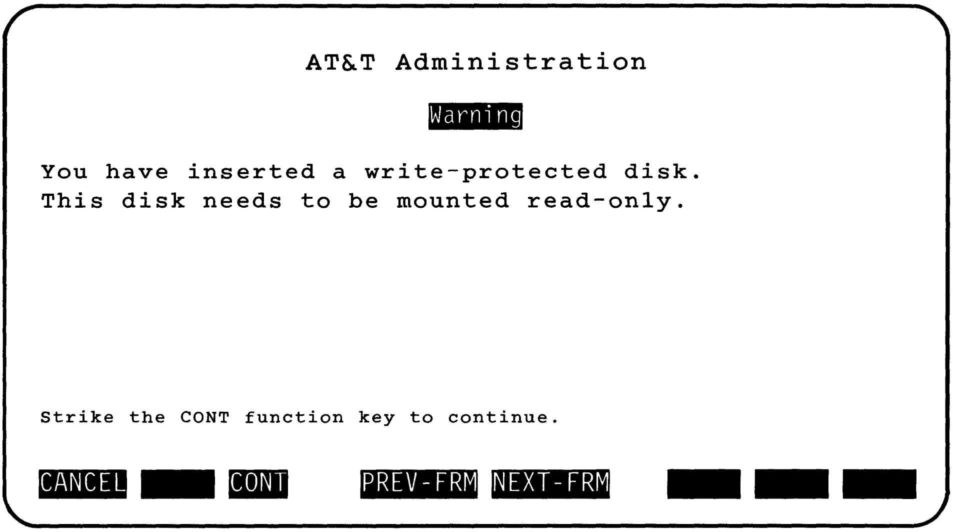
If another file system (e.g., a hard disk file system) is already mounted using the same mount directory, the following Warning message appears and the file system is not mounted.



```
                    AT&T Administration

                          Warning

/mnt currently has a file system mounted.
This directory cannot be used to mount your disk.
Please select another directory name.




Strike the CONT function key to continue.
```

Write-Protected Disk (only applies to floppy file systems)

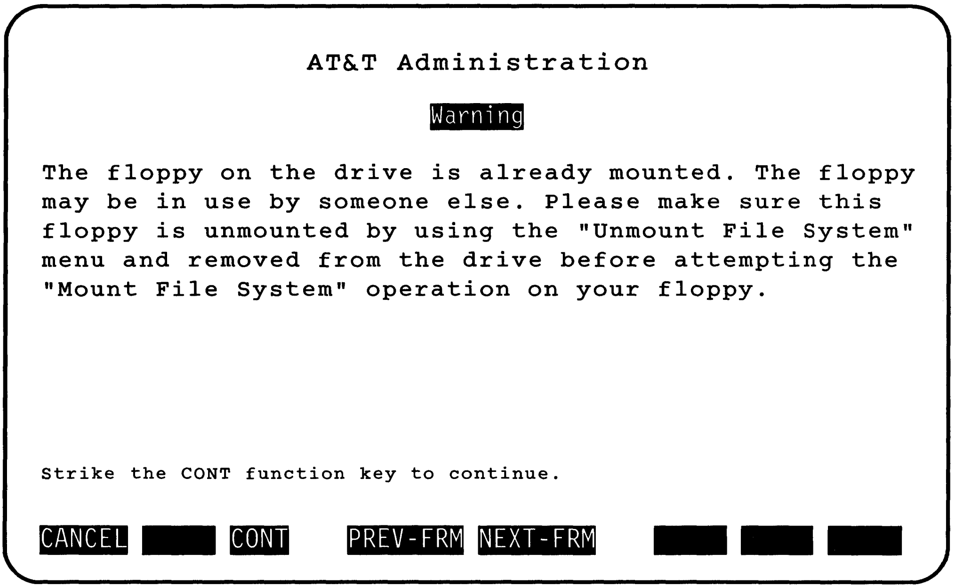
If you attempt to mount a write-protected floppy disk (i.e., the floppy has the write-protect tab on it) as read/write, the following Warning message appears:



```
                    AT&T Administration

                          Warning

You have inserted a write-protected disk.
This disk needs to be mounted read-only.




Strike the CONT function key to continue.
```

Floppy Mounted (only applies to floppy file systems)

If the floppy disk is already mounted (presumably by somebody else), the following Warning message appears:



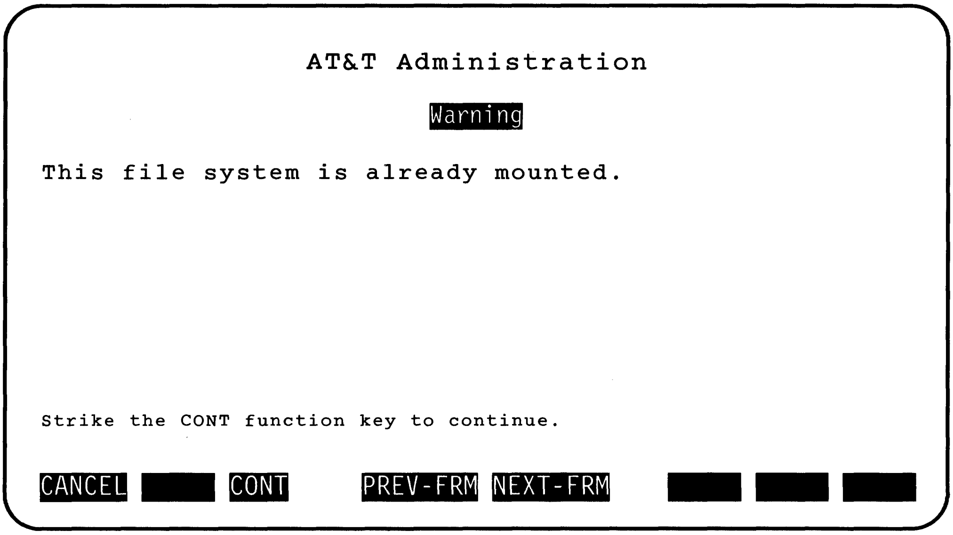
```
                    AT&T Administration

                          Warning

The floppy on the drive is already mounted. The floppy
may be in use by someone else. Please make sure this
floppy is unmounted by using the "Unmount File System"
menu and removed from the drive before attempting the
"Mount File System" operation on your floppy.




Strike the CONT function key to continue.
```

NOTE

The file system is not mounted. In this case, the file system is not mounted.

Hard Disk File System Already Mounted

If a file system on a second hard disk is already mounted, the following Warning message appears:



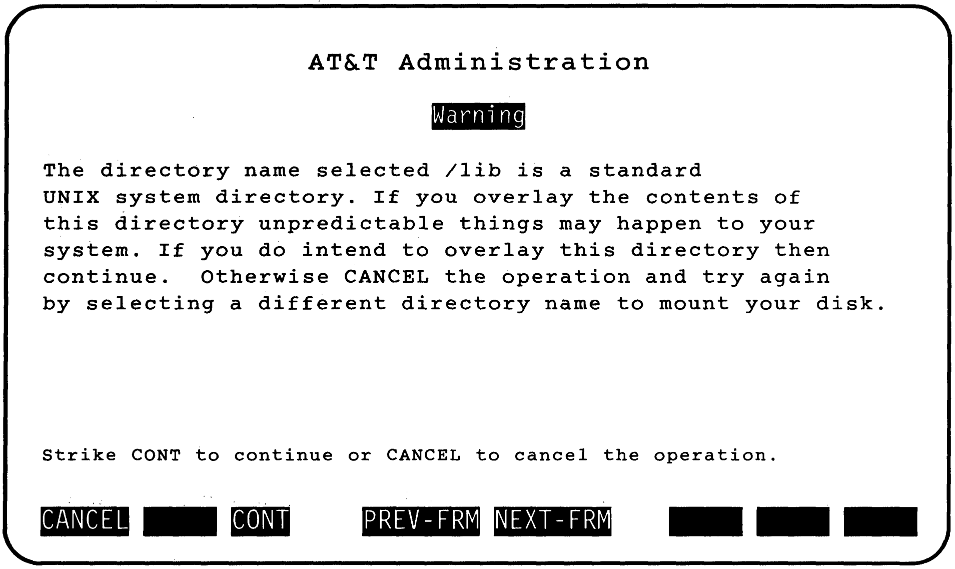
```
                 AT&T Administration

                      Warning

This file system is already mounted.





Strike the CONT function key to continue.
```

NOTE

The file system is not mounted. In this case, the file system is not mounted.

Directory Is a Standard UNIX System Directory

If the directory you specify is one of the standard UNIX system directories: */dev, bin, /lib, /usr,* and */tmp;* the following Warning message appears:



```
                    AT&T Administration

                          Warning

The directory name selected /lib is a standard
UNIX system directory. If you overlay the contents of
this directory unpredictable things may happen to your
system. If you do intend to overlay this directory then
continue.  Otherwise CANCEL the operation and try again
by selecting a different directory name to mount your disk.




Strike CONT to continue or CANCEL to cancel the operation.
```

Striking CANCEL closes this frame and makes the "Mount File System" form active.

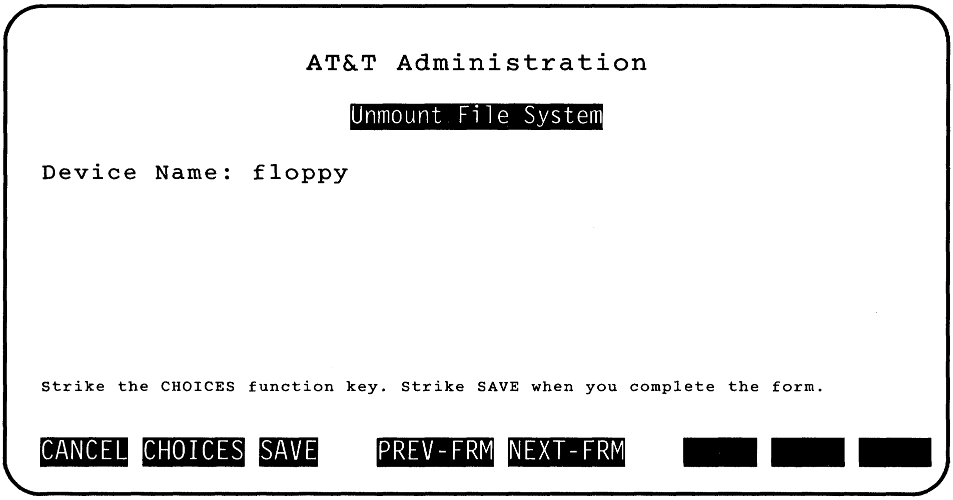
NOTE	Striking CONT will continue with the operation (i.e., the mount will be performed.)

Unmount File System

You may unmount a previously mounted UNIX system file system on a floppy diskette or a second hard disk if installed. When completed, the system will give you a confirmation message that the file system has been unmounted.

Use the following procedure to unmount a file system on a floppy diskette.

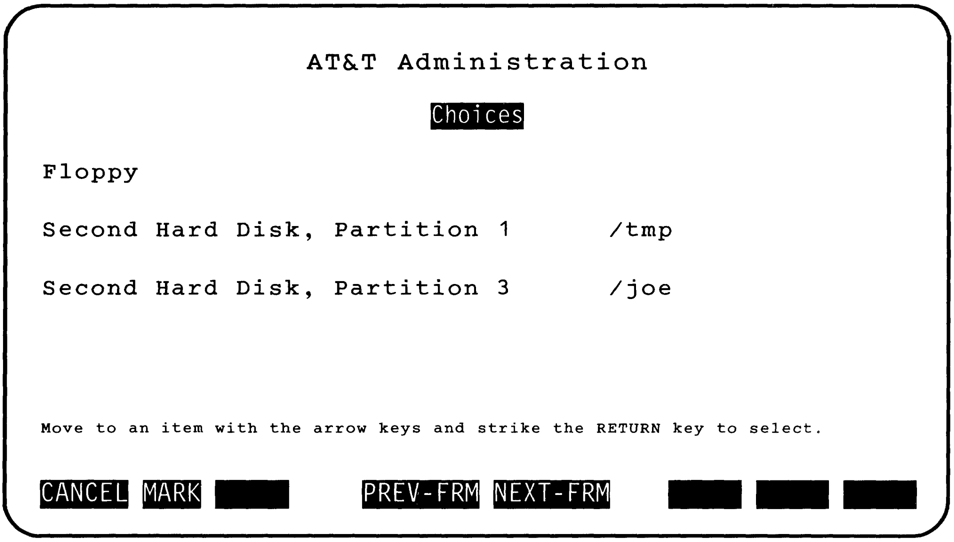
1. Highlight Unmount File System from the File System Operations menu and strike Enter. The Unmount File System form appears as follows:



```
                    AT&T Administration

                    Unmount File System

Device Name: floppy





Strike the CHOICES function key. Strike SAVE when you complete the form.
```

If your computer does not have a second hard disk set up, then the Choices pop-up menu will not appear on your screen and a floppy is assumed. Skip this step.

If your computer does have a second hard disk set up, floppy is assumed by default and the following example Choices pop-up menu appears.

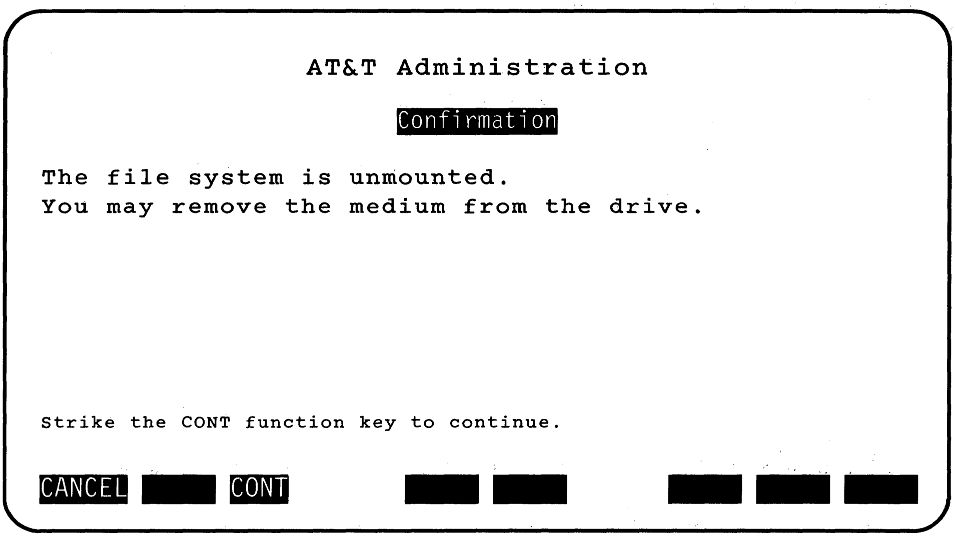


```
                    AT&T Administration

                          Choices

Floppy

Second Hard Disk, Partition 1      /tmp

Second Hard Disk, Partition 3      /joe



Move to an item with the arrow keys and strike the RETURN key to select.
```

NOTE Release 1 supports hard disks only connected to a single controller.

2. If the Unmount File System Form appears, select a device name from the second hard disk file system and strike Enter.

   The Confirmation text frame appears as follows:



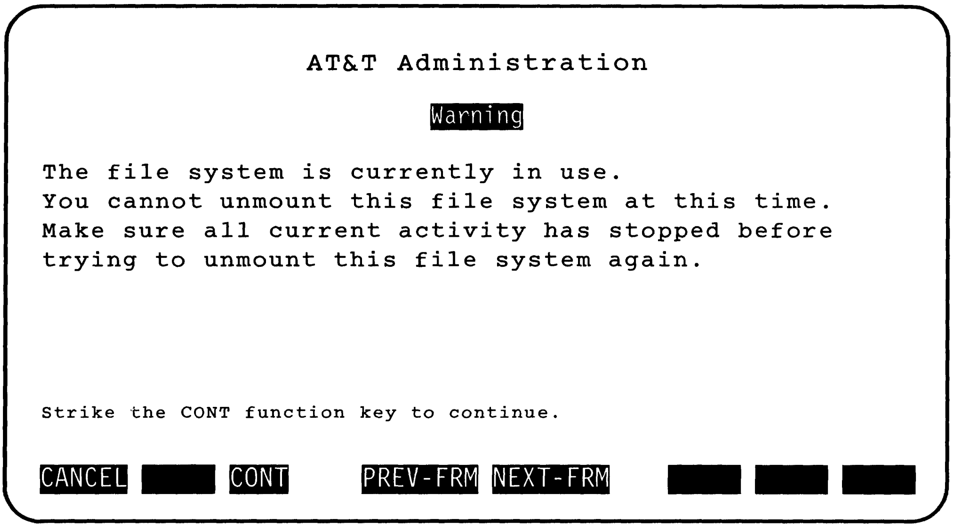
```
                    AT&T Administration

                        Confirmation

The file system is unmounted.
You may remove the medium from the drive.




Strike the CONT function key to continue.
```

3. Strike the CONT key to close this frame and make the "File System Operations" menu active.

Unmount File System Warning Messages

File System in Use

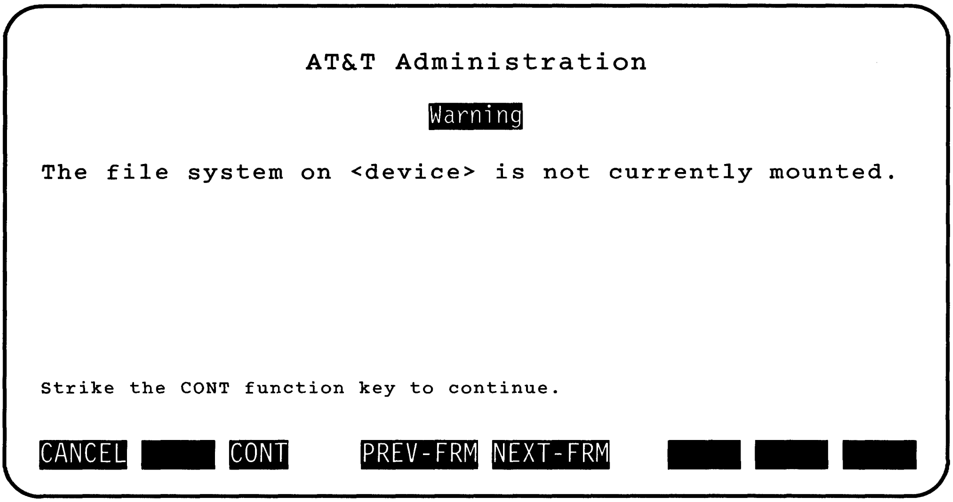
If the file system cannot be unmounted because the file system is in use, the following Warning message appears:



```
                    AT&T Administration

                          Warning

The file system is currently in use.
You cannot unmount this file system at this time.
Make sure all current activity has stopped before
trying to unmount this file system again.




Strike the CONT function key to continue.
```

File System Not Previously Mounted

If you attempt to unmount a file system not previously mounted, the following Warning message appears:



```
                    AT&T Administration

                          Warning

The file system on <device> is not currently mounted.





Strike the CONT function key to continue.
```

Striking CONT closes this frame and makes the "File System Operation" menu active.

Mail Setup

The Mail Setup menu item enables you to give your computer a node name so that other systems can send mail to your system and you can send mail to other systems.

To exchange electronic mail or files with another computer, you and the other computer user must set up and exchange some information. In general, you must do the following:

- Physically install the communicatin line to be used. You must network (connect) the computers together in some fashion such as a:

  modem

  direct connection

  data switch.

  You should exchange system names, mail names, passwords, data phone numbers if using a modem, and data communication line speeds.
- Assign your computer a mail name.
- Assign the mail login a password so that only other trusted systems can log in to your computer.
- Provide the users of other systems with your mail name and mail login password.
- Enter information about the computer receiving your mail (information you receive from the other computer administrator/operator).

If you have not installed the Electronic Mail feature, you can still use the basic UNIX system mail feature. Refer to the *UNIX System V User Guide* in the section "Communication Tutorial" for some instructions in using UNIX system mail. This document will not be included with the documents you are provided. To obtain a copy, consult your Documentation Roadmap for ordering information.

Set Up the Communication Line

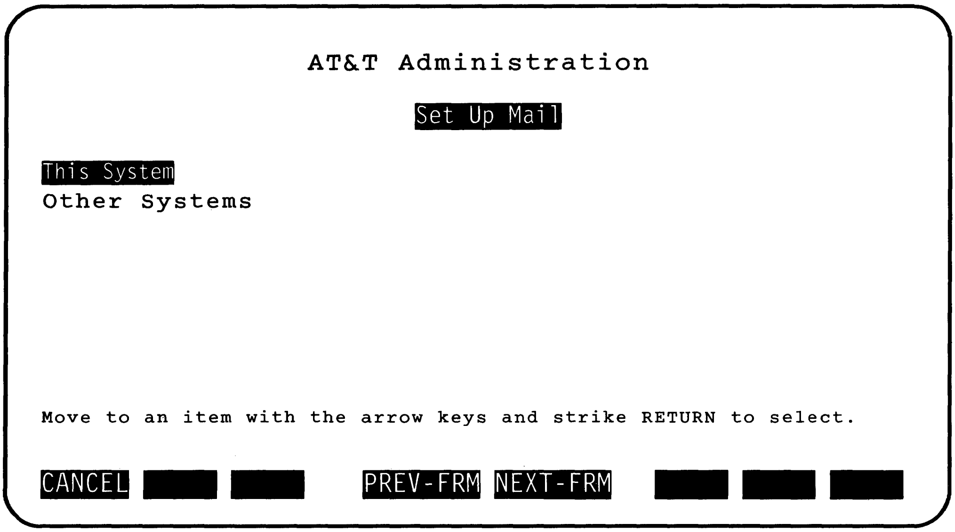
Before sending or receiving mail, you must configure the port using the Serial Ports Setup function from the Peripherals Setup menu in this chapter.

Configure the serial port for a modem, computer, or other (e.g., data switch) depending on the physical connection between your system and the other system.

Set Up This System to Receive Mail

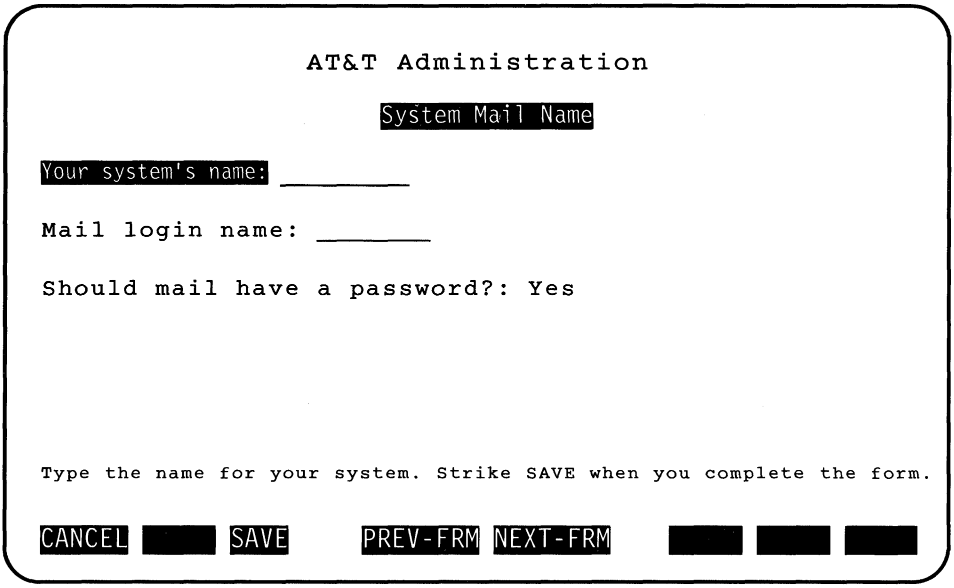
To set up your system to receive mail from other systems, use the following procedure:

1. Highlight `Mail Setup` and strike Enter from the Administration menu. The Mail Setup menu appears as follows:



Move to an item with the arrow keys and strike RETURN to select.

2. Highlight This System from the Mail Setup menu and strike (Enter). The System Mail Name form appears as follows:



```
                    AT&T Administration

                     System Mail Name

Your system's name: ________

Mail login name: ________

Should mail have a password?: Yes




Type the name for your system. Strike SAVE when you complete the form.
```

3. With the cursor resting on the "Your system's name:" field, type the name you want to call your system and strike Enter. The system name must be alphanumeric characters only, contain a maximum of eight characters, and cannot be the same name as another system.

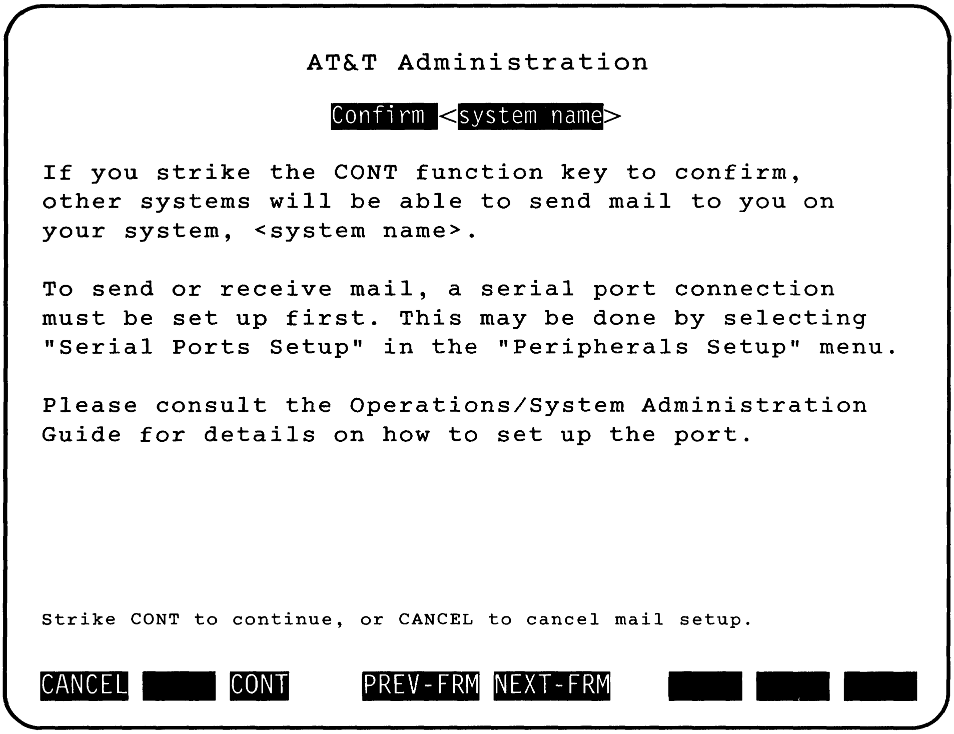
4. With the cursor resting on the "Mail login name:" field, type the mail login name for your system and strike Enter. The mail login must be alphanumeric characters only and contain a maximum of eight characters.

5. With the cursor resting on the "Should mail have a password?:" field, strike CHOICES. Toggle to Yes or No as appropriate.

   The mail password applies to the mail login that other systems will use to call your computer. If you change the password but do not change the login, the password will apply to the mail login that currently appears on the form.

6. Strike SAVE when you complete the form.

7. A confirmation will appear as follows:



```
                  AT&T Administration

                 Confirm <system name>

If you strike the CONT function key to confirm,
other systems will be able to send mail to you on
your system, <system name>.

To send or receive mail, a serial port connection
must be set up first. This may be done by selecting
"Serial Ports Setup" in the "Peripherals Setup" menu.

Please consult the Operations/System Administration
Guide for details on how to set up the port.




Strike CONT to continue, or CANCEL to cancel mail setup.
```

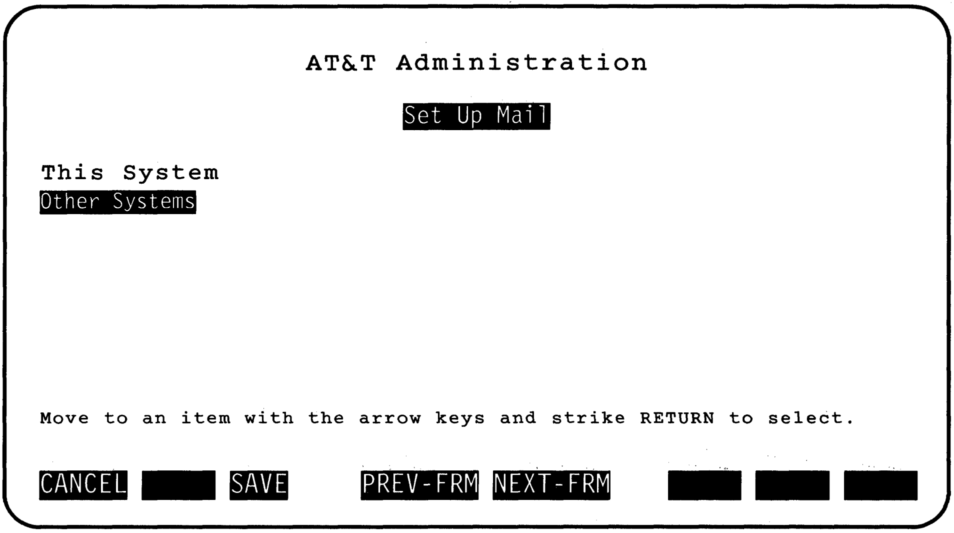
8. If you answered Yes to "Should mail have a password?:", the screen will clear when you strike CONT and you will be prompted for the mail password. If your system previously had a password, you will be prompted for the old password before you are prompted for the new password. The new password must differ from the old password by at least three characters, and it must not be a reversed or circular shift of the mail login.

9. After you give the new password, the interface will return and the active window will be the Mail Setup menu.

Set Up This System to Send Mail to Other Systems

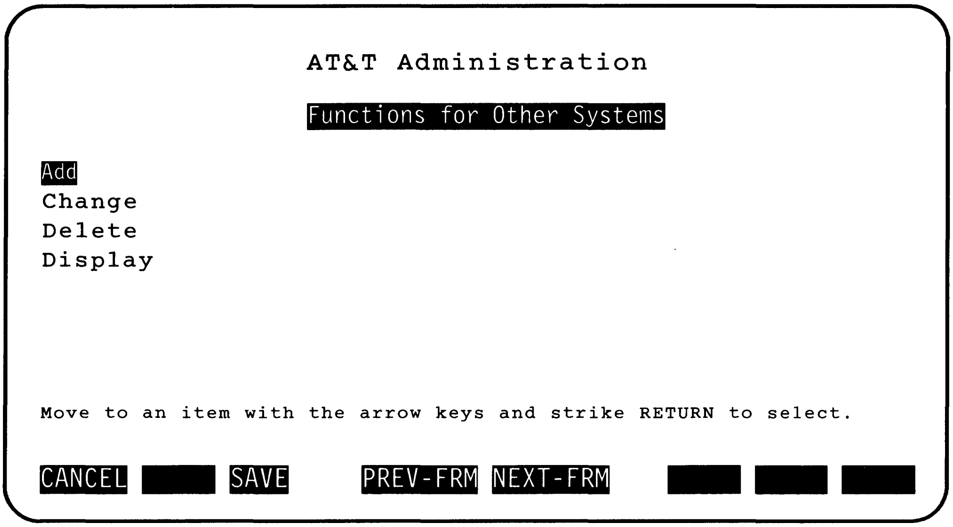
Set up your system to send mail to other systems by using the following procedure:

1. Highlight Mail Setup and strike Enter from the Administration menu. The Mail Setup menu appears as follows:



```
                    AT&T Administration

                        Set Up Mail

This System
Other Systems






Move to an item with the arrow keys and strike RETURN to select.

CANCEL        SAVE      PREV-FRM NEXT-FRM
```

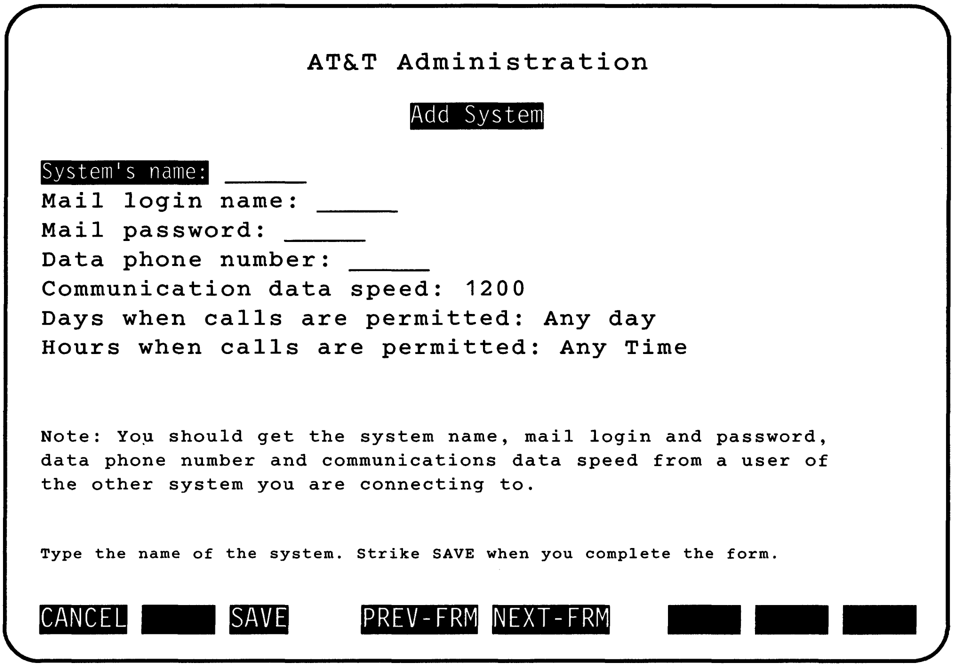
2. From the Set Up Mail menu, highlight Other Systems and strike Enter. The Functions for Other Systems menu appears as follows:



```
                    AT&T Administration

                 Functions for Other Systems

Add
Change
Delete
Display




Move to an item with the arrow keys and strike RETURN to select.

CANCEL        SAVE      PREV-FRM NEXT-FRM
```

3. Highlight Add and strike Enter. This adds information to the systems file about the computer you want to communicate with.

   The Add System form appears as follows:



```
                    AT&T Administration

                         Add System

System's name: ______
Mail login name: ______
Mail password: ______
Data phone number: ______
Communication data speed: 1200
Days when calls are permitted: Any day
Hours when calls are permitted: Any Time


Note: You should get the system name, mail login and password,
data phone number and communications data speed from a user of
the other system you are connecting to.


Type the name of the system. Strike SAVE when you complete the form.
```

4. With the cursor resting on the "System's name" field, enter the name of the other system. You can get the system's name from users of the "other" system. The system name must be alphanumeric characters only, contain a maximum of eight characters, and cannot be the same name as another system.

5. With the cursor resting on the "Mail login name" field, enter the mail login name. User's of the other system should provide you with the mail login name. The mail login name is actually the **uucp** login for the other system. The mail login name must be alphanumeric characters only and contain a maximum of eight characters.

6. With the cursor resting on the "Mail password" field, enter the system's mail password. User's of the other system should provide you with the mail password. This is actually the **uucp** password for the other system. The mail password must be alphanumeric characters only and contain a maximum of eight characters.

7. With the cursor resting on the "Data phone number" field, enter the phone number of the other system if your systems are connected by a modem and telephone lines. The phone number cannot contain any white space. The format of the Data phone number could be as follows:

   9=1209329329-

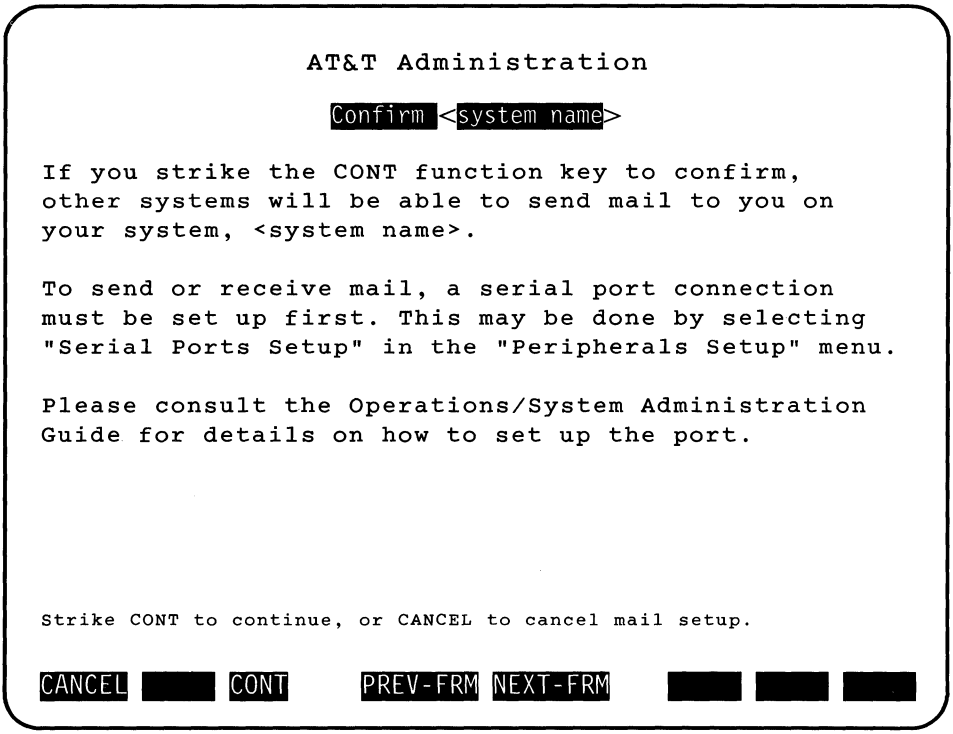
   The first 9 is used to dail a number outside of the building. The "=" pauses for a dial tone. The next three digits (120) are the area code. The last seven digits are the telephone number. The "-" causes a delay before hanging up. This is useful for long distance calls, since it might take some time to make the connection.

   If your systems are connected by a data switch or direct line, enter the system name instead of the telephone number here.

8. With the cursor resting on the "Communication data speed" field, strike CHOICES and select the desired data line speed from the pop-up menu.

NOTE	The Communication data speed must be the same as the device speed for the serial port that you set up through Peripherals Setup.

9. With the cursor resting on the "Days when calls are permitted" field, strike CHOICES and select the desired days from the pop-up menu. Calls will only be made to the other system on the days that you specify.
10. With the cursor resting on the Hours when calls are permitted" field, strike CHOICES and select the desired hours from the pop-up menu. Calls will only be made to the other system during the hours that you specify.
11. Strike SAVE when you have provided input for all fields.
12. When the confirmation text frame appears, showing the information you entered, check to make sure the information is correct. This frame appears as follows:



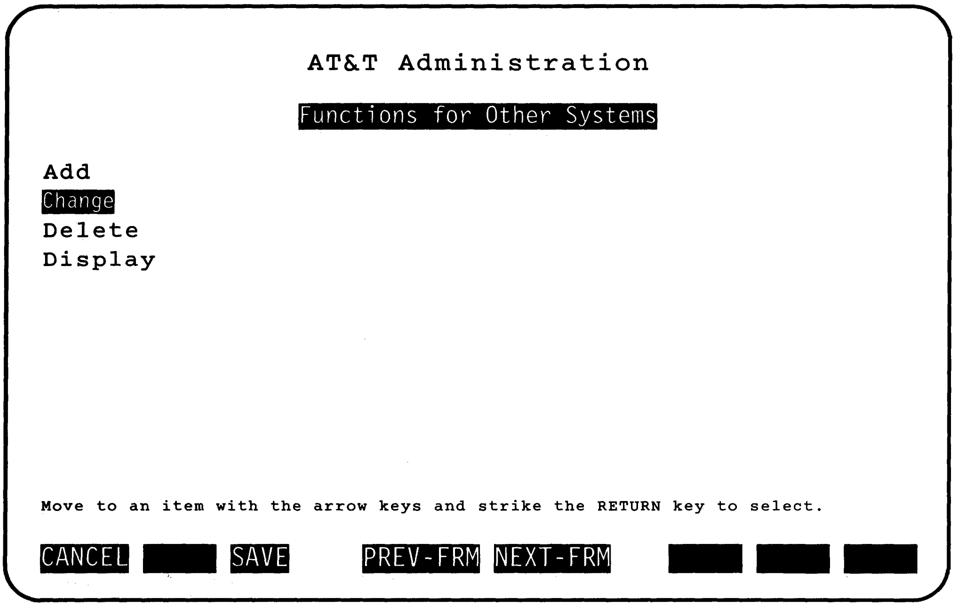
```
                    AT&T Administration

                   Confirm <system name>

If you strike the CONT function key to confirm,
other systems will be able to send mail to you on
your system, <system name>.

To send or receive mail, a serial port connection
must be set up first. This may be done by selecting
"Serial Ports Setup" in the "Peripherals Setup" menu.

Please consult the Operations/System Administration
Guide for details on how to set up the port.




Strike CONT to continue, or CANCEL to cancel mail setup.

CANCEL          CONT     PREV-FRM NEXT-FRM
```

13. Strike CONT to continue, then strike CANCEL to cancel the mail setup.

Change Setup for Other Systems

To change other systems, do the following:

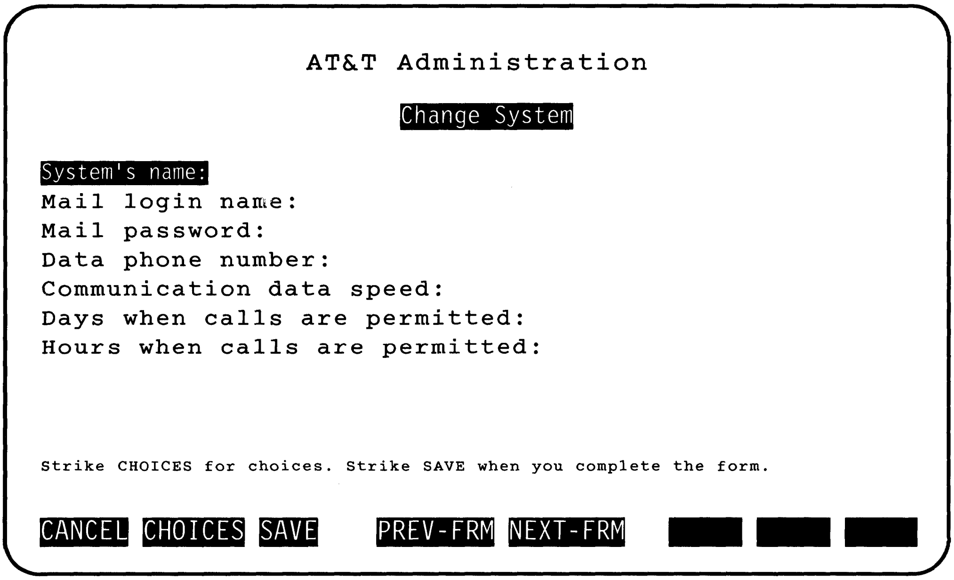
1. Highlight Other Systems from the Mail Setup menu and strike [Enter]. The Functions for Other Systems menu appears as follows:



```
                    AT&T Administration

                Functions for Other Systems

Add
Change
Delete
Display

Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL        SAVE       PREV-FRM NEXT-FRM
```

2. Highlight Change and strike [Enter].

   The Change System form appears as follows:



```
                    AT&T Administration

                         Change System

System's name:
Mail login name:
Mail password:
Data phone number:
Communication data speed:
Days when calls are permitted:
Hours when calls are permitted:




Strike CHOICES for choices. Strike SAVE when you complete the form.
```

3. With the cursor resting on the "System's name" field, strike the CHOICES key, and then select the system to be changed from the list on the pop-up menu that appears.

   If you type the system name and system does not exist, the following message appears:

   ```
   That system does not exist. Type another system name.
   ```

   Once you have selected a system name, the remaining fields on the form will be filled in with their existing values.

4. If the other system's mail login name has changed, you need to change the mail login name. Move the cursor to the "Mail login name:" field and enter the changed mail login name. The mail login name must be alphanumeric characters only and contain a maximum of eight characters.

5. If the other system's mail password has changed, you need to change the mail password. User's of the other system should provide you with the new mail password. Move the cursor to the Mail password" field, enter the system's mail password. The mail password must be

alphanumeric characters only and contain a maximum of eight characters.

6. When the phone number has been changed for the other system, you need to change the data phone number if your systems are connected by modems and telephone lines. Move the cursor to the "Data phone number:" field; enter the phone number of the other system. The phone number cannot contain any white space. The format of the Data phone number could be as follows:

   9=1209329329-

   The first 9 is used to dail a number outside of the building. The "=" pauses for a dial tone. The next three digits (120) are the area code. The last seven digits are the telephone number. The "-" cause a delay before hanging up. This is useful for long distance calls, since it might take some time to make the connection.

   If your systems are connected by a data switch or direct line and the System's name has changed, enter the new system name in the "Data phone number:" field.

7. If the speed of the modem (as provide by user's of the other system) changes, you need to change the data communications speed. Move the cursor to the "Communication data speed:" field , strike CHOICES and select the desired data line speed from the pop-up menu.

8. To change the days when calls are permitted, move the cursor to the "Days when calls are permitted:" field, strike CHOICES, and select the desired days from the pop-up menu. Calls will only be made to the other system on the days that you specify.

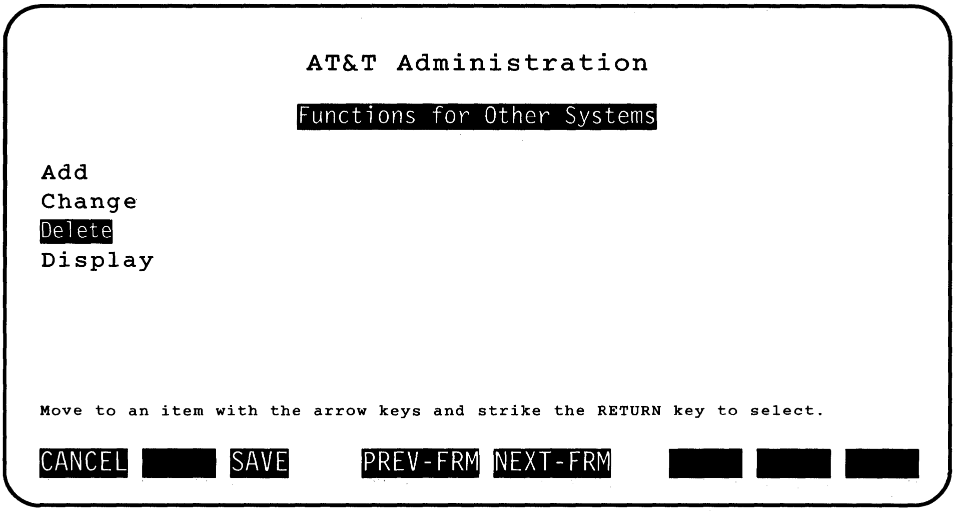
9. To change the hours when calls are permitted, move the cursor to the "Hours when calls are permitted:" field, strike CHOICES, and select the desired hours from the pop-up menu. Calls will only be made to the other system during the hours that you specify.

10. Strike SAVE when you have changed all appropriate fields.

Delete Other Systems (Previously Set Up)

To delete other systems, do the following:

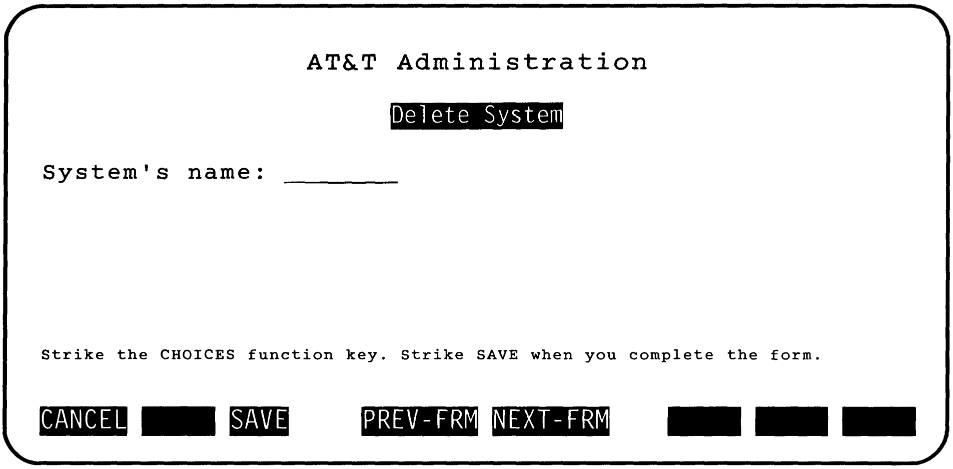
1. From the Mail Setup menu, highlight Other Systems and strike Enter. The Functions for Other Systems menu appears as follows:



```
                     AT&T Administration

                 Functions for Other Systems

Add
Change
Delete
Display




Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL        SAVE      PREV-FRM NEXT-FRM
```

2. Highlight Delete and strike Enter.

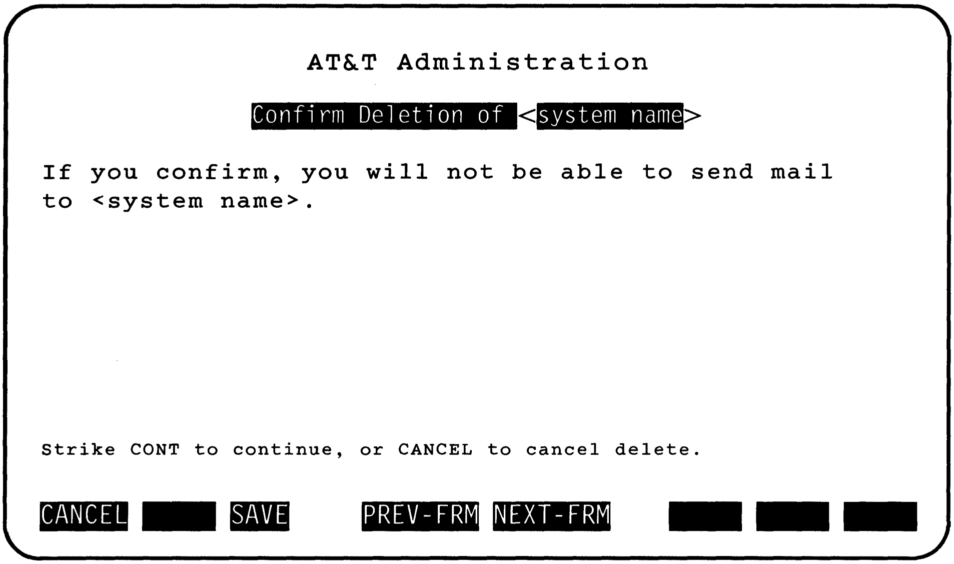
   The Delete System form appears as follows:



```
                    AT&T Administration

                       Delete System

System's name: ________




Strike the CHOICES function key. Strike SAVE when you complete the form.
```

3. With the cursor resting on the "System's name" field, strike the CHOICES key and select the system from the pop-up menu that appears. If less than three systems have previously been added, the system name appears in the field. Strike SAVE the system name you want to delete appears on the form.

4. When the confirmation text frame appears to show the system you will delete if you confirm, check to make sure the information is correct. This frame appears as follows:



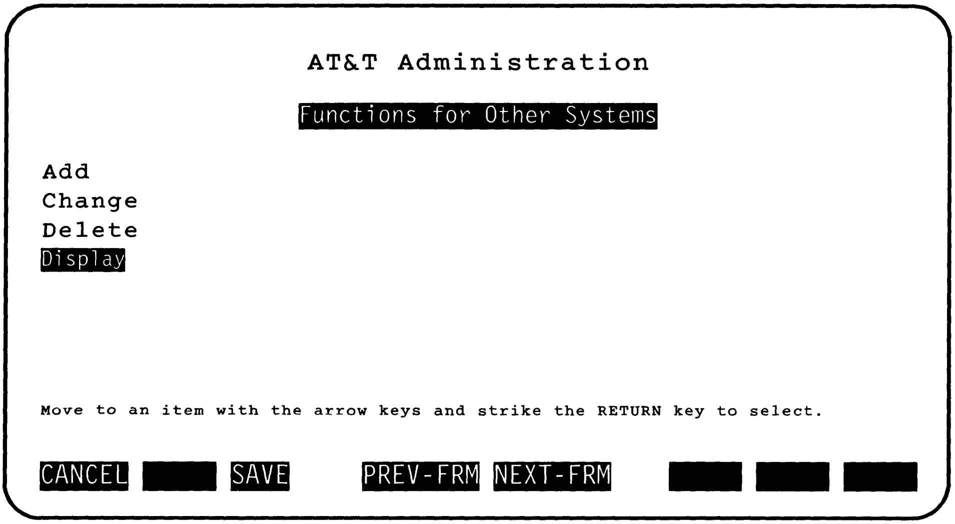
```
                    AT&T Administration

              Confirm Deletion of <system name>

If you confirm, you will not be able to send mail
to <system name>.







Strike CONT to continue, or CANCEL to cancel delete.
```

5. Strike CONT if the information is correct. If the information is not correct, strike CANCEL to stop the process of deleting the other system and return to the Mail Setup menu.

Display Other Systems (Previously Set Up)

To display other systems, do the following:

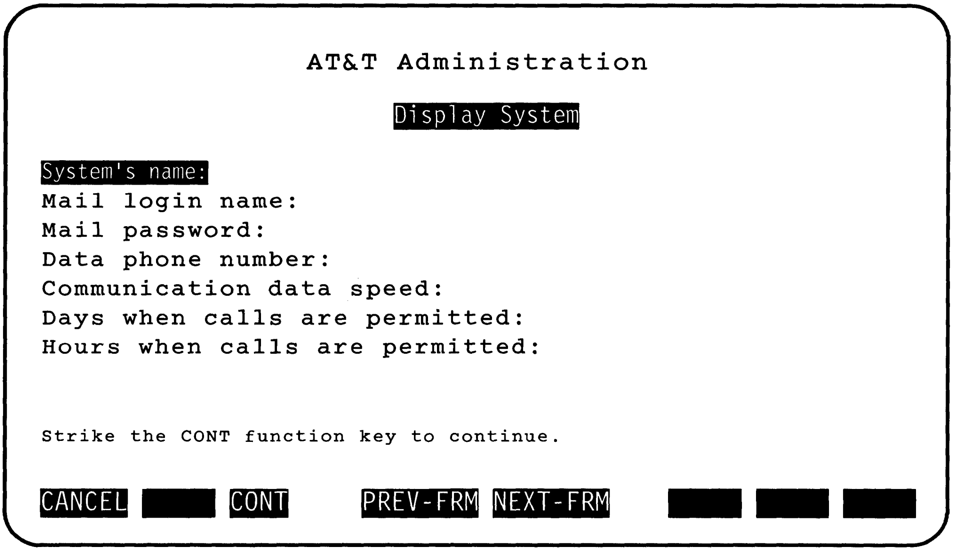
1. Highlight Other Systems from the Mail Setup menu and strike Enter. The Functions for Other Systems menu appears as follows:



Move to an item with the arrow keys and strike the RETURN key to select.

2. Highlight Display and strike Enter.

   The Display System form appears as follows:



```
                    AT&T Administration

                       Display System

System's name:
Mail login name:
Mail password:
Data phone number:
Communication data speed:
Days when calls are permitted:
Hours when calls are permitted:


Strike the CONT function key to continue.
```

3. Strike the CHOICES function key. If you have more than three systems, a list of systems appear. Select a system and the remaining fields on the form will be filled in. If you have added three or less systems, you can toggle between the systems.

   You can change the values of the fields, but no changes will be made in the systems file.

4. When finished, strike CONT to continue to the mail setup menu.

Peripherals Setup

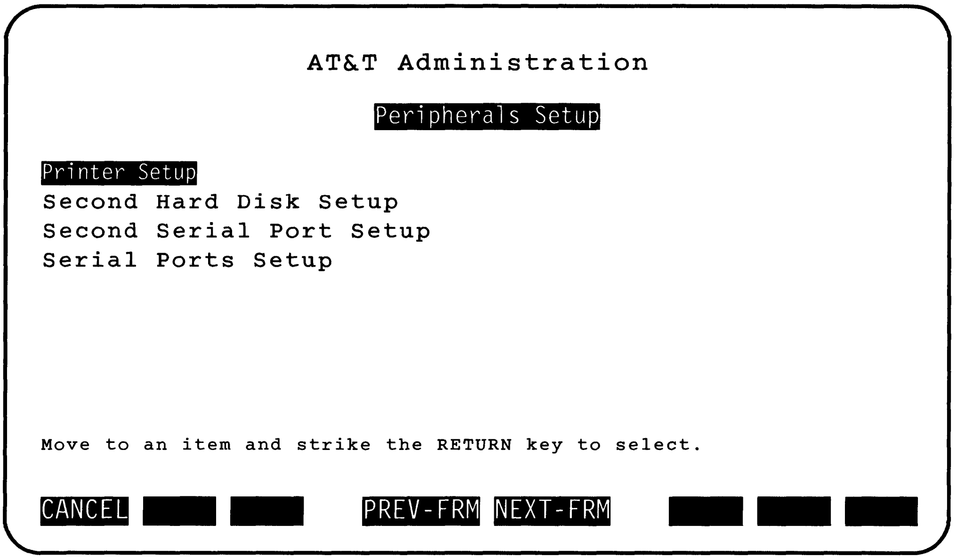
The peripherals setup function allows you to set up the computer to support your printers, second hard disk, and additional parallel and serial port connections. The serial ports can be used for terminals, modems, a serial printer, or a connection link to another computer. The parallel ports can only be used with a printer.

Printer Setup

To connect a printer to your computer, you must tell the system which type of printer (printer model) you are connecting and whether the printer is serial or parallel. Unless a ports board is present, the Printer Setup menu item assumes there is only one parallel port in the machine. Parallel printers connect to the printer port labeled **Parallel**; serial printers connect to the serial port labeled **RS-232**.

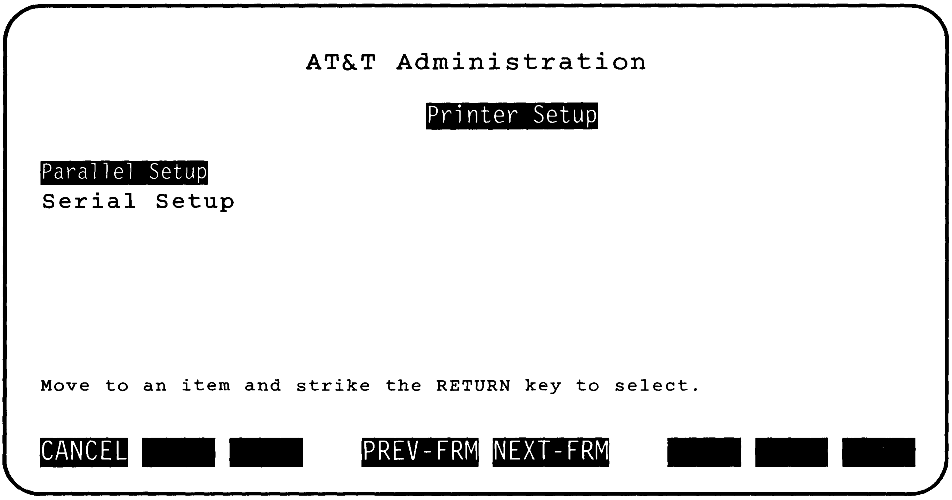
Before you begin, physically connect your printer to the serial or parallel port, then use the following procedure.

1. From the Administration menu, highlight `Peripherals Setup` and strike Enter. The Peripherals Setup menu appears as follows.



```
                    AT&T Administration

                     Peripherals Setup

Printer Setup
Second Hard Disk Setup
Second Serial Port Setup
Serial Ports Setup




Move to an item and strike the RETURN key to select.

CANCEL              PREV-FRM NEXT-FRM
```

2. Highlight Printer Setup and strike Enter. The Printer Setup menu appears as follows.



```
                    AT&T Administration

                         Printer Setup

Parallel Setup
Serial Setup






Move to an item and strike the RETURN key to select.
```

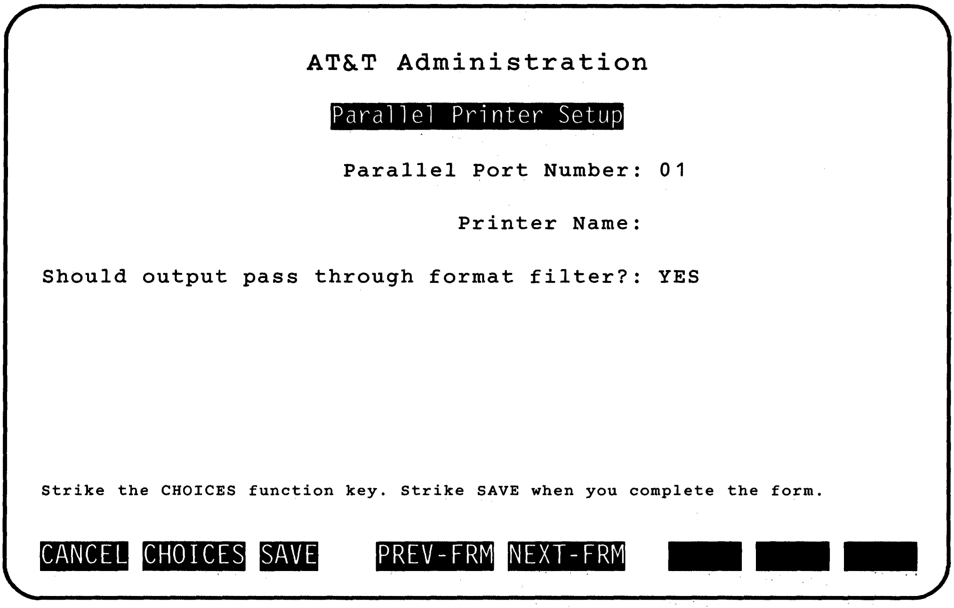
3. Highlight the type of printer you are connecting, Parallel Setup for a parallel printer or Serial Setup for a Serial printer and strike Enter.

Parallel Printer Setup

NOTE

If you did not choose Parallel Setup, skip to the Serial Printer Setup. The following steps pertain to Parallel Printer Setup only.

If you chose Parallel Setup, the Parallel Printer Setup Form appears as follows:



```
                    AT&T Administration

                   Parallel Printer Setup

                  Parallel Port Number: 01

                          Printer Name:

Should output pass through format filter?: YES




Strike the CHOICES function key. Strike SAVE when you complete the form.
```

NOTE: Either "None" or the "printer name" previously configured for the port appears in the "Printer Name:" field.

1. Select a valid port number. If no add-on multi-ports board(s) have been installed, port 01 is the only valid port number. If you have installed the multi-ports board(s), there are additional valid port numbers. There are only two parallel ports per additional multi-ports board. For example, if one additional multi-ports board has been installed then, the choice port 01 through port 03 will appear.

   The default value for the "Parallel Port Number:" field is 01. If this value is appropriate, skip this step.

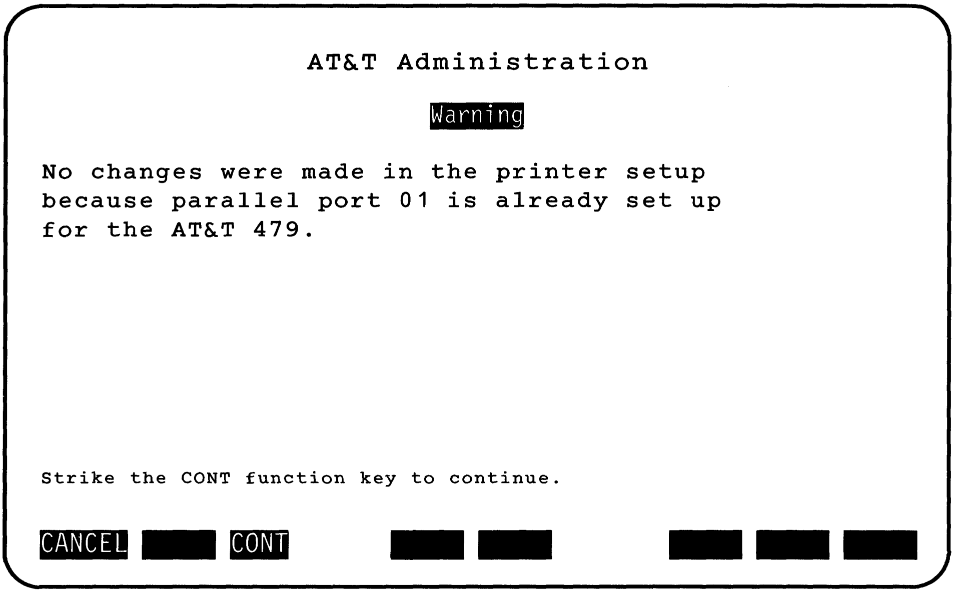
   If you want to configure a port other than port 01, strike the CHOICES function key. If a multi-ports board has been installed, the Parallel Port Number pop-up menu appears.

Select the appropriate port number, and strike [Enter].

2. If the port is currently free or available, the value "None" will appear in the "Printer Name:" field.
3. If a printer is already set up for your parallel port, the name will appear in the "Printer Name:" field. If you want to set up a different printer strike CHOICES for a list of printers. The Printer Names pop-up menu appears.
4. Use the arrow keys to scroll throught the list of the available printers; Highlight the name of the printer model you are connecting; and strike [Enter].

   The name of the printer model you selected appears in the "Printer Name:" field.

   If you do not change the printer name, you will receive a message similar to the following sample.



```
                    AT&T Administration

                          Warning

No changes were made in the printer setup
because parallel port 01 is already set up
for the AT&T 479.






Strike the CONT function key to continue.
```

5. With the cursor resting on the "Should output pass through format filter?:" field, select either Yes or No depending on whether the printer output needs to be filtered to display appropriately for your printer type. You need to respond Yes if you use a UNIX system application package that generates printer control sequences such as graphics mode. In most cases, the answer to this prompt will be Yes. The only cases where you may need to answer No is where the application itself is doing its own printer filtering. Most of these applications are MS-DOS applications. Very few UNIX system applications do their own filtering.

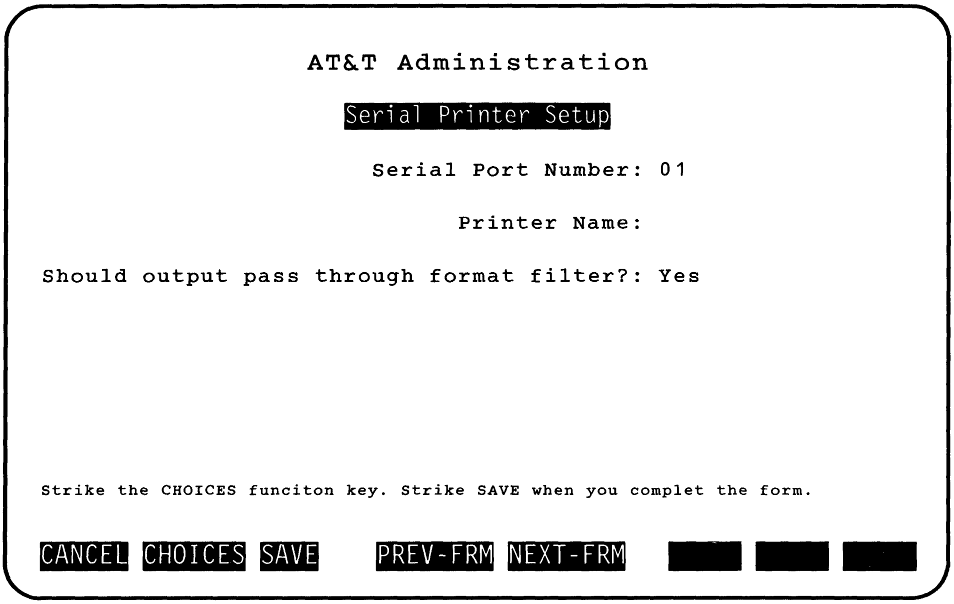
   Otherwise, strike CHOICES and toggle (Yes or No) until your choice appears in the field. Then strike Enter.

6. Strike SAVE and a confirmation message appears telling you that the printer you selected is set up on the port you selected. Also, the printer will be set up as your default printer destination.

   Strike CONT to make the "Peripherals Setup" menu active.

Serial Printer Setup

If you chose Serial Setup, the Serial Printer Setup form appears as follows.



```
                    AT&T Administration

                    Serial Printer Setup

                    Serial Port Number: 01

                         Printer Name:

Should output pass through format filter?: Yes




Strike the CHOICES funciton key. Strike SAVE when you complet the form.
```

NOTE The following steps pertain to Serial Printer Setup only. Either "None" or the "printer name" previously configured for the port appears in the "Printer Name:" field.

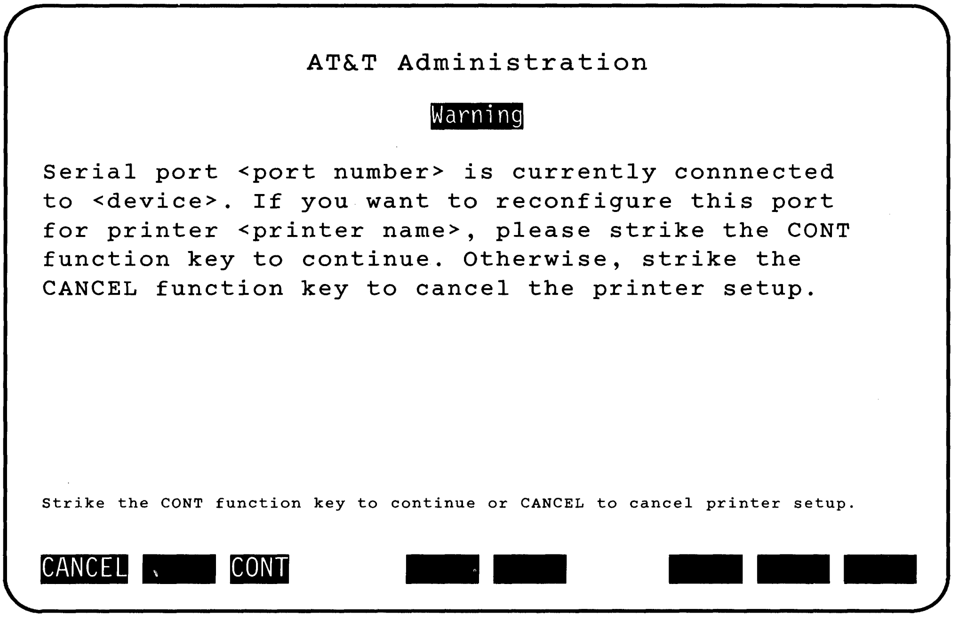
1. Select a valid port number. If no add-on multi-ports board(s) have been installed, port 01 is the valid port number. There are eight serial ports per additional multi-ports board. If one multi-ports board is added, then ports 01, 03 - 10 appears. Port 02 is a special case because it is the second internal serial port (COM2). The "Second Serial Port Setup" entry in the Peripherals Setup menu is used to enable this port (once hardware has been installed in the port). Once this port is enabled, it appears as a CHOICE.

   The default value for the "Serial Port Number:" field is 01. If this value is appropriate, skip this step.

If you want to configure a port other than port 01, strike the CHOICES function key. If a multi-ports board has been installed the Serial Port Number pop-up menu appears. Select the appropriate port number, and strike Enter.

Port Already Configured

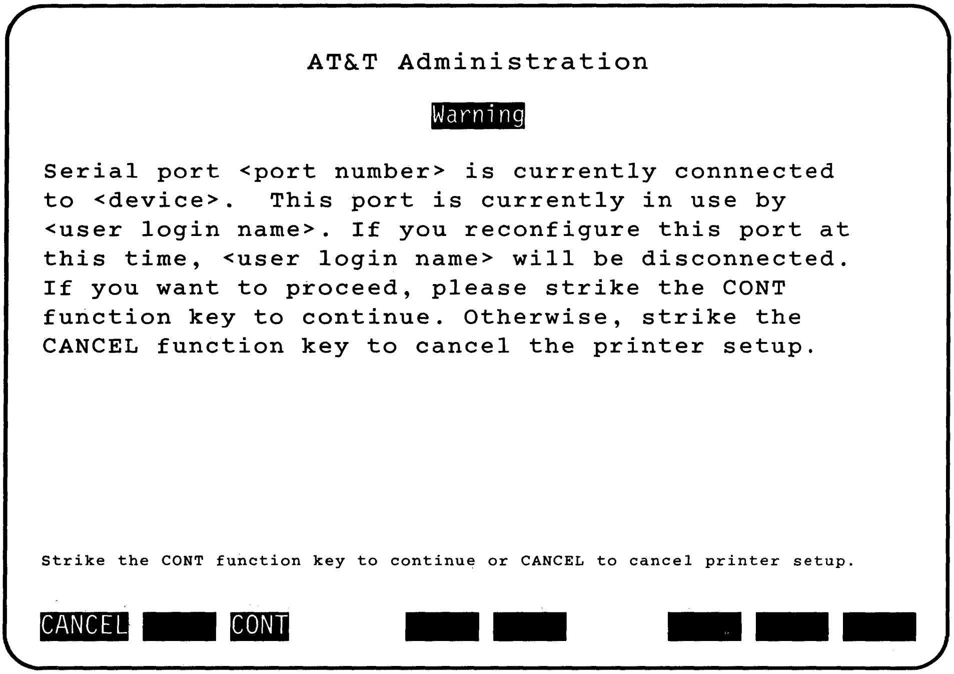
When you select a port number, if a device is connected to the port (already configured or connected to a device different from printer) the following Warning messages appear depending upon whether or not the port is currently in use.



```
                    AT&T Administration

                          Warning

Serial port <port number> is currently connnected
to <device>. If you want to reconfigure this port
for printer <printer name>, please strike the CONT
function key to continue. Otherwise, strike the
CANCEL function key to cancel the printer setup.

Strike the CONT function key to continue or CANCEL to cancel printer setup.
```

Port Already Configured and Currently in Use

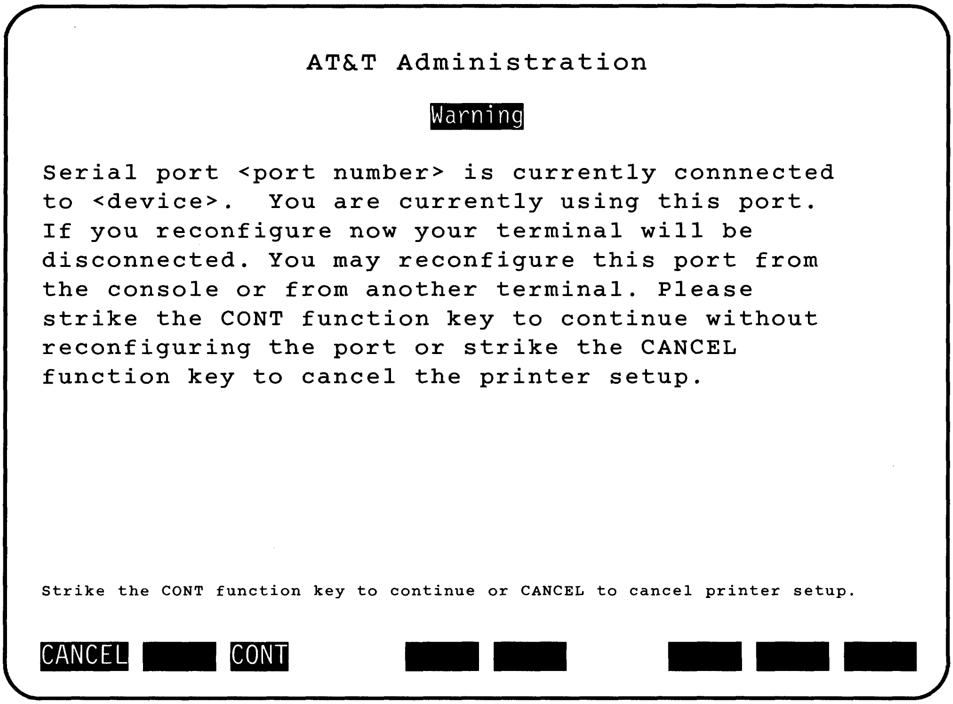
If the serial port being reconfigured is currently being used by a user different from the invoking user, the invoking user will be notified and asked to confirm if he or she wants to continue. The following message appears instead of the previous warning.



```
                    AT&T Administration

                          Warning

Serial port <port number> is currently connnected
to <device>.  This port is currently in use by
<user login name>. If you reconfigure this port at
this time, <user login name> will be disconnected.
If you want to proceed, please strike the CONT
function key to continue. Otherwise, strike the
CANCEL function key to cancel the printer setup.





Strike the CONT function key to continue or CANCEL to cancel printer setup.
```

Port Already Configured and Currently in Use by the Invoking User

If the serial port being reconfigured is currently in use by the invoking user, the following message appears instead of the previous ones.



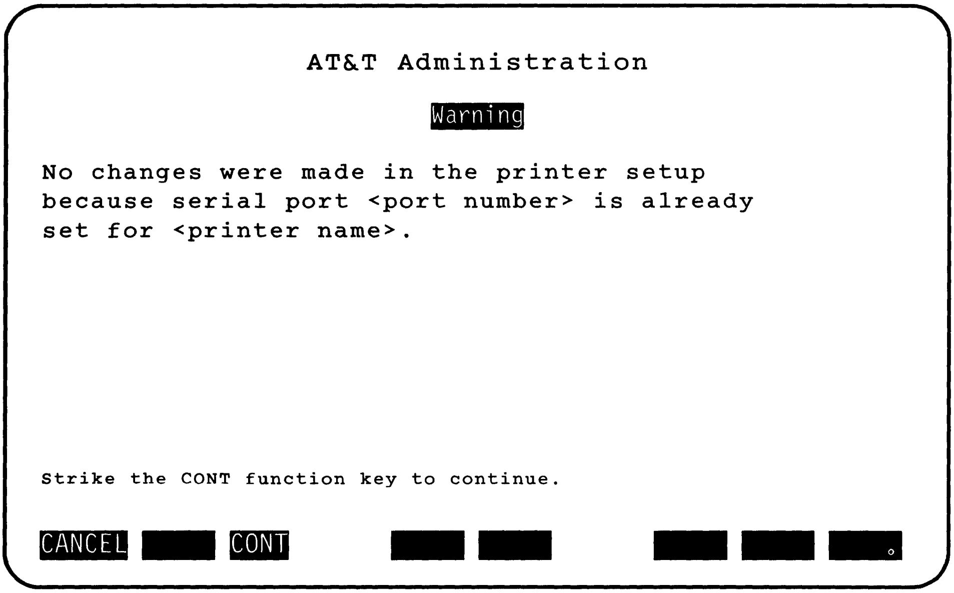
```
                         AT&T Administration

                               Warning

Serial port <port number> is currently connnected
to <device>.  You are currently using this port.
If you reconfigure now your terminal will be
disconnected. You may reconfigure this port from
the console or from another terminal. Please
strike the CONT function key to continue without
reconfiguring the port or strike the CANCEL
function key to cancel the printer setup.






Strike the CONT function key to continue or CANCEL to cancel printer setup.
```

2. If a printer is already set up for your serial port, the name will appear in the "Printer Name:" field. If you want to set up a different printer, strike CHOICES for a list of printers. The Printer Names pop-up menu appears.

3. Use the arrow keys to scroll through the list of the available printers; highlight the name of the printer model you are connecting; and strike Enter.

   The name of the printer model you selected appears in the "Printer Name:" field.

   If you do not change the printer name you will receive a message similar to the following sample:



```
AT&T Administration

Warning

No changes were made in the printer setup
because serial port <port number> is already
set for <printer name>.

Strike the CONT function key to continue.
```

4. With the cursor resting on the "Should output pass through format filter?:" field, select either YES or NO depending on whether the printer output needs to be filtered to display appropriately for your printer type. You need to respond Yes if you use a UNIX system application package that generates printer control sequences such as graphics mode. In most cases, the answer to this prompt will be Yes. The only cases where you may need to answer No is where the application itself is doing its own printer filtering. Most of these applications are MS-DOS applications. Very few UNIX system applications do their own filtering.

   Otherwise, strike CHOICES and toggle (Yes or No) until your choice appears in the field. Then strike [Enter].

5. Strike SAVE and a confirmation message appears telling you that the printer you selected is now set up on the port you selected. Also, the printer will be set up as your default printer destination.

   Striking CONT makes the "Peripherals Setup" menu active.

Refer to the documentation supplied with your printer for proper DIP switch settings.

Second Serial Port Setup

The Second Serial Port Setup feature configures additional serial connection over that provided by the built-in serial port on your computer. This will allow for a total of two serial devices: asynchronous terminals and/or serial printers. You can make use of existing PC/XT internal modem and multifunction boards (plug-in boards which provide a serial and/or parallel interface using the COM2 interrupt.)

You must plan how you want to use your COM2 interrupt. You may decide to add expansion hardware which may use the COM2 interrupt such as the AT&T StarLAN Network Access Unit (NAU) which can be configured to use the COM2 interrupt.

To create a second serial port to connect to serial devices, you must (1) make the physical connection using a cable with RS-232 connectors, and (2) set up (configure) your system to recognize the type of connection you're making through the AT&T Administration interface.

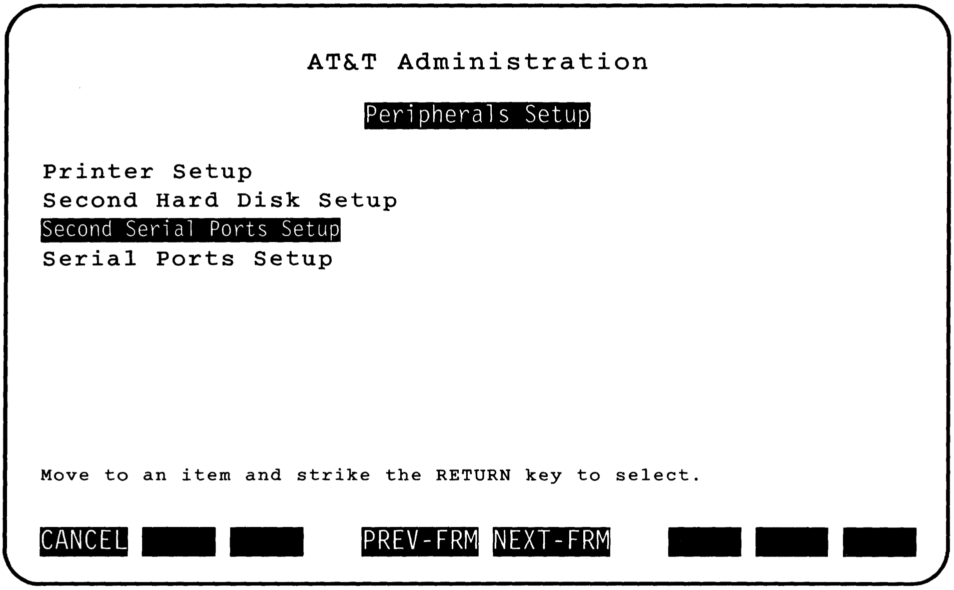
NOTE

This section addresses only the steps involved in defining a second serial port to the system. To configure the port with a modem, printer, terminal, or another computer refer to section Serial Port Setup.

Add Second Serial Port

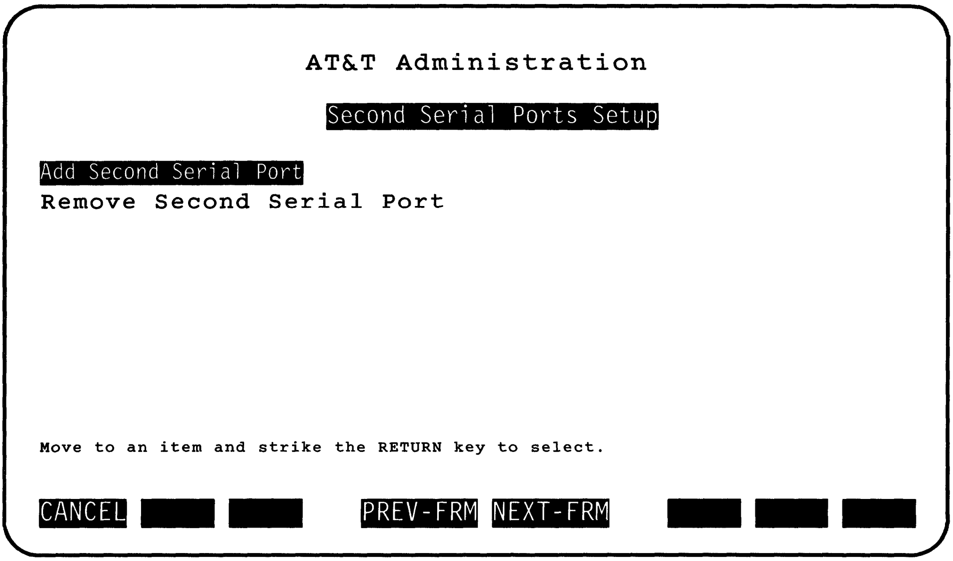
To set up the second serial port, use the following procedure.

1. From the Administration menu, highlight `Peripherals Setup` and strike [Enter]. The Peripherals Setup menu appears as follows:



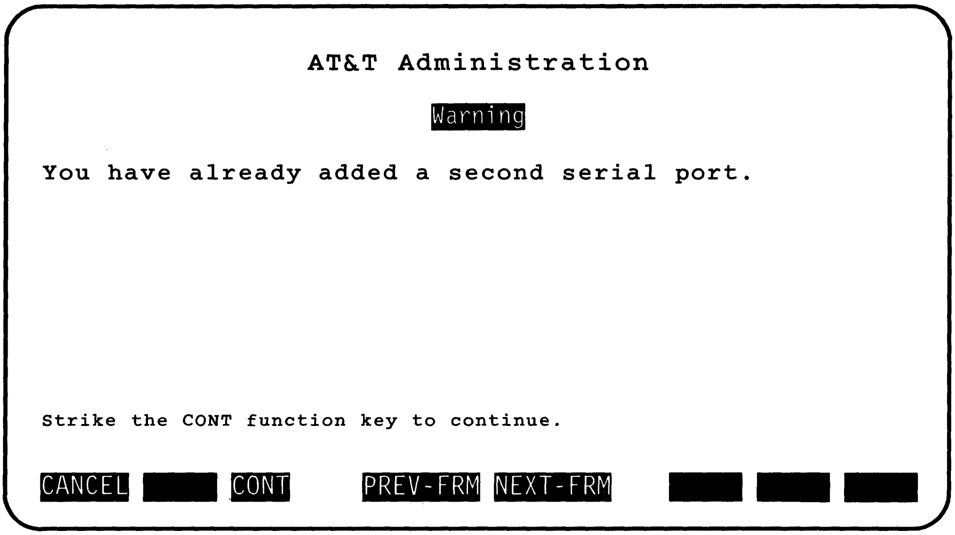
```
                    AT&T Administration

                     Peripherals Setup

Printer Setup
Second Hard Disk Setup
Second Serial Ports Setup
Serial Ports Setup







Move to an item and strike the RETURN key to select.

CANCEL                 PREV-FRM NEXT-FRM
```

2. Highlight Second Serial Ports Setup from the Peripherals Setup menu and strike Enter. The Second Serial Ports Setup form appears as shown in the following sample:



```
                        AT&T Administration

                     Second Serial Ports Setup

Add Second Serial Port
Remove Second Serial Port







Move to an item and strike the RETURN key to select.
```

3. Highlight Add Second Serial Port from the Second Serial Ports Setup menu and strike Enter.
4. If you are trying to add a second serial port and the second serial port is already configured, the following message appears:

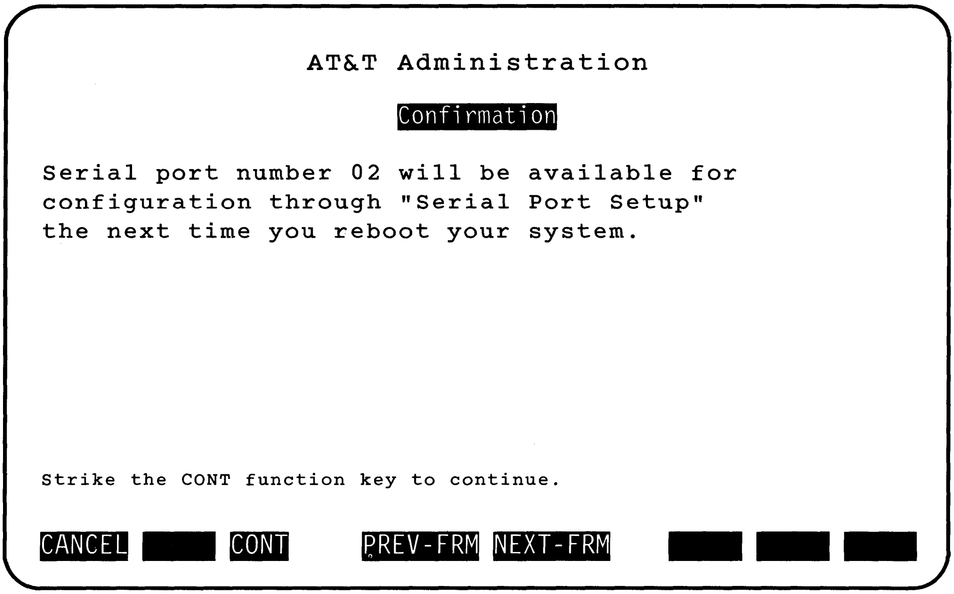


```
AT&T Administration

Warning

You have already added a second serial port.

Strike the CONT function key to continue.
```

Otherwise, a second serial port is configured for the COM2 interrupt and you will receive the following message:

```
Rebuild UNIX system takes approximately 2 minutes.
Please wait ...
```

After the UNIX system is rebuilt, the following confirmation message appears:



```
                    AT&T Administration

                        Confirmation

Serial port number 02 will be available for
configuration through "Serial Port Setup"
the next time you reboot your system.








Strike the CONT function key to continue.
```

5. Next, you will receive a message asking if you want to reboot the system. If you enter 'y' the automatic shutdown proceeds or if you enter 'n' the shutdown is canceled.

NOTE

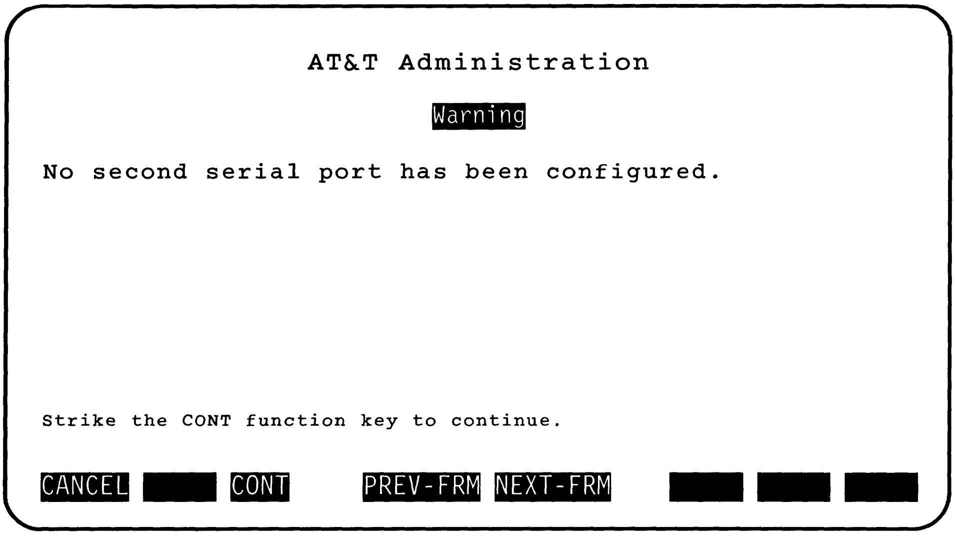
If you cancel the shutdown, you will be returned to the Peripherals Setup menu. However, port number 02 will not be available throught the Serial Port Setup menu until reboot occurs.

After the subsequent reboot, the second serial port will be set up. The next time you select Serial Port Setup in the Peripherals Setup menu, port number 02 will be a valid choice.

Remove Second Serial Port

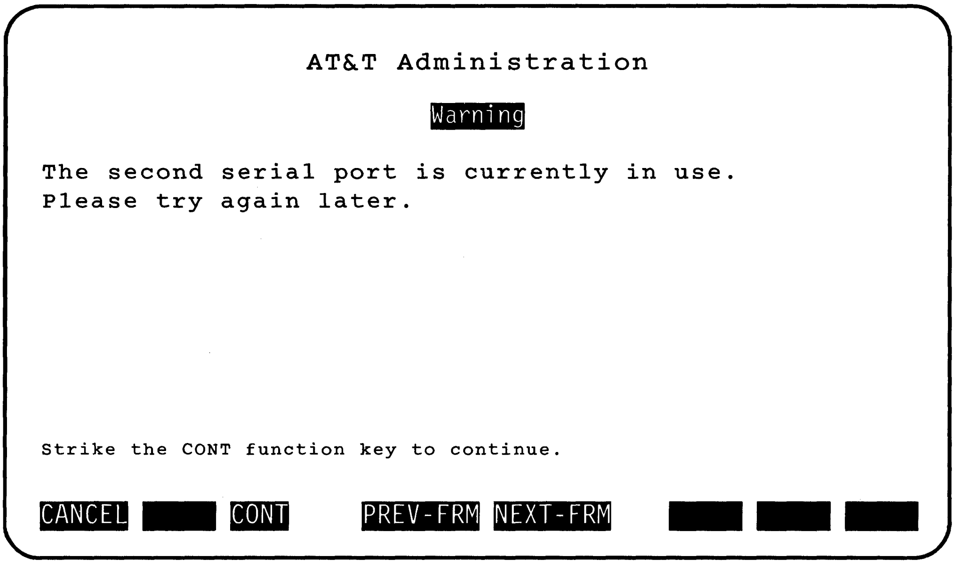
To remove a previously configured second serial port, use the following procedure.

1. From the Administration menu, highlight Peripherals Setup and strike Enter.
2. Highlight Second Serial Port Setup from the Peripherals Setup menu and strike Enter.
3. Highlight Remove Second Serial Port from the Second Serial Port Setup menu and strike Enter.
4. If the second serial port has not been enabled, the following message appears and you are returned to the Peripherals Setup menu.



```
                    AT&T Administration

                          Warning

No second serial port has been configured.





Strike the CONT function key to continue.

CANCEL       CONT      PREV-FRM NEXT-FRM
```

5. If the second serial port is currently in use, the following message appears and you are returned to the Peripherals Setup menu.

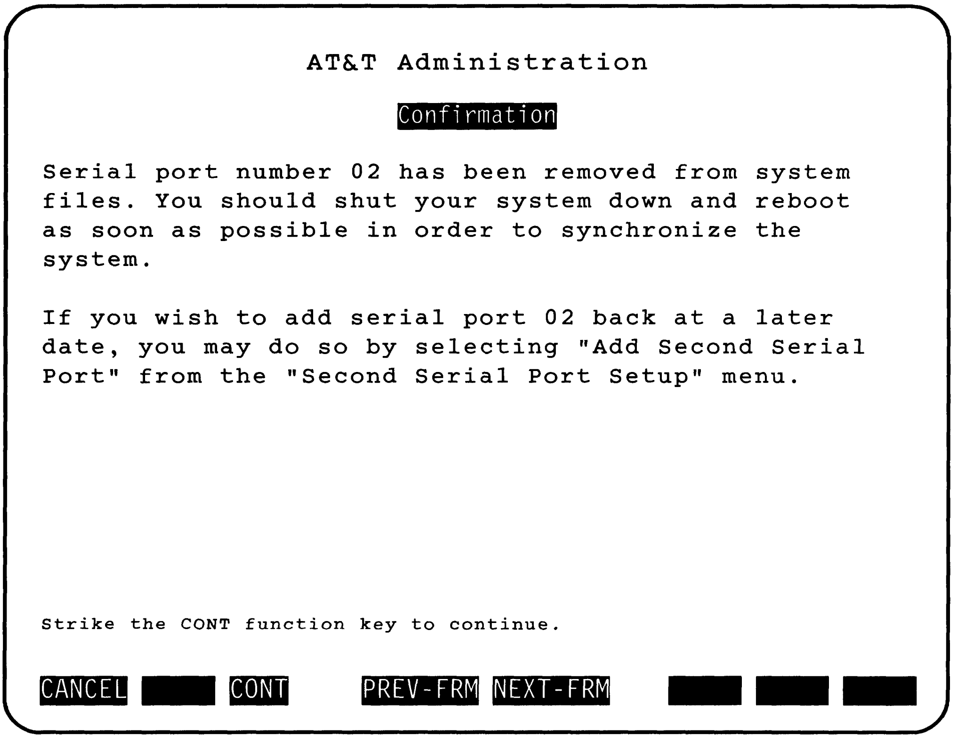


```
                    AT&T Administration

                          Warning

The second serial port is currently in use.
Please try again later.







Strike the CONT function key to continue.
```

6. Otherwise, if the port has been configured using Serial Port Setup or Printer Setup, it is unconfigured from the system files or the printer is disabled respectively and the following message appears:

   ```
   Rebuild UNIX system takes approximately 2 minutes.
   Please wait ...
   ```

   After the UNIX system is rebuilt, the folling confirmation message appears:



```
                AT&T Administration

                   Confirmation

Serial port number 02 has been removed from system
files. You should shut your system down and reboot
as soon as possible in order to synchronize the
system.

If you wish to add serial port 02 back at a later
date, you may do so by selecting "Add Second Serial
Port" from the "Second Serial Port Setup" menu.





Strike the CONT function key to continue.
```

7. Strike CONT to continue and you will receive a message asking if you want to reboot the system. If you enter 'y' the automatic shutdown proceeds or if you enter 'n' the shutdown is canceled. If you do not shutdown, you will return to the Peripheral Setup menu.

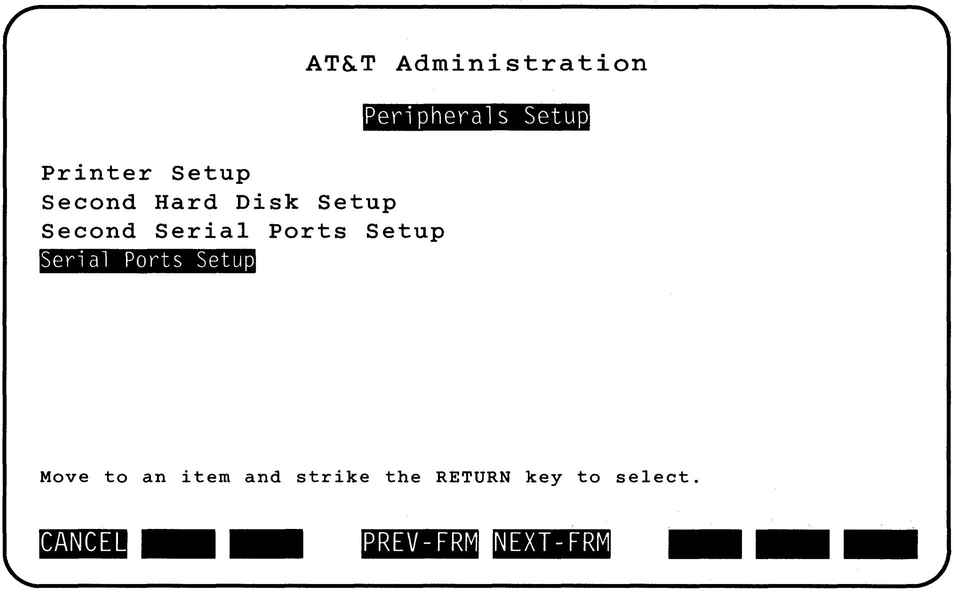
Serial Port Setup

The serial port on the back of your computer can be used to connect your computer to an external modem, a serial printer, a terminal, another computer, or other devices such as a Develcon switch. You also have the option of connecting a plotter or other type of serial device to the serial port if you've installed the software to support its use.

To use the serial port to connect to serial devices, you must (1) make the physical connection using a cable with RS-232 connectors, and (2) set up (configure) your system to recognize the type of connection you're making. When installing a modem, make sure the modem is connected to the serial port and power is turned on before proceeding.

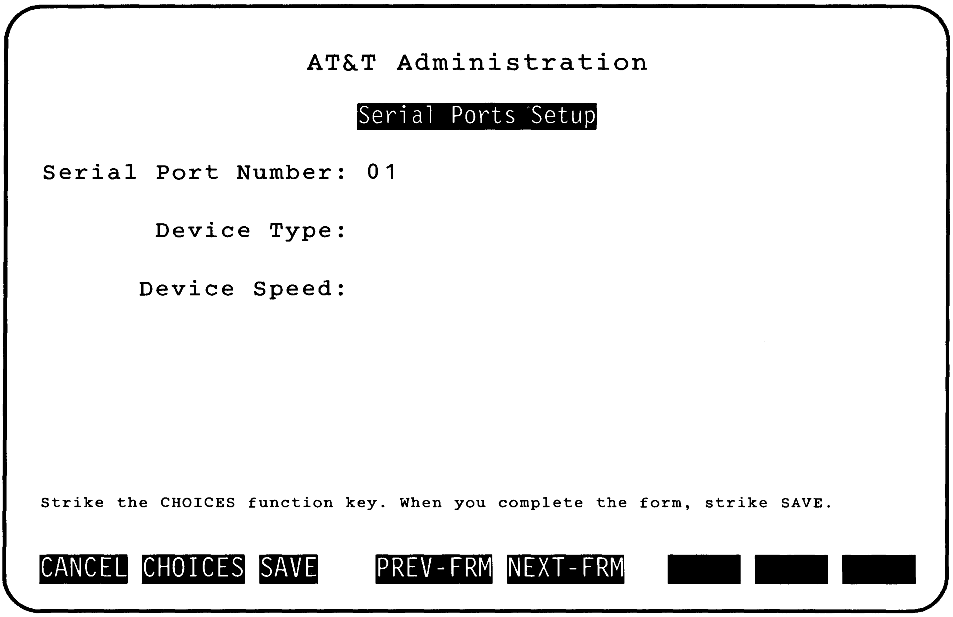
To set up the serial port, use the following procedure.

1. From the Administration menu, highlight Peripherals Setup and strike Enter. The Peripherals Setup menu appears as follows:



```
                    AT&T Administration

                     Peripherals Setup

Printer Setup
Second Hard Disk Setup
Second Serial Ports Setup
Serial Ports Setup







Move to an item and strike the RETURN key to select.

CANCEL                PREV-FRM NEXT-FRM
```

2. Highlight Serial Ports Setup from the Peripherals Setup menu and strike Enter. The Serial Ports Setup form appears as shown in the following sample:



```
                    AT&T Administration

                     Serial Ports Setup

Serial Port Number: 01

      Device Type:

     Device Speed:




Strike the CHOICES function key. When you complete the form, strike SAVE.
```

NOTE

The "Device Type:" field will be None or the printer name previously configured for the port. The "Device Speed:" field will be blank or the speed previously set up for the port.

3. With the cursor resting on the "Serial Port Number" field, enter the port number or strike CHOICES to obtain a list of the valid port numbers.

   If you striked CHOICES, the Port Number pop-up menu appears.

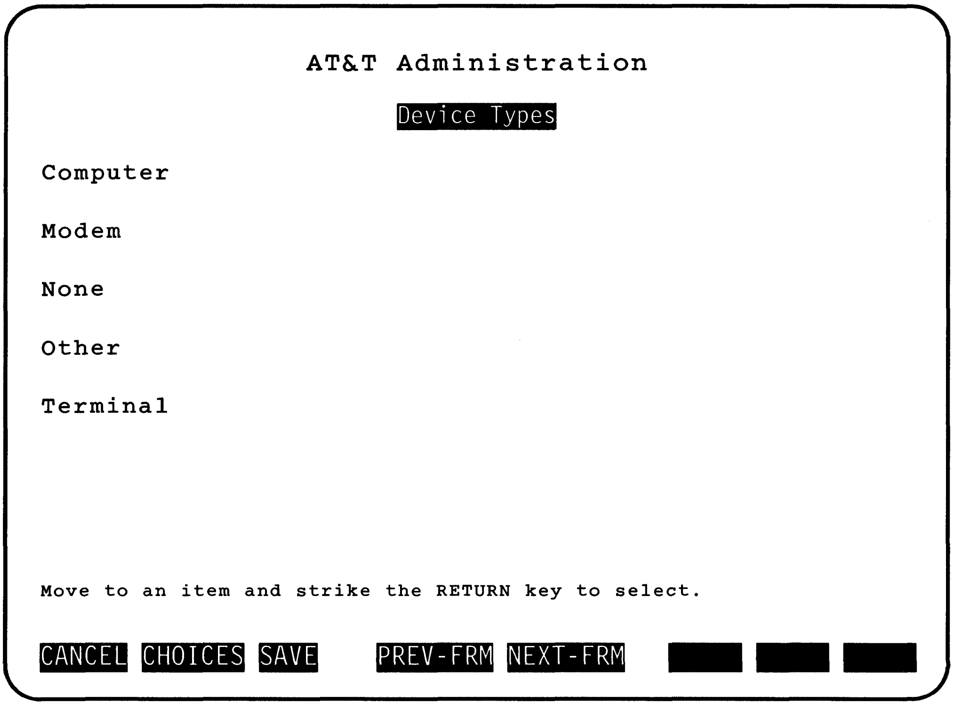
   If you striked CHOICES, highlight the port number you want and strike Enter. The port number appears in the "Port Number" field.

NOTE

If you have not installed a multi-ports board nor enabled the second internal serial port, port 01 should be assumed and the Port Number pop-up menu will not appear.

4. Move the cursor to the "Device Type:" field and enter a device type for serial port connection or strike CHOICES for a list of device types.

   If you striked CHOICES, the Device Types pop-up menu appears as follows:



```
                    AT&T Administration

                        Device Types

Computer

Modem

None

Other

Terminal




Move to an item and strike the RETURN key to select.

CANCEL CHOICES SAVE     PREV-FRM NEXT-FRM
```

A brief explanation of each device follows:

Computer
Configures the serial port for connection to a computer. You would connect directly to a computer for sending mail or remotely logging in on that computer.

Modem
Configures the serial port for connection to a modem so your computer can communicate with other computers.

None
Clears the current port configuration.

Other
Configures the serial port for a plotter or Develcon switch or whatever other supported serial device you specify. A Develcon switch is another way to contact another computer for sending mail or remotely logging in on that computer.

Terminal
Configures the serial port for connection to a terminal.

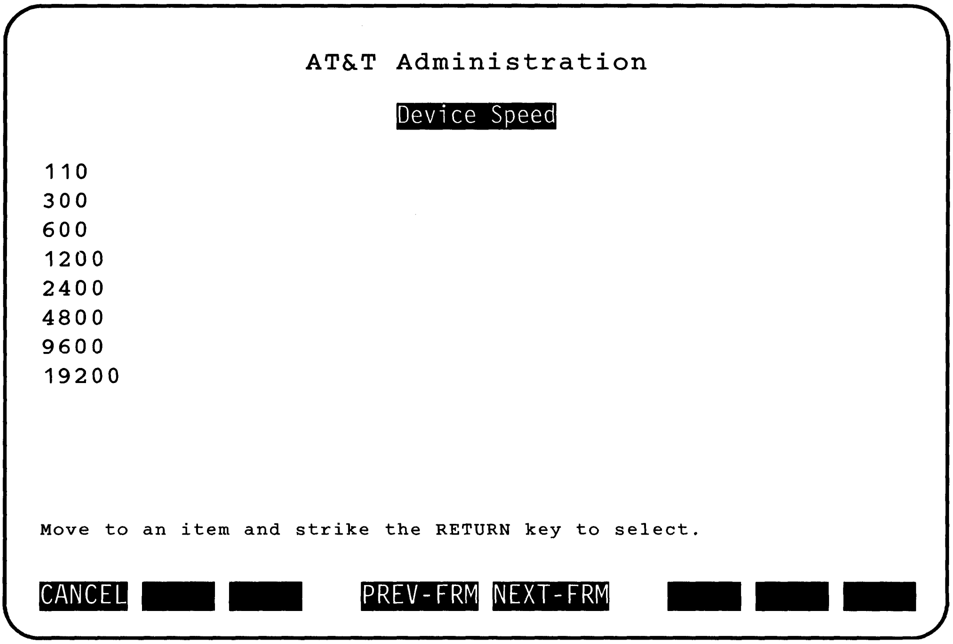
NOTE

Modem, Computer, and Other can all be used to communicate with other computers. The type of device that you select depends on the physical connection between your computer and the other computer.

If you striked CHOICES, highlight the device type you want and strike Enter. The device type appears in the "Device Type:" field.

5. Move the cursor to the "Device Speed:" Field and enter the device speed or strike CHOICES to obtain a list of the valid device speeds.

   If you striked CHOICES, the Device Speed pop-up menu appears as follows:



```
                    AT&T Administration

                        Device Speed

110
300
600
1200
2400
4800
9600
19200



Move to an item and strike the RETURN key to select.
```

If you striked CHOICES, highlight the device speed you want and strike Enter. The device speed appears in the "Device Speed:" field. Generally, you can set your device speed to 9600 baud for a connection to a terminal and 9600 baud for direct connection to a computer. See the appropriate documentation for the modem being used or the device being used for "other" for more information on device speed.

6. Strike SAVE.

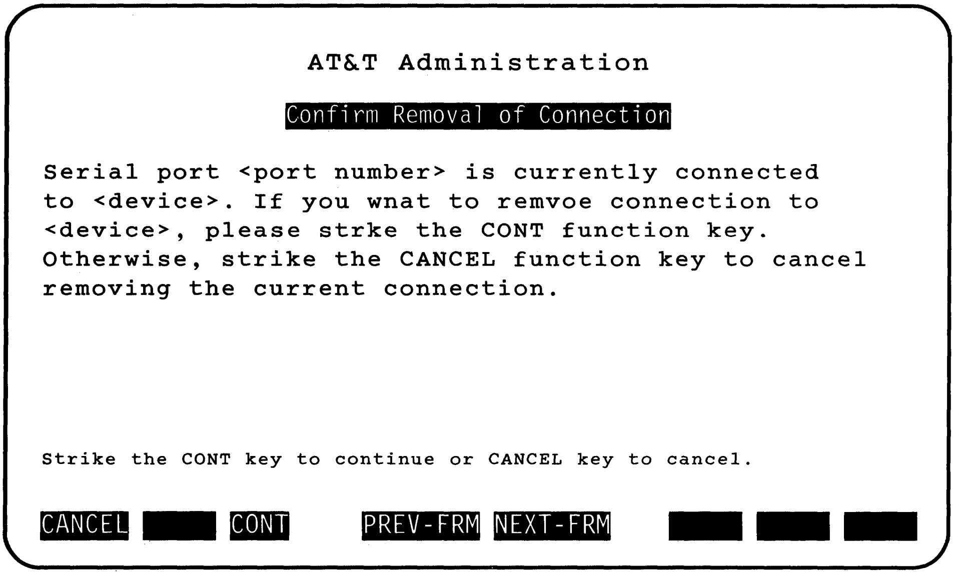
If you're connecting none, modem, terminal, computer or other, pick the correct procedure from the following pages and use it to finish setting up your serial port.

Connection to None

This procedure assumes you want to remove a previously configured serial port.

1. After you have selected "None" for the device type and struck SAVE from the Serial Ports Setup form, a warning or completion form appears.

   If you had previously assigned a device (e.g., printer, terminal, modem, computer, or other device) to the same port and the port is not currently active, the following warning message appears:

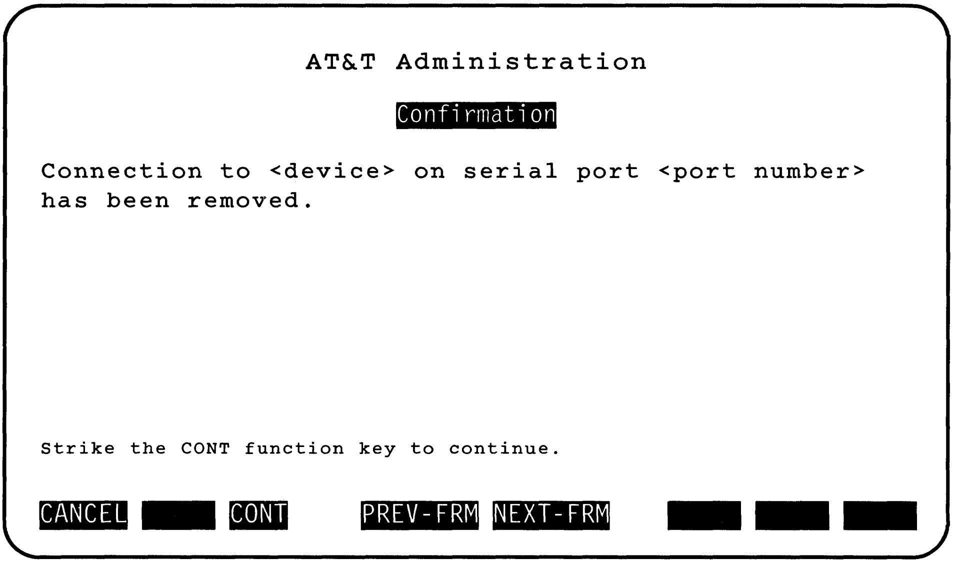


```
                    AT&T Administration

               Confirm Removal of Connection

Serial port <port number> is currently connected
to <device>. If you wnat to remvoe connection to
<device>, please strke the CONT function key.
Otherwise, strike the CANCEL function key to cancel
removing the current connection.




Strike the CONT key to continue or CANCEL key to cancel.
```

NOTE

CANCEL closes this frame, stops the "Connect to None" task, and makes the "Peripherals Setup" menu active.

Otherwise, the following completion message appears:



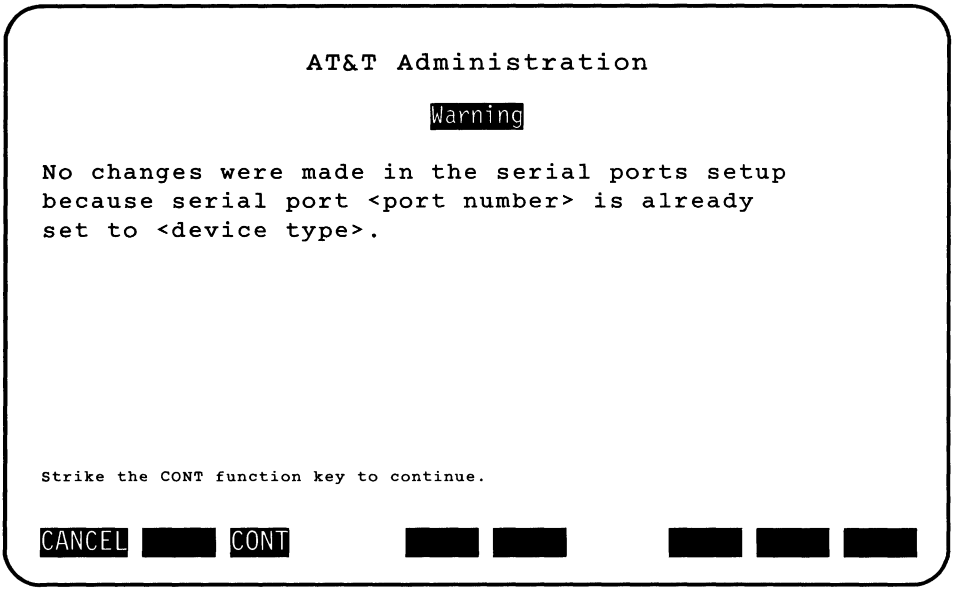
```
                    AT&T Administration

                        Confirmation

Connection to <device> on serial port <port number>
has been removed.





Strike the CONT function key to continue.
```

The <device> could be printer, terminal, modem, computer, or the name of the "other" device (e.g., Develcon).

2. Strike CONT to go back to the "Peripherals Setup" menu.

Connection to None Warning Messages

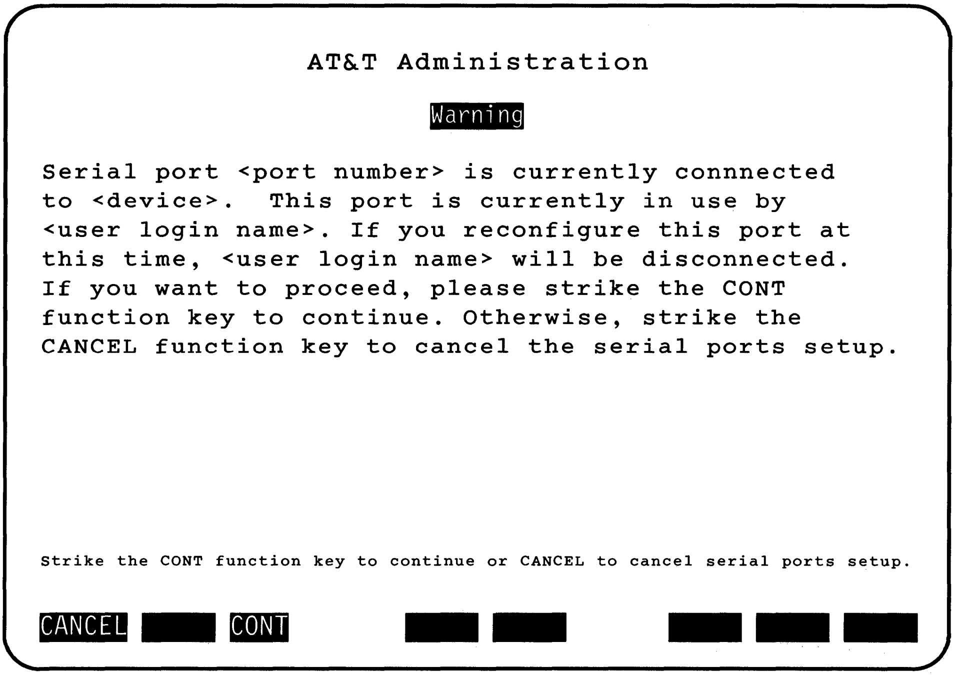
If the previous connection was "None" (i.e., the port is free), then the following message will appear instead of the confirmation and completion messages.



```
AT&T Administration

Warning

No changes were made in the serial ports setup
because serial port <port number> is already
set to <device type>.

Strike the CONT function key to continue.
```

Port Already Configured and Currently in Use

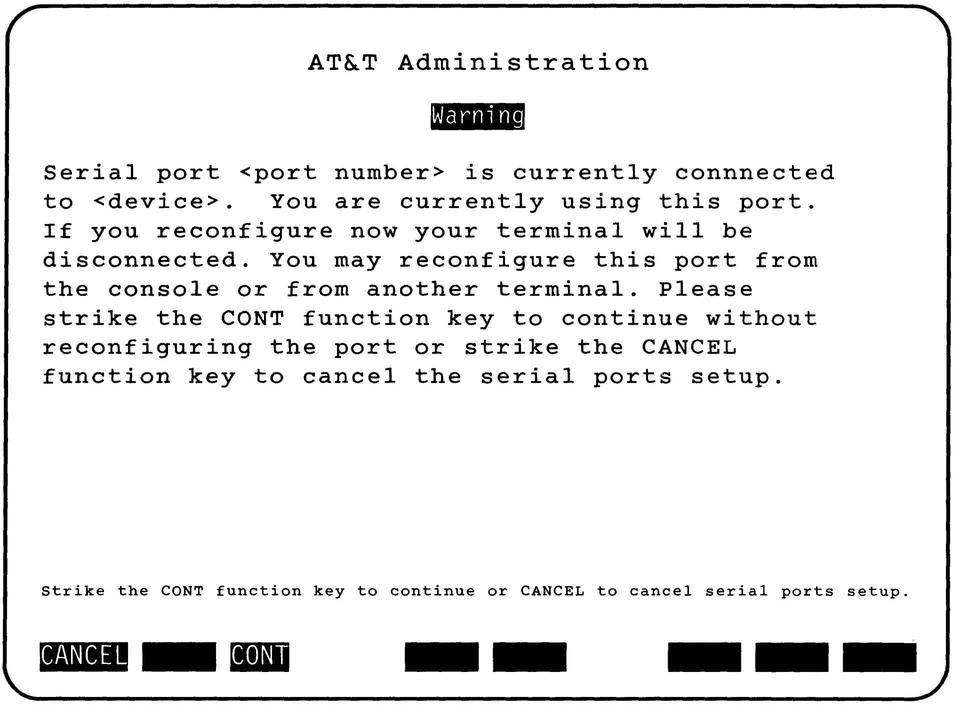
If the serial port being reconfigured to "None" is currently being used by a user different from the invoking user, the invoking user will be notified and asked to confirm if he or she wants to continue. The following message appears instead of the previous warning.



```
                     AT&T Administration

                           Warning

Serial port <port number> is currently connnected
to <device>.  This port is currently in use by
<user login name>. If you reconfigure this port at
this time, <user login name> will be disconnected.
If you want to proceed, please strike the CONT
function key to continue. Otherwise, strike the
CANCEL function key to cancel the serial ports setup.




Strike the CONT function key to continue or CANCEL to cancel serial ports setup.
```

Port Already Configured and Currently in Use by the Invoking User

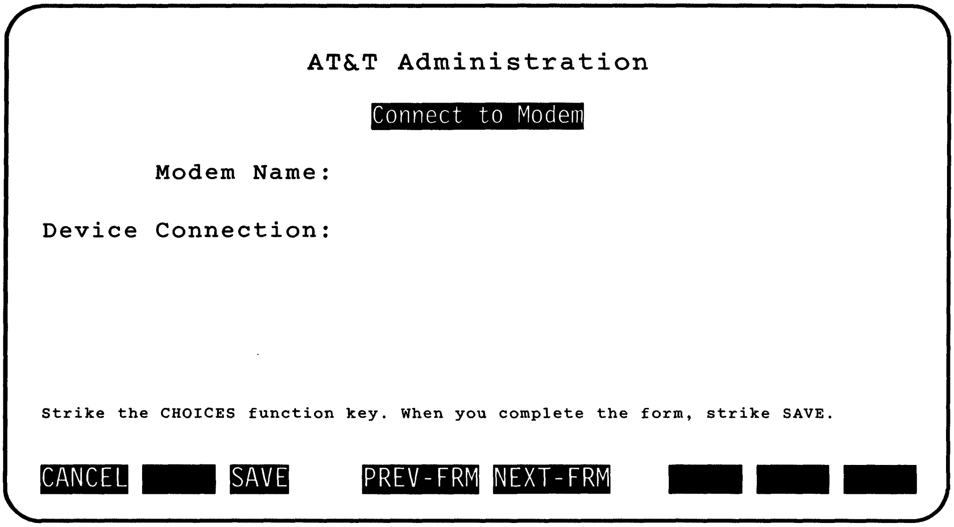
If the serial port being reconfigured to "None" is currently in use by the invoking user, the following message appears instead of the previous ones.



```
                    AT&T Administration

                          Warning

Serial port <port number> is currently connnected
to <device>.  You are currently using this port.
If you reconfigure now your terminal will be
disconnected. You may reconfigure this port from
the console or from another terminal. Please
strike the CONT function key to continue without
reconfiguring the port or strike the CANCEL
function key to cancel the serial ports setup.





Strike the CONT function key to continue or CANCEL to cancel serial ports setup.
```

Connection to Modem

If you're connecting a modem, a form requesting the name of the modem will appear as follows:

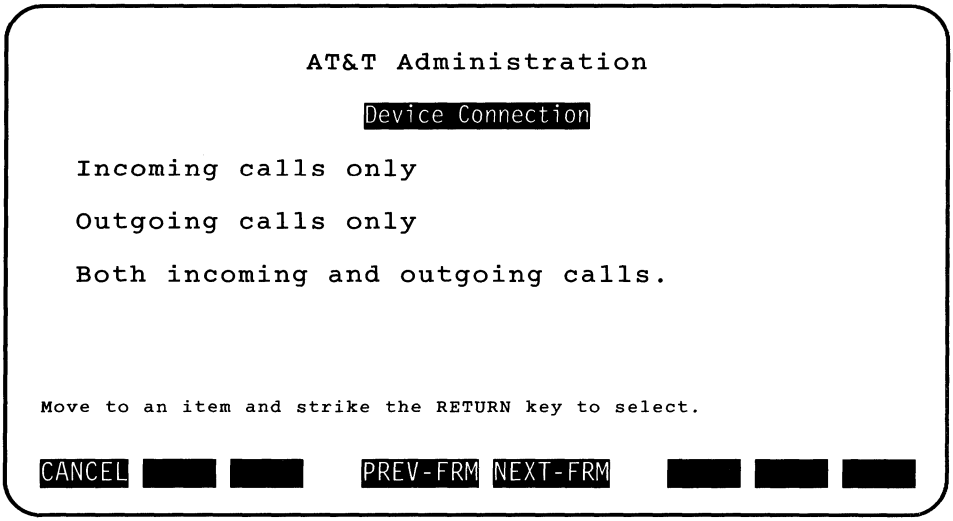


```
                 AT&T Administration

                  Connect to Modem

      Modem Name:

Device Connection:





Strike the CHOICES function key. When you complete the form, strike SAVE.
```

Refer to Appendix D for instructions on initial modem installation, reinitializing modems, and recommended modem switch settings.

1. With the cursor resting on the "Modem Name:" field, strike CHOICES. The Modems pop-up menu appears.
2. Highlight the appropriate modem name and strike Enter.

   The Device Connection frame appears as follows:



```
                    AT&T Administration

                     Device Connection

   Incoming calls only

   Outgoing calls only

   Both incoming and outgoing calls.



Move to an item and strike the RETURN key to select.
```

3. Highlight the correct device connection and strike Enter.

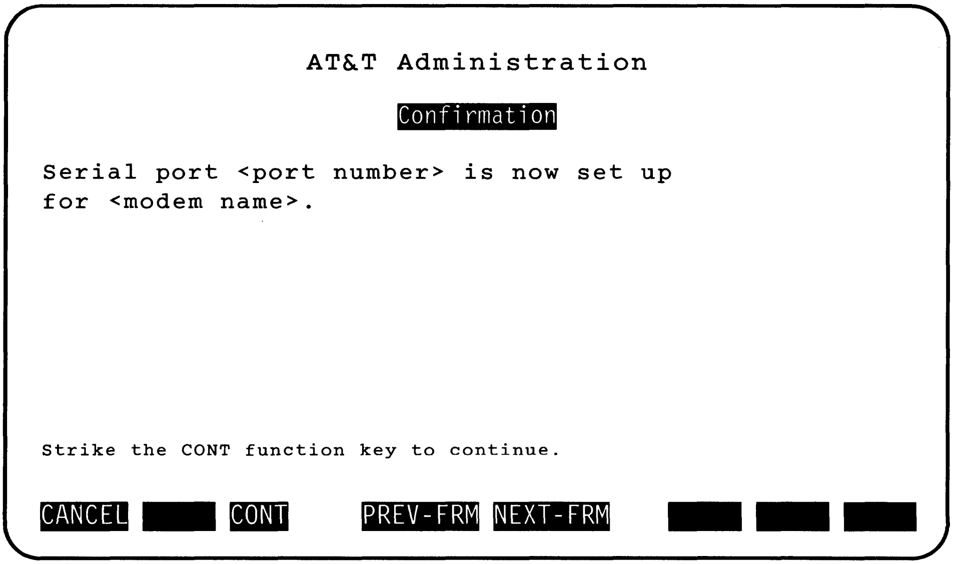
   Valid device connections include:

   - **Incoming calls only**: Incoming data calls can be received. Outgoing calls are not possible using this port. This means you can receive mail but you cannot login to another computer or send mail.
   - **Outgoing calls only**: Outgoing data calls can be sent. The UNIX system will not respond to incoming calls on this port. This means other systems cannot dial into your computer. You can send mail, but you cannot receive mail. This option is usually selected for security reasons.
   - **Both incoming and outgoing calls**: Outgoing and incoming data calls can be placed and received. The UNIX system will respond to incoming calls. You can send and receive mail, and dial up to other computers.

4. Strike SAVE.

   If you selected a serial port number from the "Serial Ports Setup" form that is currently in use by a device different from modem (i.e., one of the following : terminal, computer, printer, or other, like a plotter) and not "None", you will receive a warning message.

If there are no problems, a completion message appears telling you the serial port is now set up for a modem as follows:



```
                    AT&T Administration

                        Confirmation

Serial port <port number> is now set up
for <modem name>.








Strike the CONT function key to continue.
```

5. Strike CONT to close this frame and display the "Peripherals Setup" menu.

NOTE

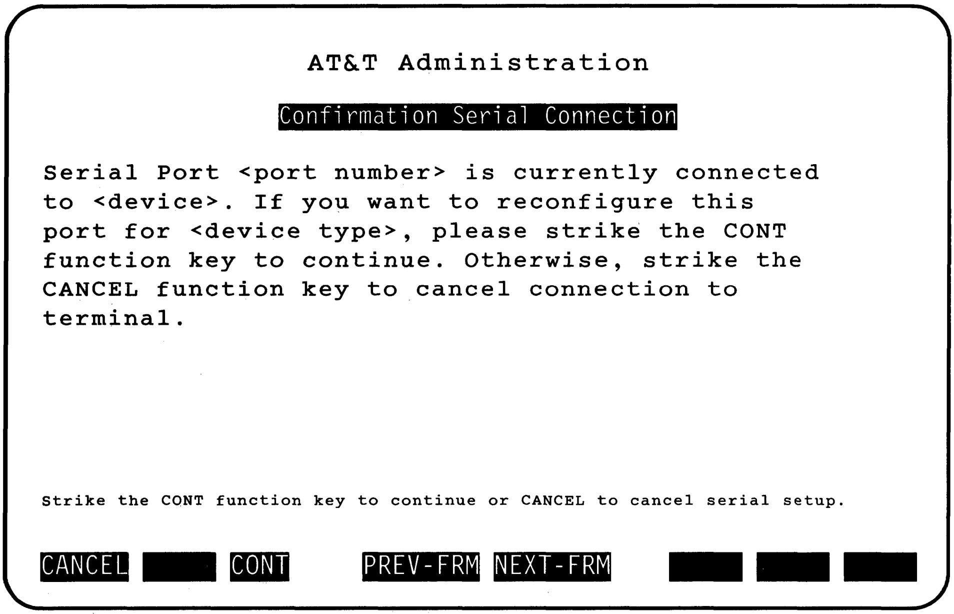
If you set your modem to operate at 2400 bps, then you will be able to exchange mail only with systems that also support a 2400 bps modem. You must change the mail information for each such system to set the communications data speed to "2400." Do this through the "Mail Setup" function under Administration. To exchange mail with systems whose communications data speed is "1200," make sure you set up your modem for 1200 bps.

Connection to Terminal

If you're connecting a terminal, a confirmation form or warning message appears.

1.

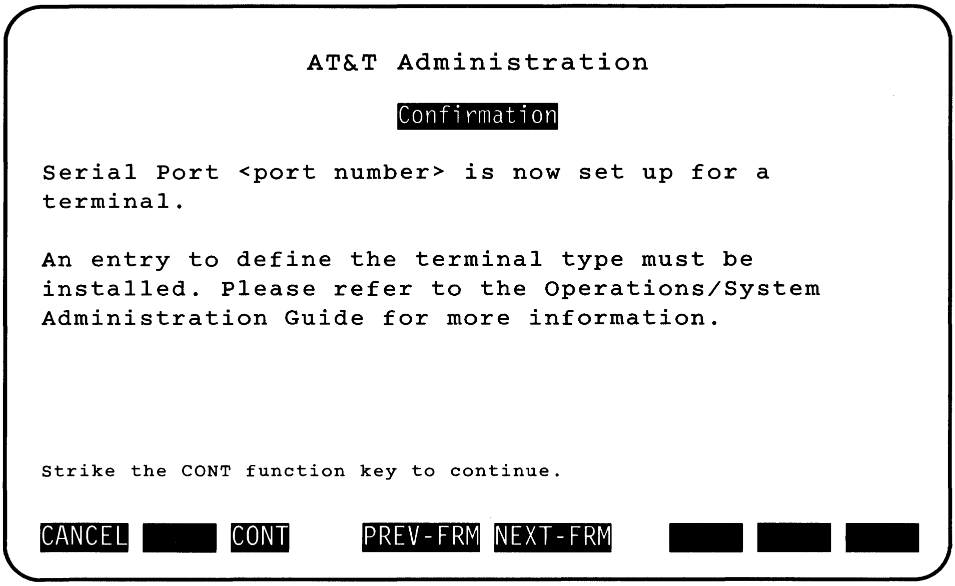
   If you selected a port that is currently in use by a device different from terminal (i.e., one of the following: modem, computer, printer, or other, like a plotter) and not "None", you will receive a the following message. This port is not currently in use/active.



```
                AT&T Administration

            Confirmation Serial Connection

Serial Port <port number> is currently connected
to <device>. If you want to reconfigure this
port for <device type>, please strike the CONT
function key to continue. Otherwise, strike the
CANCEL function key to cancel connection to
terminal.




Strike the CONT function key to continue or CANCEL to cancel serial setup.
```

   If you strike CONT, the port will be reconfigured for terminal. The frames associated with Serial Ports Setup will be closed and the Peripherals Setup menu is activated.

2. Otherwise (port is not currently in use by a device different from terminal), a completion message appears telling you the serial port is now set up for a terminal and that a terminfo entry for the terminal being added should be installed as follows:



```
                    AT&T Administration

                        Confirmation

Serial Port <port number> is now set up for a
terminal.

An entry to define the terminal type must be
installed. Please refer to the Operations/System
Administration Guide for more information.




Strike the CONT function key to continue.
```

For a list of compatible terminals, refer to Chapter 5, Customizing Your Computer, under "Installing Software Support for Additional Terminals."

3. Strike CONT to go back to the Peripherals Setup menu.

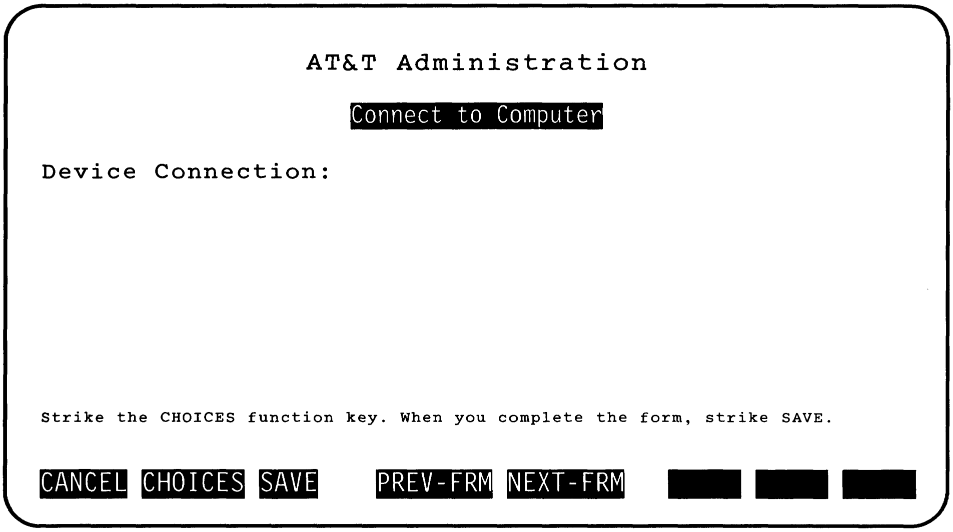
Connection to Computer

If you're connecting a computer, the Serial Setup form requesting the device type and device speed will appear.

NOTE This type of hard-wired link between your computer and another computer requires a null modem.

1. To change the default device speed, move the cursor to the "Device Speed:" field and strike CHOICES. Highlight the appropriate speed and strike Enter.

2. When the form is complete, strike SAVE. The Connect to Computer form appears as follows:



```
                    AT&T Administration

                    Connect to Computer

Device Connection:





Strike the CHOICES function key. When you complete the form, strike SAVE.
```

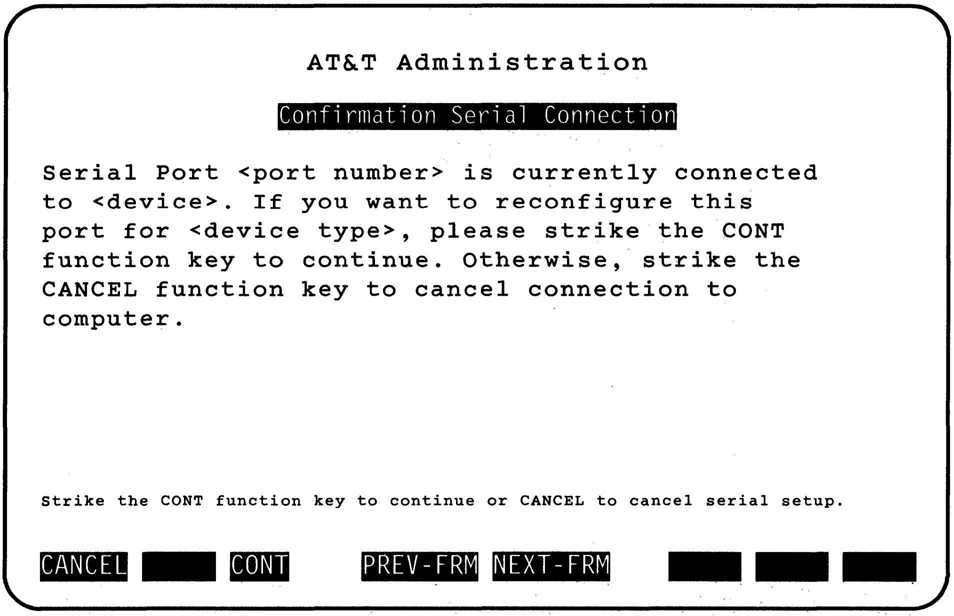
3. With the cursor resting on the "Device Connection:" field, strike CHOICES until the correct device connection appears. See "Connection To Modem" for an explantion of device connections.

   The Device Connection field toggles between the following:

   - Incoming calls only
   - Outgoing calls only
   - Both incoming and outgoing calls.

4. Select one and strike SAVE.

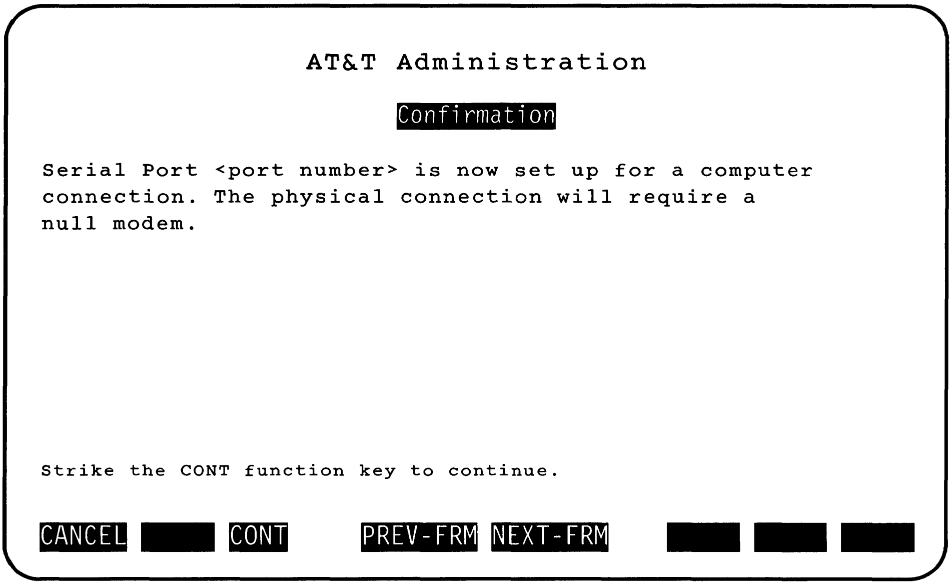
   If you selected a port that is currently in use by a device different from computer (i.e., one of the following: modem, terminal, printer, or other, like a plotter) and not "None", you will receive a the following message. This port is not currently in use/active.



```
                    AT&T Administration

               Confirmation Serial Connection

Serial Port <port number> is currently connected
to <device>. If you want to reconfigure this
port for <device type>, please strike the CONT
function key to continue. Otherwise, strike the
CANCEL function key to cancel connection to
computer.




Strike the CONT function key to continue or CANCEL to cancel serial setup.
```

If you strike CONT, the port will be reconfigured for computer. The frames associated with Serial Ports Setup will be closed and the Peripherals Setup menu is activated.

Otherwise (if you selected a port that is not currently in use by a device different from computer), a completion message appears telling you the serial port is now set up for a computer connection as follows:



```
                    AT&T Administration

                        Confirmation

Serial Port <port number> is now set up for a computer
connection. The physical connection will require a
null modem.







Strike the CONT function key to continue.
```

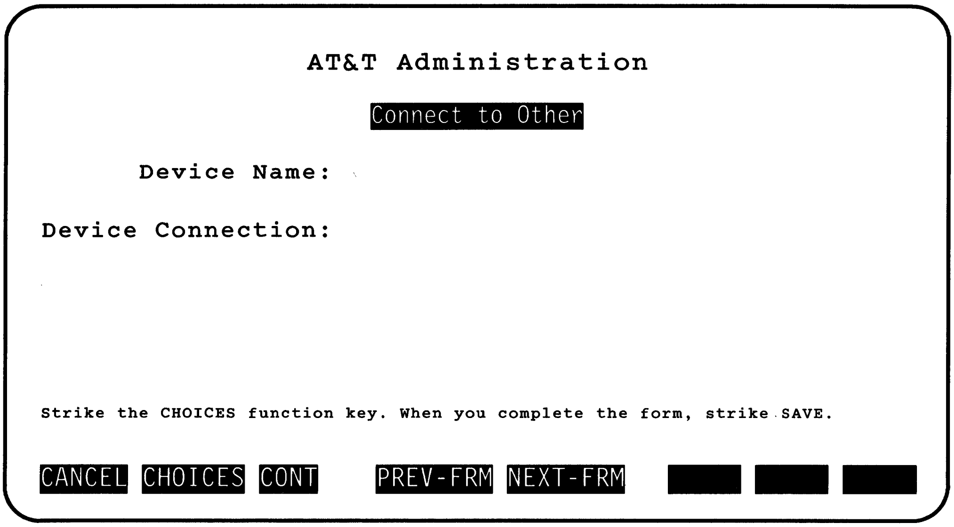
5. Strike CONT to go back to the Periperals Setup menu.

Connection to Computer Warning Messages

The warning messages in sections "Port Already Configured and Currently in Use" and "Port Already Configured and in Use by the Invoking User" apply here. These sections are located under section "Connection to None Warning Message".

Connection to Other

If you are connecting to a special device, the Connect to Other form requesting the device name (like a Develcon dataswitch) appears as follows:



AT&T Administration

Connect to Other

Device Name:

Device Connection:

Strike the CHOICES function key. When you complete the form, strike SAVE.

1. With the cursor resting on the "Device Name" field, enter the name of the special device, such as plotter, Develcon, etc. This name will not be validated.

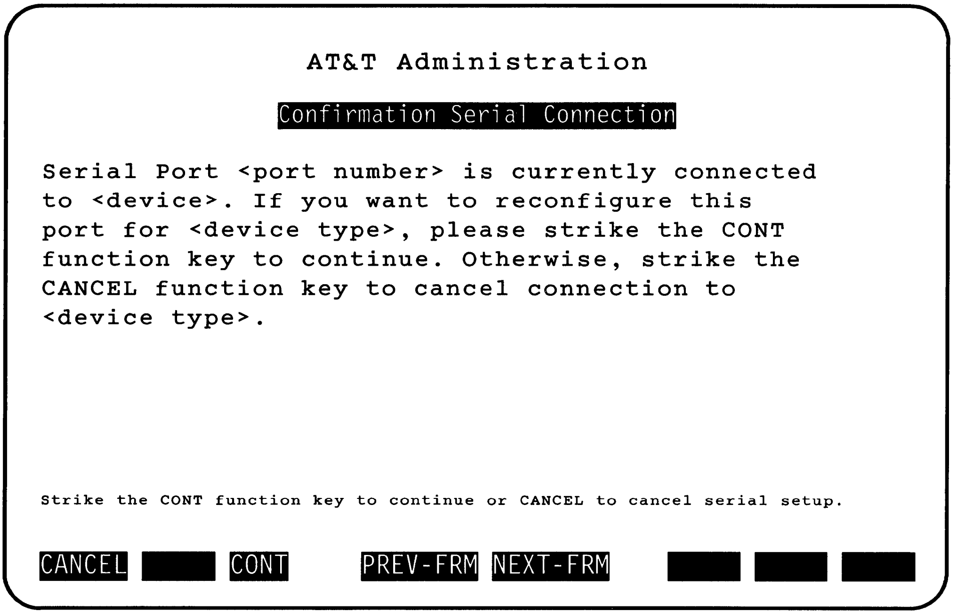
2. With the cursor resting on the "Device Connection:" field, strike CHOICES until the appropriate device connection appears. See "Conneciton to Modem" for an explanation of device connections.

   The Device Connection toggles between the following:

   - Incoming calls only
   - Outgoing calls only
   - Both incoming and outgoing calls.

3. Select one and strike SAVE.

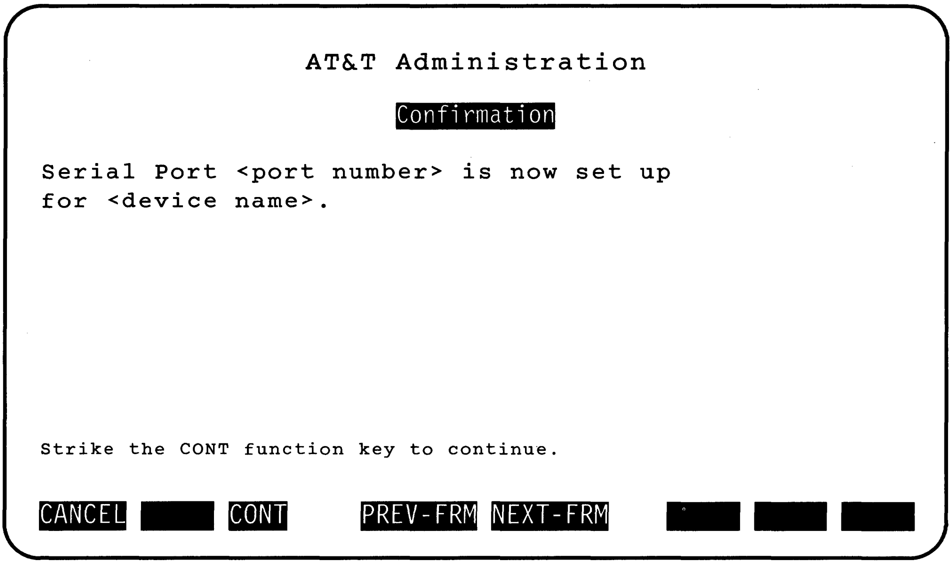
   If you selected a port that is currently in use by a device different from other (i.e., one of the following: modem, terminal, printer, or computer, like a plotter) and not "None", you will receive the following message. This port is not currently in use/active.



```
                    AT&T Administration

               Confirmation Serial Connection

Serial Port <port number> is currently connected
to <device>. If you want to reconfigure this
port for <device type>, please strike the CONT
function key to continue. Otherwise, strike the
CANCEL function key to cancel connection to
<device type>.




Strike the CONT function key to continue or CANCEL to cancel serial setup.
```

If you strike CONT, the port will be reconfigured for <device type>. The frames associated with Serial Ports Setup will be closed and the Peripherals Setup menu is activated.

Otherwise (if you selected a port that is not currently in use by a device different from other), a completion message appears telling you the serial port is now set up for a "other" connection as follows:



```
                    AT&T Administration

                        Confirmation

Serial Port <port number> is now set up
for <device name>.





Strike the CONT function key to continue.
```

4. Strike CONT to go back to the Periperals Setup menu.

Connection to Other Warning Messages

The warning messages in sections "Port Already Configured and Currently in Use" and "Port Already Configured and in Use by the Invoking User" apply here. These sections are located under section "Connection to

None Warning Message".

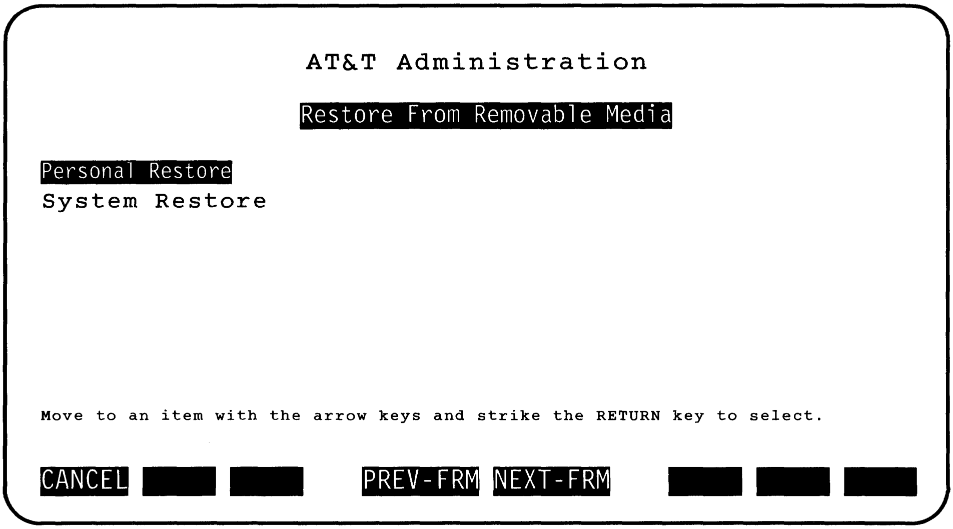
Second Hard Disk Setup

When this menu is selected the screen clears and the **diskadd** procedure takes over. This is an interactive program that prompts you for information about the setup of a second disk. For more information about the disk partitioning utility, see **diskadd**(1M) in the *User's/System Administrator's Reference Manual.*

Restore from Removable Media

If you've lost files either from operator error or from a system failure, you can restore your files using the latest floppy disk or cartridge tape backup. Remember, you can only restore files that you've backed up. Any changes made since your last backup will be lost. Also, any file that is in use at the time of restore, cannot be restored.

When you select Restore from Removable Media from the Administration menu, a menu appears as follows:



AT&T Administration

Restore From Removable Media

Move to an item with the arrow keys and strike the RETURN key to select.

There are two main types of restore from removable media operations as follows:

- Personal Restore is used to restore all files (directories or regular files) or selectively restore particular files backed up on removable media from your HOME directory to the hard disk.
- System Restore is used to restore all files from the removable media to hard disk. The system restore function provides you with the ability to do a restore (from system backup), do a selective system restore, or restore other user files (not just your own). The selective system restore function can be used to restore specific files. The files on the removable

media are displayed on the screen, and you can select the ones to restore.

To restore your computer, you must select the type of restore you need as explained above and follow that procedure.

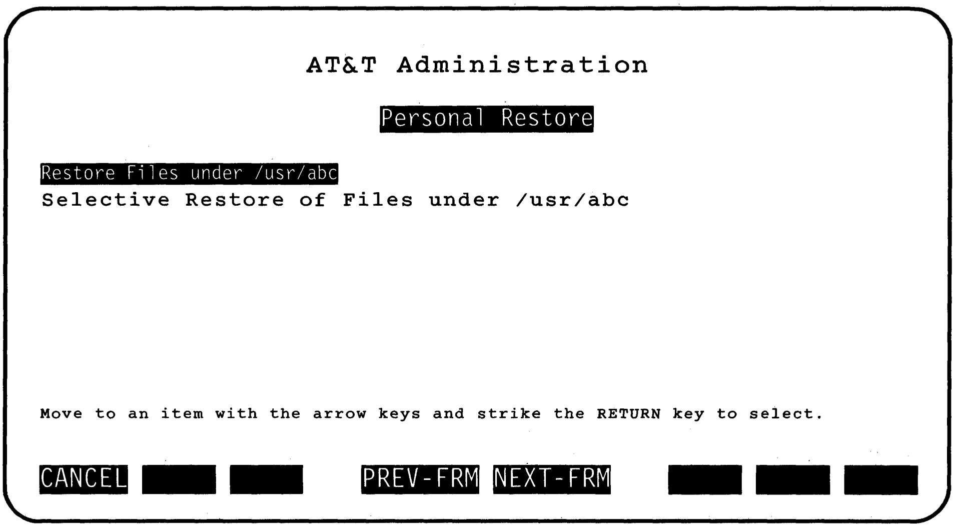
NOTE

Before attempting a restore, be sure the hard disk on your computer has enough space for all incoming information.

Personal Restore

To restore all files in your HOME directory, use the following procedure. The following procedure assumes your HOME directory is */usr/abc*.

1. Highlight `Personal Restore` from the Restore from Removable Media menu and strike Enter. The Personal Restore menu appears as follows:



AT&T Administration

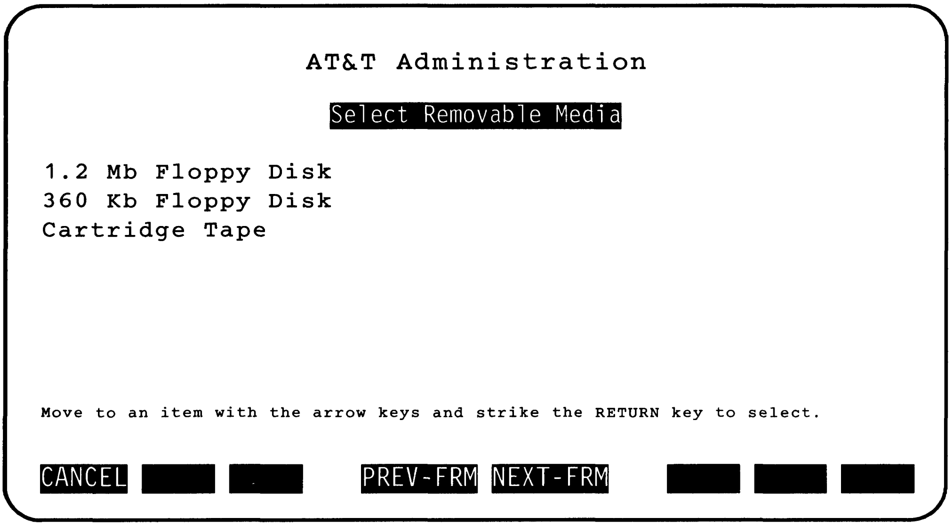
Personal Restore

Restore Files under /usr/abc
Selective Restore of Files under /usr/abc

Move to an item with the arrow keys and strike the RETURN key to select.

2. From the Personal Restore menu, highlight Restore Files under /usr/abc and strike Enter.

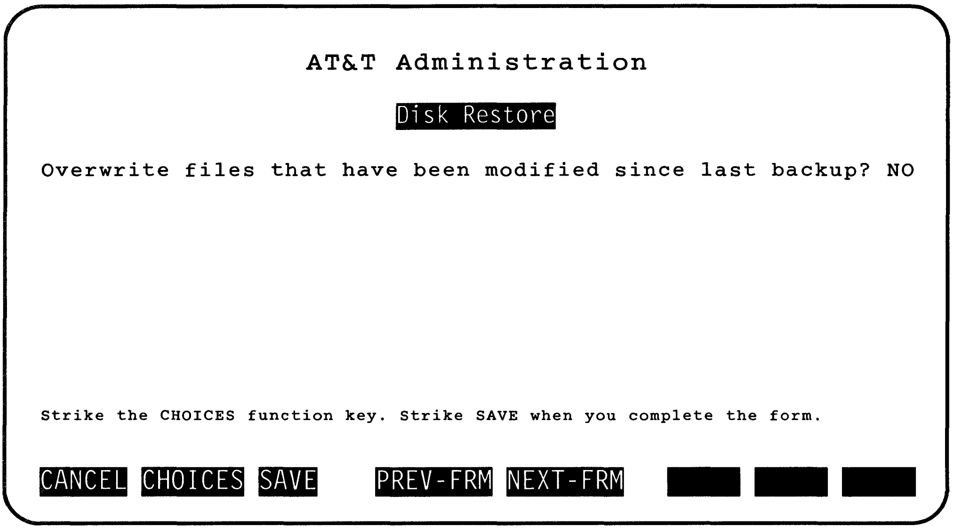
   The Select Removable Media menu appears as follows:



```
                    AT&T Administration

                   Select Removable Media

1.2 Mb Floppy Disk
360 Kb Floppy Disk
Cartridge Tape

Move to an item with the arrow keys and strike the RETURN key to select.
```

3. Depending on what options are available on your computer and what your needs are, select the appropriate media to restore your data from by highlighting one of the three choices (1.2 MB floppy disk, 360 KB floppy disk, or cartridge tape) and striking [Enter].

4. The Disk Restore form asks whether existing files on disk should be overwritten with restored files. The Disk Restore form appears as follows:



```
                    AT&T Administration

                        Disk Restore

Overwrite files that have been modified since last backup? NO






Strike the CHOICES function key. Strike SAVE when you complete the form.
```

5. Choose YES or NO to the question

   `Overwrite Files that have been modified since last backup?`

   by striking CHOICES and toggling between YES and NO. The default is NO.

   If you choose YES, all files on the floppy disk or tape will be transferred to your file system overwriting any files with the same name regardless of whether the file on the hard disk is newer than the one on the removable media. If you choose NO, files on the hard disk that are newer (i.e., have been modified after the backup was done) will not be overwritten.

6. Strike SAVE.

   A restore confirmation message appears telling you to insert the floppy disk or tape containing files you want to restore as follows:

   `Insert first floppy/tape to restore from.`

7. Insert the floppy disk or tape containing the files you want to restore and strike [Enter].

Once the restore is in progress, you will receive the following instruction:

```
Restore in progress. Do not remove the floppy/tape.
```

8. In addition, if the restore spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

   ```
   You may remove floppy number 1.
   To exit, please press 'q' followed by RETURN.

   To continue, insert floppy number 2
   and strike the RETURN key.
   ```

   If you press 'q' to exit, the following message is displayed:

   ```
   You have canceled the Restore from Removable Media.
   ```

9. Make sure you insert the floppy disk or tape in numerical order. Repeat inserting and removing until a message appears indicating the restore is complete as follows:

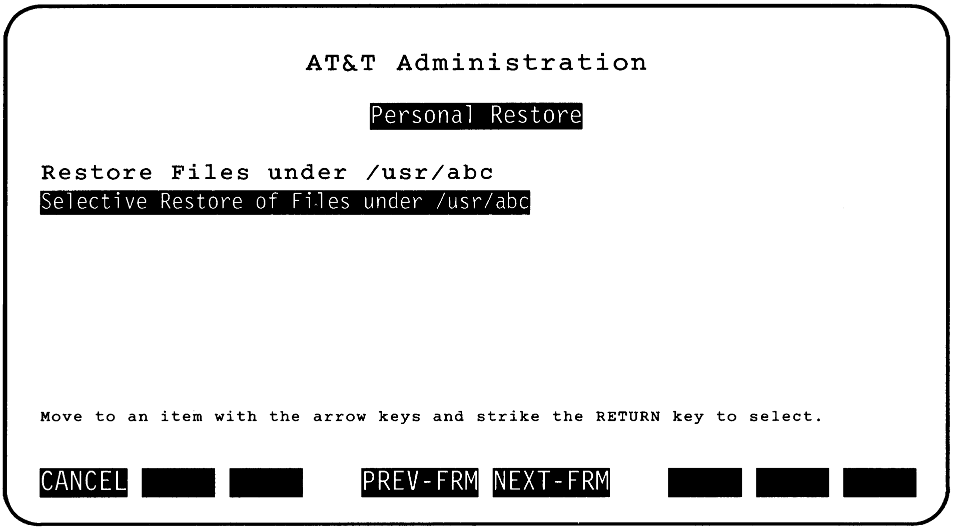
   ```
   Restore is done. You may remove the floppy.
   ```

10. Remove the last floppy disk when the system informs you that it has completed the restore.

Selective Personal Restore of Files

To selectively restore files in your HOME directory, use the following procedure. The following procedure assumes your HOME directory is */usr/abc*.

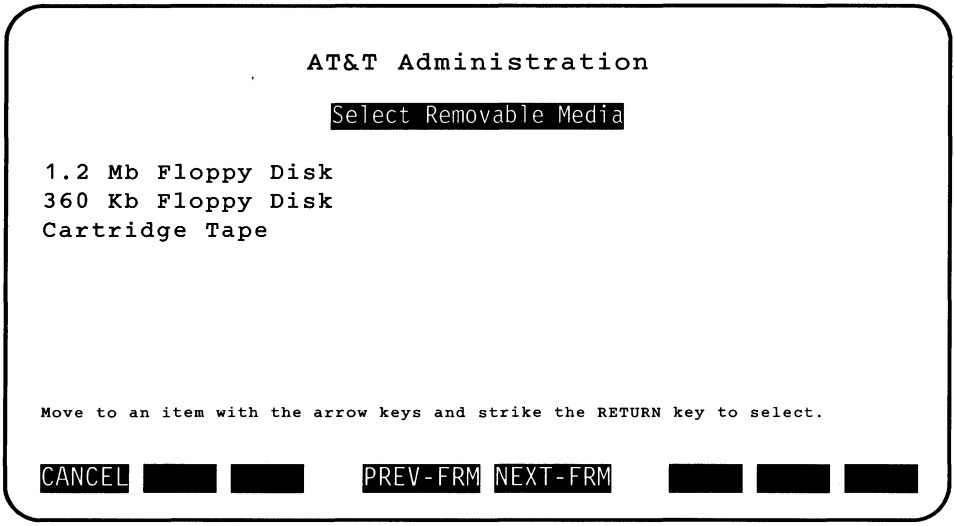
1. Highlight Personal Restore from the Restore from Removable Media menu and strike Enter. The Personal Restore menu appears as follows:



```
AT&T Administration

Personal Restore

Restore Files under /usr/abc
Selective Restore of Files under /usr/abc

Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL          PREV-FRM NEXT-FRM
```

2. From the Personal Restore menu, highlight Selective Restore of Files under /usr/abc and strike Enter.

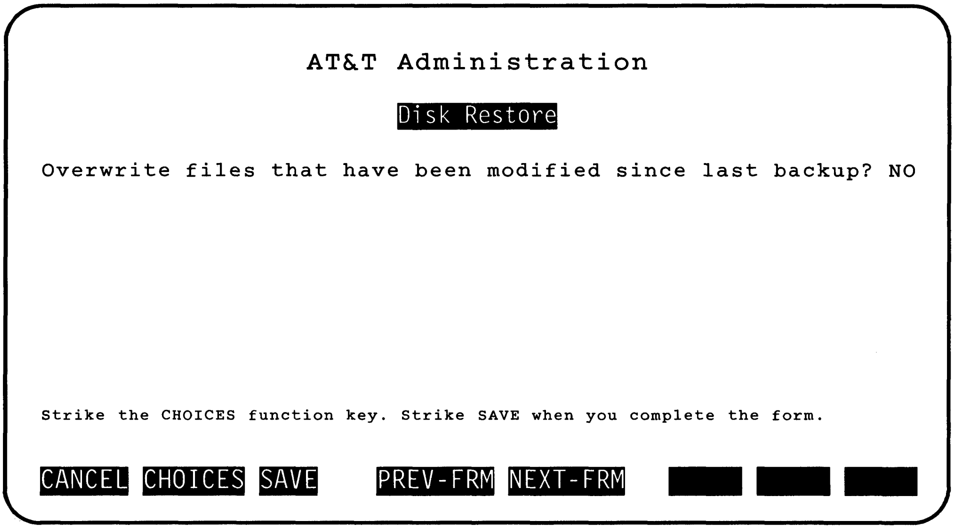
   The Select Removable Media menu appears as follows:



```
                    AT&T Administration

                   Select Removable Media

1.2 Mb Floppy Disk
360 Kb Floppy Disk
Cartridge Tape




Move to an item with the arrow keys and strike the RETURN key to select.
```

3. Depending on what options are available on your computer and what your needs are, select the appropriate media to restore your data from by highlighting one of the three choices (1.2 MB floppy disk, 360 KB floppy disk, or cartridge tape) and striking Enter.

4. The following Disk Restore form asks whether existing files on disk should be overwritten with restored files.



```
                    AT&T Administration

                       Disk Restore

Overwrite files that have been modified since last backup? NO








Strike the CHOICES function key. Strike SAVE when you complete the form.
```

5. Choose YES or NO to the question

   `Overwrite Files that have been modified since last backup?`

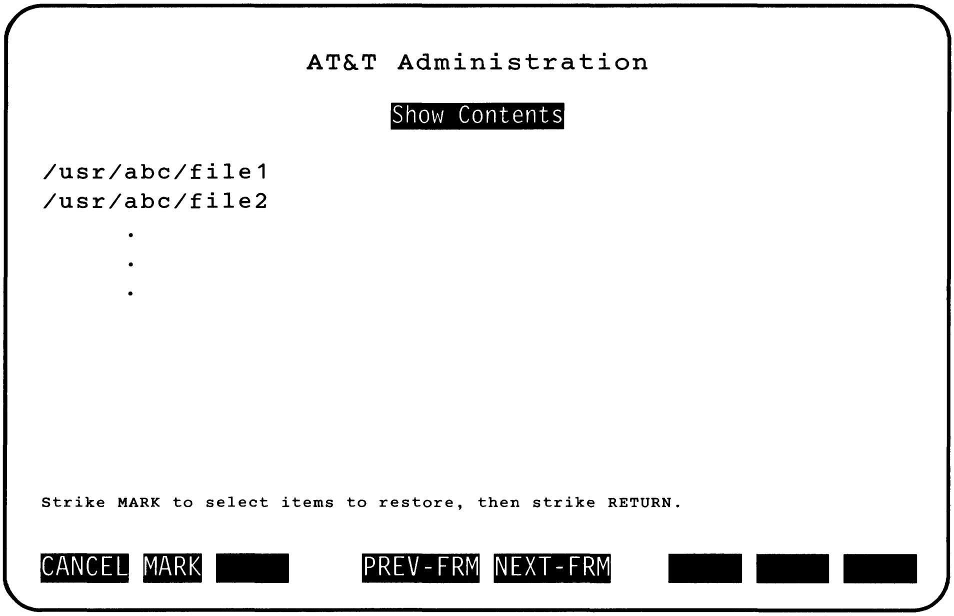
   by striking CHOICES and toggling between YES and NO. The default is NO.

   If you choose YES, selected files on the floppy disk or tape will be transferred to your file system overwriting any files with the same name regardless of whether the selected file on the hard disk is newer than the one on the removable media. If you choose NO, selected files on the hard disk that are newer (i.e., have been modified after the backup was done) will not be overwritten.

6. Strike SAVE.

NOTE	This procedure reads the media set twice: once to read the list of files that you can choose and then again to execute the restoring of the files that you chose.

7. You are prompted to insert the floppy or tape. Then the Show Contents menu appears as follows:



```
                    AT&T Administration

                       Show Contents

/usr/abc/file1
/usr/abc/file2
      .
      .
      .

Strike MARK to select items to restore, then strike RETURN.
```

NOTE: If you do not wish to restore any files at this time, strike CANCEL.

8. Select the files you want to restore with MARK until you have marked all the files you want to restore.
9. Strike [Enter] to restore the marked files.

   A restore confirmation message appears telling you to insert the floppy disk or tape containing files you want to restore as follows:

   ```
   Insert first floppy/tape to restore from.
   ```

10. Insert the floppy disk or tape containing the files you want to restore and strike Enter.

    Once the restore is in progress, you will receive the following instruction:

    ```
    Restore in progress. Do not remove the floppy/tape.
    ```

11. In addition, if the restore spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

    ```
    You may remove floppy number 1.
    To exit, please press 'q' followed by RETURN.

    To continue, insert floppy number 2
    and strike the RETURN key.
    ```

    If you press 'q' to exit, the following message is displayed:

    ```
    You have canceled the Restore from Removable Media.
    ```

12. Make sure you insert the floppy disk or tape in numerical order. Repeat inserting and removing until a message appears indicating the restore is complete as follows:

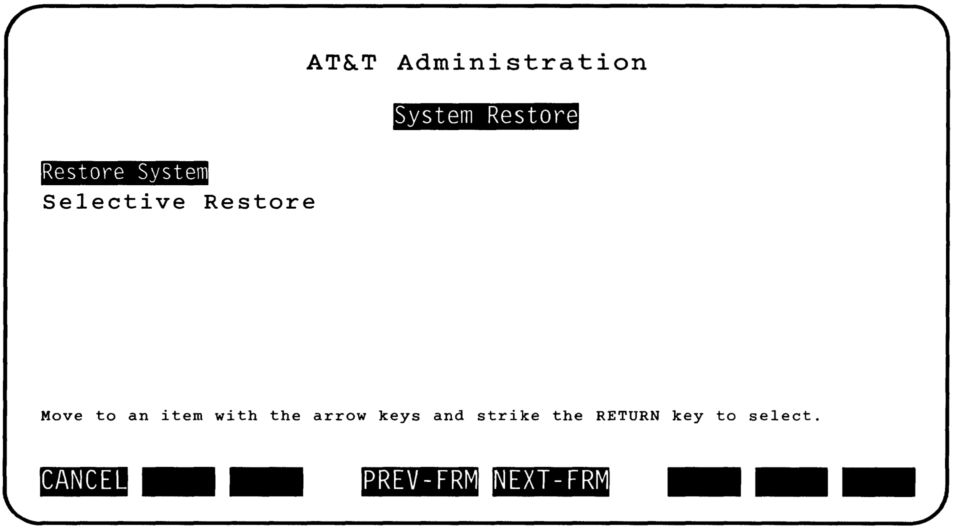
    ```
    Restore is done. You may remove the floppy.
    ```

13. Remove the last floppy or tape media when the system informs you that it has completed the restore.

System Restore

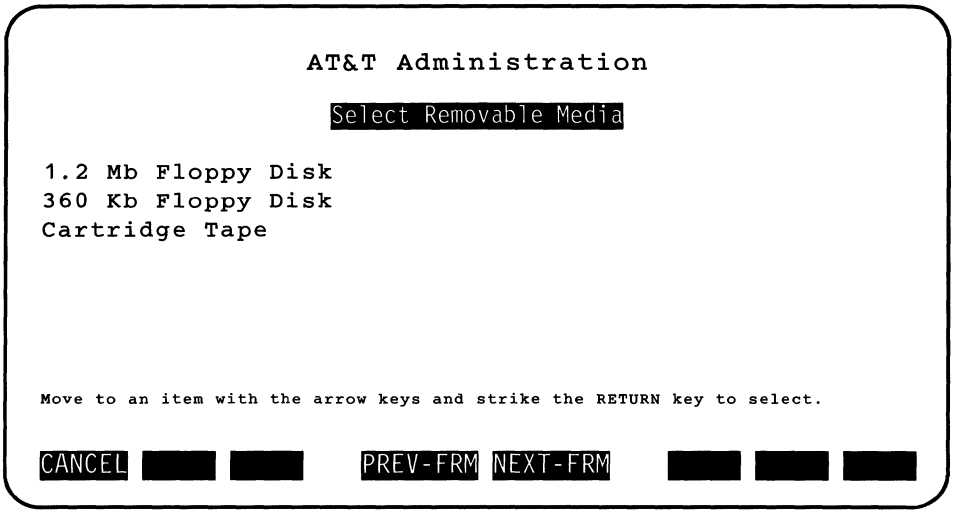
To restore system and user files from a system or incremental backup, use the following procedure. You must have system administration privileges to use this function.

1. Highlight System Restore from the Restore from Removable Media menu and strike Enter. The System Restore menu appears as follows:



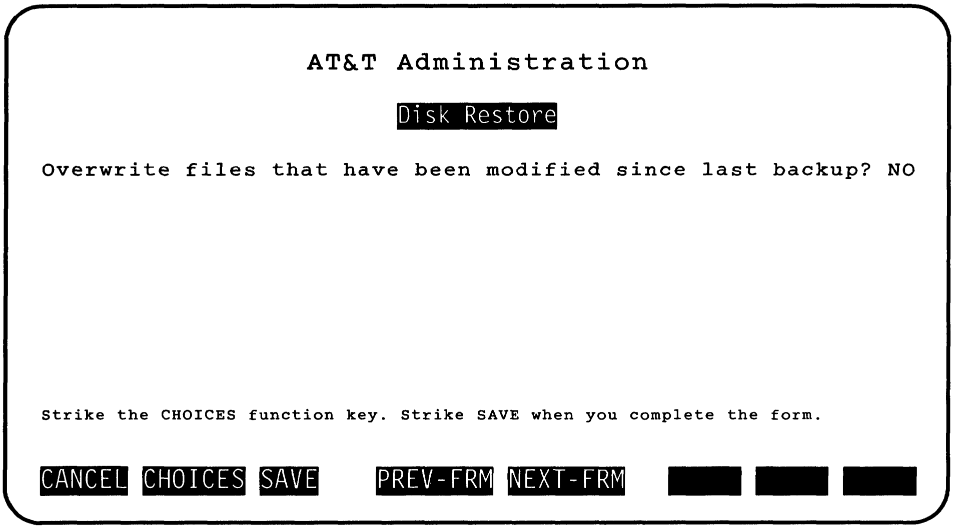
```
                    AT&T Administration

                      System Restore

Restore System
Selective Restore




Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL                PREV-FRM NEXT-FRM
```

2. From the System Restore menu, highlight Restore System and strike Enter. The Select Removable Media menu appears as follows:



```
                     AT&T Administration

                    Select Removable Media

1.2 Mb Floppy Disk
360 Kb Floppy Disk
Cartridge Tape




Move to an item with the arrow keys and strike the RETURN key to select.
```

3. Depending on what options are available on your computer and what your needs are, select the appropriate media to restore your data from by highlighting one of the three choices (1.2 MB floppy disk, 360 KB floppy disk, or cartridge tape) and striking Enter.
4. The following Disk Restore asks whether existing files on disk should be overwritten with restored files.



```
AT&T Administration

Disk Restore

Overwrite files that have been modified since last backup? NO




Strike the CHOICES function key. Strike SAVE when you complete the form.
```

Choose YES or NO to the question

```
Overwrite Files that have been modified since last backup?
```

by striking CHOICES and toggling between YES and NO. The default is NO.

If you choose YES, all files on the floppy disk or tape will be transferred to your file system overwriting any files with the same name regardless of whether the file on the hard disk is newer than the one on the removable media. If you choose NO, files on the hard disk that are newer (i.e., have been modified after the backup was done) will not be overwritten. All backed up files will be restored when SAVE is striked. By default, a file that is newer (most recent date) on the hard disk than on the removable medium will not be overwritten unless you specify that they should.

5. Strike SAVE.

A restore confirmation message appears telling you to insert the floppy disk or tape containing files you want to restore as follows:

```
Insert first floppy/tape to restore from.
```

6. Insert the floppy disk or tape containing the files you want to restore and strike [Enter].

   Once the restore is in progress, you will receive the following instruction:

   ```
   Restore in progress. Do not remove the floppy/tape.
   ```

7. In addition, if the restore spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

   ```
   You may remove floppy number 1.
   To exit, please press 'q' followed by RETURN.

   To continue, insert floppy number 2
   and strike the RETURN key.
   ```

   If you press ′q′ to exit, the following message is displayed:

   ```
   You have canceled the Restore from Removable Media.
   ```

8. Make sure you insert the floppy disk or tape in numerical order. Repeat inserting and removing until a message appears indicating the restore is complete as follows:

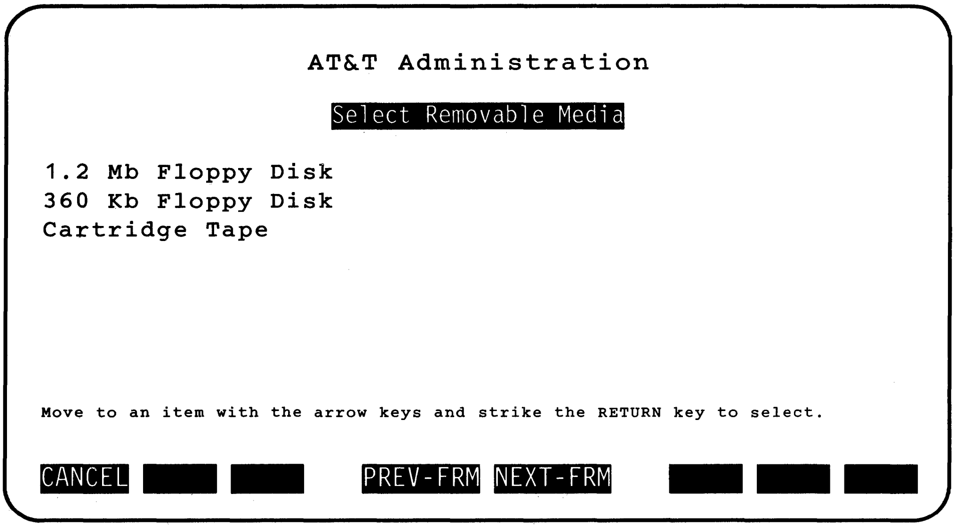
   ```
   Restore is done. You may remove the floppy.
   ```

9. Remove the last floppy disk when the system informs you that it has completed the restore.

Selective Restore

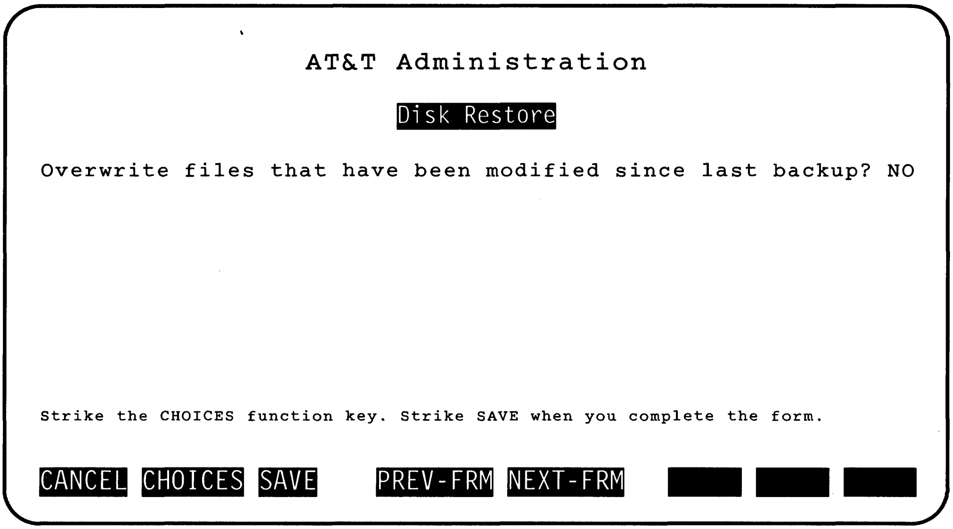
To selectively restore system or user files from the system or incremental backup of floppy or tape, use the following procedure. You must have system administration privileges to use this function.

1. From the System Restore menu, highlight Selective Restore and strike (Enter). The Select Removable Media menu appears as follows:



```
                    AT&T Administration

                  Select Removable Media

1.2 Mb Floppy Disk
360 Kb Floppy Disk
Cartridge Tape




Move to an item with the arrow keys and strike the RETURN key to select.

CANCEL                PREV-FRM NEXT-FRM
```

2. Depending on what options are available on your computer and what your needs are, select the appropriate media to restore your data from by highlighting one of the three choices (1.2 MB floppy disk, 360 KB floppy disk, or cartridge tape) and striking (Enter).
3. The following Disk Restore form asks whether existing files on disk should be overwritten with restored files.



```
                    AT&T Administration

                       Disk Restore

Overwrite files that have been modified since last backup? NO







Strike the CHOICES function key. Strike SAVE when you complete the form.
```

Choose YES or NO to the question

`Overwrite Files that have been modified since last backup?`

by striking CHOICES and toggling between YES and NO. The default is NO.

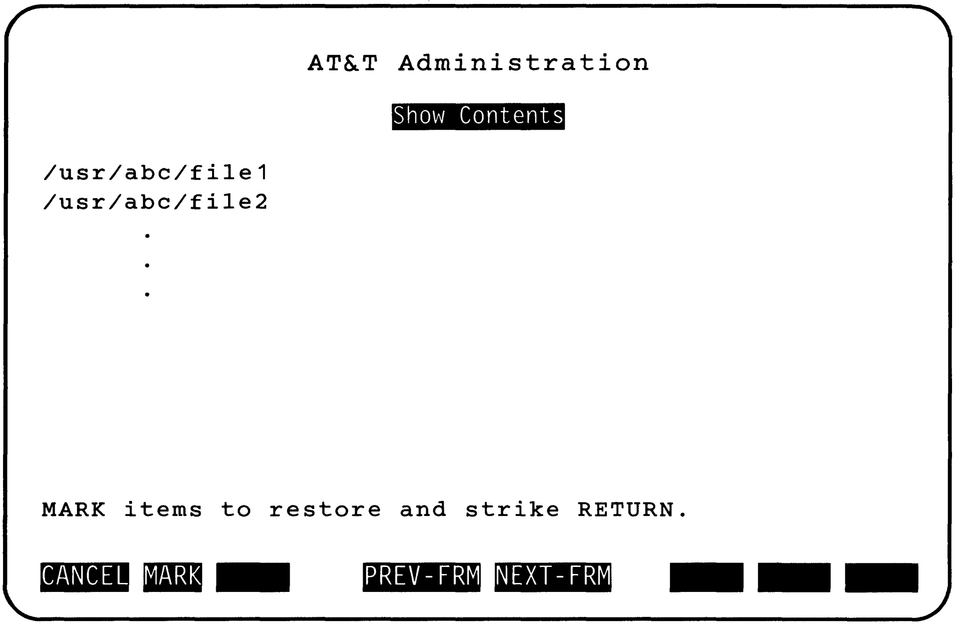
If you choose YES, selected files on the floppy disk or tape will be transferred to your file system overwriting any files with the same name regardless of whether the file on the hard disk is newer than the one on the removable media. If you choose NO, files on the hard disk that are newer (i.e., have been modified after the backup was done) will not be overwritten.

4. Strike SAVE.

NOTE

This procedure reads the media set twice: once to read the list of files that you can choose and then again to execute the restoring of the files that you chose.

5. You are prompted to insert the floppy or tape. Then the Show Contents menu appears as follows:



```
                    AT&T Administration

                        Show Contents

/usr/abc/file1
/usr/abc/file2
       .
       .
       .

MARK items to restore and strike RETURN.
```

NOTE: If you do not wish to restore any files at this time, strike CANCEL.

6. Select the files you want to restore and strike MARK until you have marked all the files you want to restore.
7. Strike Enter to restore the marked files.

   A restore confirmation message appears telling you to insert the floppy disk or tape containing files you want to restore as follows:

   ```
   Insert first floppy/tape to restore from.
   ```

8. Insert the floppy disk or tape containing the files you want to restore and strike Enter.

   Once the restore is in progress, you will receive the following instruction:

   ```
   Restore in progress. Do not remove the floppy/tape.
   ```

9. In addition, if the restore spans multiple floppies or tapes, you will be notified when to remove the current floppy or tape and insert the next one in sequence. When the contents of floppy 1 has been restored, for example, the following instructions would appear:

   ```
   You may remove floppy number 1.
   To exit, please press 'q' followed by RETURN.

   To continue, insert floppy number 2
   and strike the RETURN key.
   ```

   If you press 'q' to exit, the following message is displayed:

   ```
   You have canceled the Restore from Removable Media.
   ```

10. Make sure you insert the floppy disk or tape in numerical order. Repeat inserting and removing until a message appears indicating the restore is complete as follows:

    ```
    Restore is done. You may remove the floppy.
    ```

11. Remove the last floppy disk when the system informs you that it has completed the restore.

Shutdown

The "Shutdown" function allows you to bring down the UNIX system before turning off the power or rebooting the system.

You must have system administration permissions and be at the console to use shutdown.

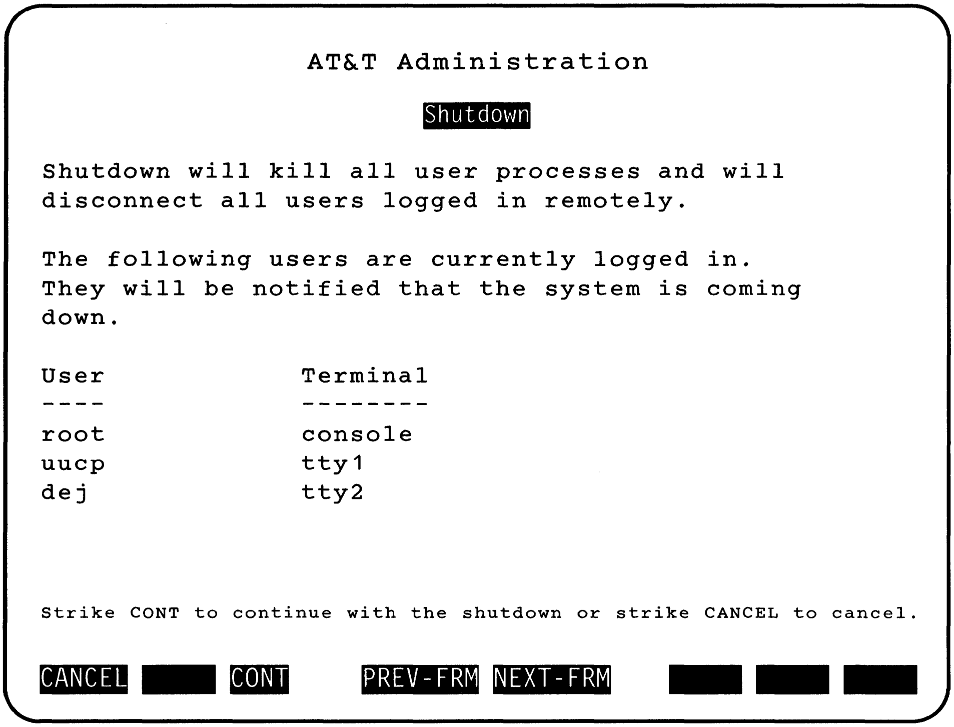
When you are performing this function, the system will ask you to specify how much time (seconds) to wait for users to finish whatever they are doing (i.e., the grace period).

All users are notified that a shutdown will start in "grace period" to allow time for them to finish their work and log out. At the end of the grace period, users that are still logged in receive another message to notify them that the shutdown will resume. Next, all processes are killed and you are notified of the completion of the shutdown. You may then turn off the machine or strike the **RESET** button to reboot the machine as appropriate.

Shutdown Procedures

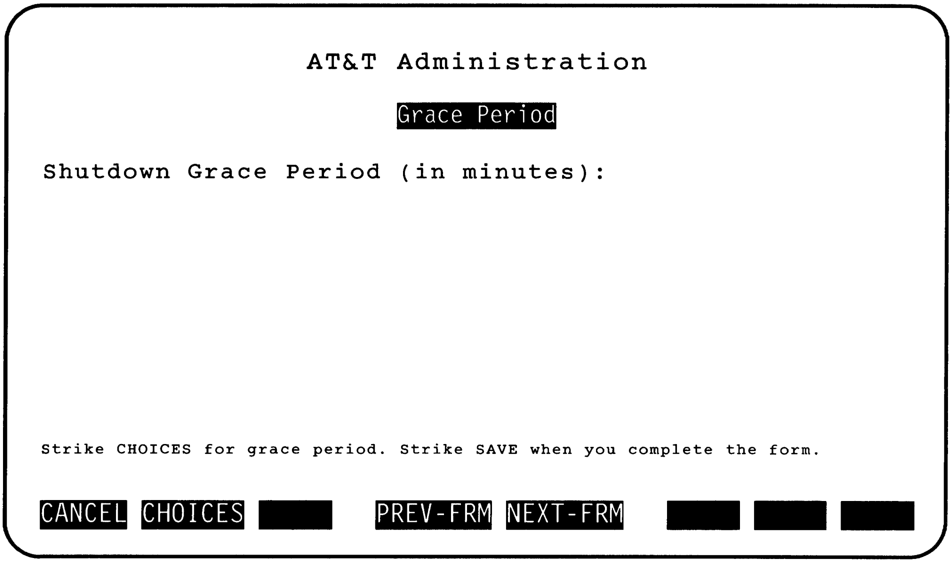
To shutdown your system, do the following:

1. From the Administration menu, highlight Shutdown and strike (Enter). The Shutdown pop-up form appears as follows:



```
                    AT&T Administration

                          Shutdown

Shutdown will kill all user processes and will
disconnect all users logged in remotely.

The following users are currently logged in.
They will be notified that the system is coming
down.

User            Terminal
----            --------
root            console
uucp            tty1
dej             tty2



Strike CONT to continue with the shutdown or strike CANCEL to cancel.
```

2. To continue with the shutdown process, strike CONT. The Grace Period form appears as follows:



```
                    AT&T Administration

                        Grace Period

Shutdown Grace Period (in minutes):











Strike CHOICES for grace period. Strike SAVE when you complete the form.
```

3. Strike CHOICES and select a shutdown grace period from the list that is displayed in the pop-up menu. Move to an item with the arrow keys and strike Enter to select.

4. If you select "0," a shutdown will start immediately. In this case, all remote users (if any) will be notified that a shutdown will start immediately. The following message is displayed on the remote user's screen wherever the cursor is positioned:

   ```
   The system is being shut down NOW!
   ```

5. If you enter a value other than "0," the remote users will get the following message:

   ```
   The system will be shut down in less than X minutes.
   ```

   At the end of the grace period, the remote users will get the following message:

   ```
   The system is being shut down NOW!
   ```

6. You, at the console, will get the following message:

   ```
   The system is coming down.
   Please wait for completion message.
   ```

7. Shutdown will start killing all processes and will bring the system to init state 0 which means it is safe to turn off your machine. At this point, the following message will appear:

   ```
   The system is down. You may turn off the machine
   or strike RESET to reboot.
   ```

System Information

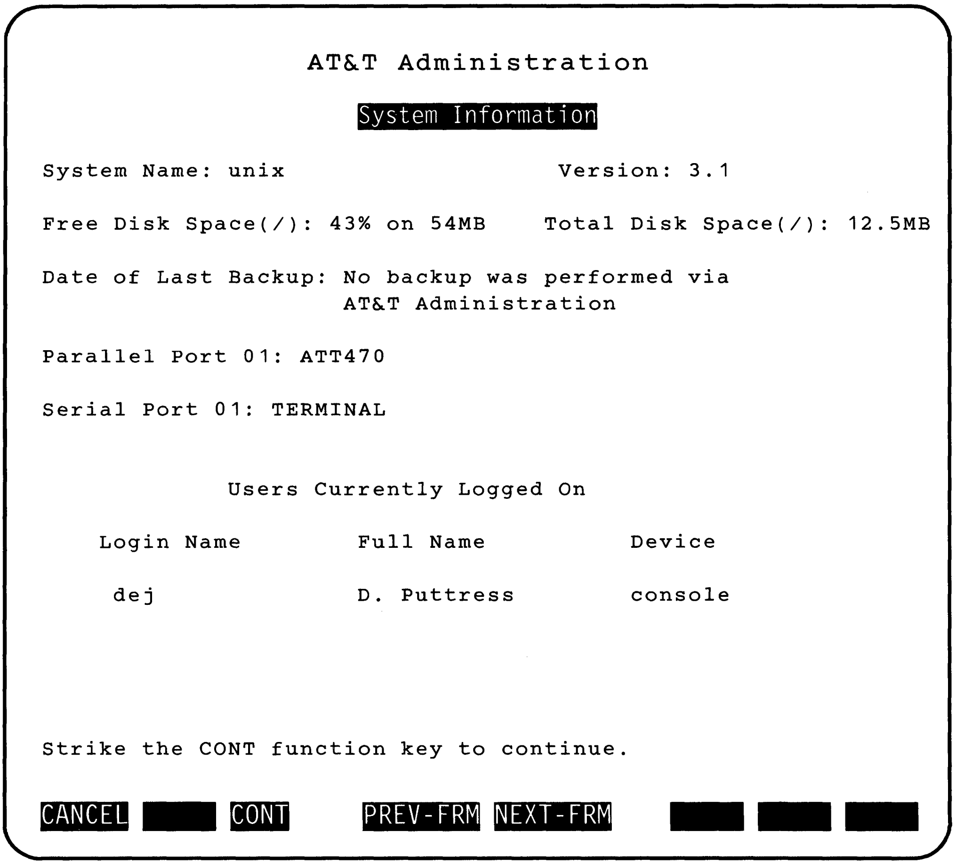
The System Information text frame displays the following user information for the computer.

- System name
- UNIX system version
- Hard disk space
- Floppy disk space
- Mounted file system space
- Device that is on serial ports (default = Not setup via AT&T Administration)
- Device that is on parallel ports (default = Not setup via AT&T Administration)
- Date of last backup (default = No backup was ever performed via AT&T Administration)
- A listing of users currently logged in. This listing includes login names, user's full names, and their device connection with the computer.

To access the System Information form, use the following procedure.

1. From the Administration menu, highlight System Information and strike Enter.

   The System Information text frame appears. If you do not have mounted floppies and multiple file systems, the following is a sample text frame.



```
                      AT&T Administration

                       System Information

System Name: unix                    Version: 3.1

Free Disk Space(/): 43% on 54MB    Total Disk Space(/): 12.5MB

Date of Last Backup: No backup was performed via
                     AT&T Administration

Parallel Port 01: ATT470

Serial Port 01: TERMINAL


             Users Currently Logged On

    Login Name          Full Name          Device

     dej                D. Puttress        console




Strike the CONT function key to continue.
```

NOTE

System Information only reports on ports that have been setup via "Peripheral Setup" and backups that have been done through AT&T Administration.

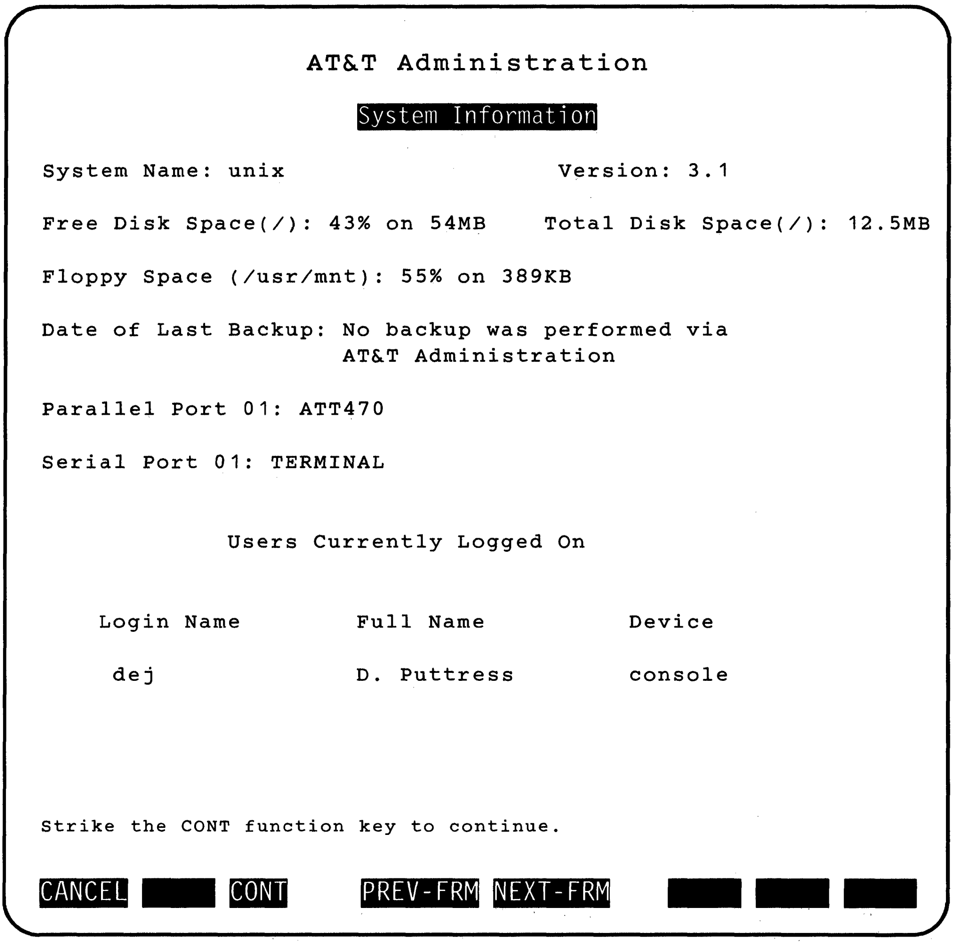
If ports were not set up through AT&T Administration, the following message appears:

```
Not setup via AT&T Administration.
```

If a backup was never performed through AT&T Administration, the following message appears:

```
No backup was performed via AT&T Administration.
```

If you have mounted a floppy disk, the System Information frame will report space on the floppy as shown in the following sample:



```
                    AT&T Administration

                     System Information

System Name: unix                   Version: 3.1

Free Disk Space(/): 43% on 54MB    Total Disk Space(/): 12.5MB

Floppy Space (/usr/mnt): 55% on 389KB

Date of Last Backup: No backup was performed via
                     AT&T Administration

Parallel Port 01: ATT470

Serial Port 01: TERMINAL


             Users Currently Logged On


    Login Name          Full Name          Device

     dej                D. Puttress        console




Strike the CONT function key to continue.
```

2. Strike CONT or CANCEL.

User Logins

The User Logins Administration function provides a method for:

- adding,
- changing,
- deleting, or
- displaying user logins.

You must have system administration privileges to add, change, or delete a user login. However, any user can display user login information.

A login identifies the user and helps prevent unauthorized people from using your computer. But, a login alone can not prevent unauthorized access to your work. Assigning a password to your login name helps guard against unauthorized use.

When a login name is no longer needed, it should be removed. Each login has a login directory or HOME assigned to it. This directory is named */usr/login_name* (where login_name is the name used when you originally added the login).

When you remove a login, you will be asked if you want the files in the login directory removed. To save these files before you delete them, back them up with the Backup to Removable Media function explained in this chapter.

Add User Logins

You must have system administration privileges to add a new user login name to your computer. These privileges are assigned when users are added to the system. Use lowercase letters when assigning logins.

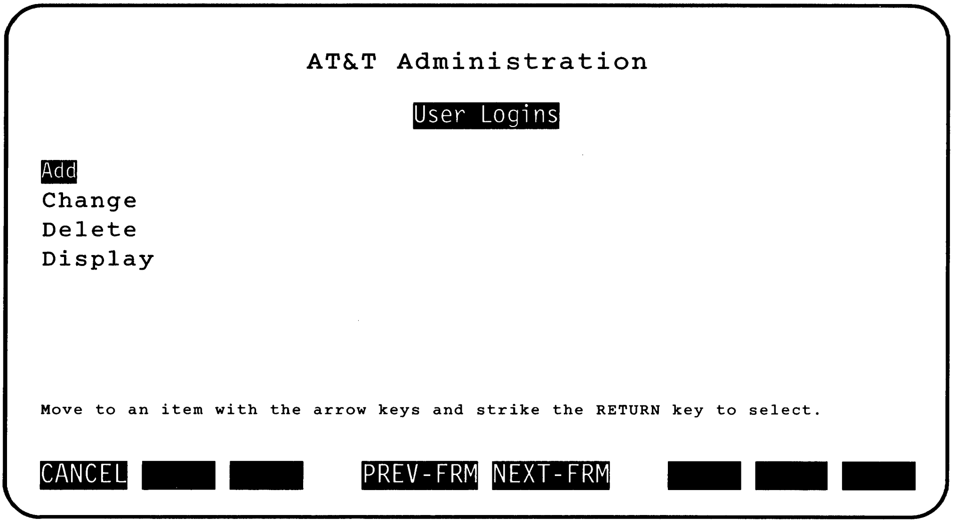
When adding a new user login, the User Logins menu automatically does the following:

- creates a HOME directory(*/usr/<login-name>*),
- chooses the next available uid (user identification number) greater than 100,
- assigns the default gid (group identification number), and
- creates a default *.profile* file.

NOTE Only user logins that were created with AT&T Administration can be administered through the interface.

To add a login, use the following procedure.

1. From the Administration menu, highlight User Logins and strike Enter. The User Logins menu appears as follows:



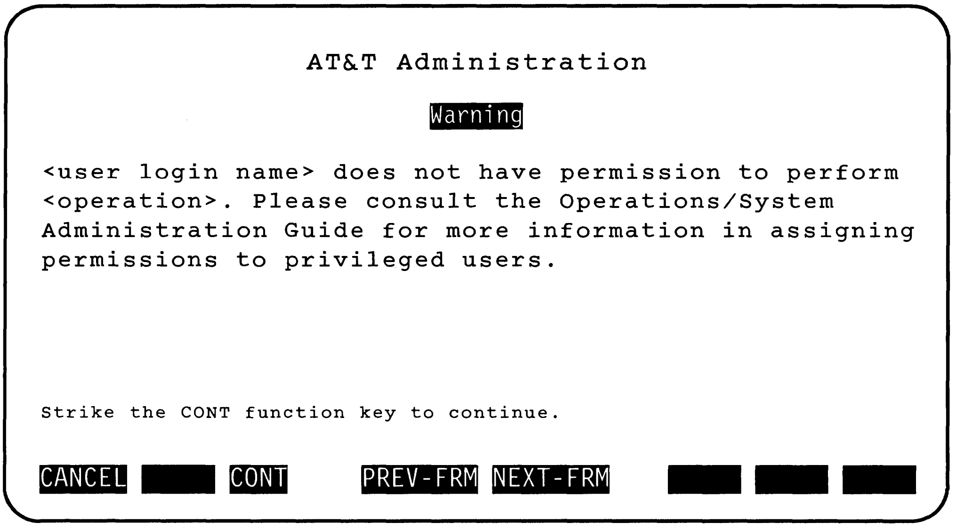
AT&T Administration

User Logins

Move to an item with the arrow keys and strike the RETURN key to select.

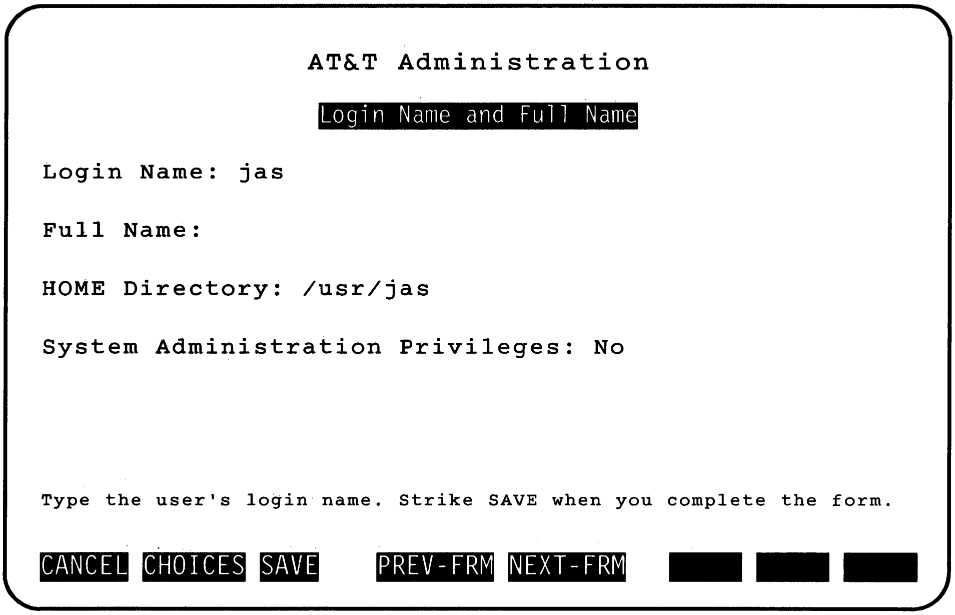
The Add User Logins is a special administrative function that requires the user invoking it to have special system administration privileges.

A validation check, against the login you are currently using, determines if you have system administration privileges. If you do not have system administration privileges, you will receive a warning message as follows:



```
AT&T Administration

Warning

<user login name> does not have permission to perform
<operation>. Please consult the Operations/System
Administration Guide for more information in assigning
permissions to privileged users.



Strike the CONT function key to continue.
```

2. Highlight Add from the User Logins menu and strike Enter. The Login Name and Full Name form appears as follows:



```
                    AT&T Administration

                  Login Name and Full Name

Login Name: jas

Full Name:

HOME Directory: /usr/jas

System Administration Privileges: No



Type the user's login name. Strike SAVE when you complete the form.
```

3. With the cursor resting on the "Login Name:" field, type the new login name to be added (up to eight characters in lowercase letters) and strike Enter.

   If you strike SAVE without giving a login name the following error message will appear:

   > You must provide a login name to add a user.

   The login name you type must be different from all other login names on your computer. You can check the other login names using the Display selection of the User Logins function. If you try to add a login name that already exists, the following message appears:

   > That login name already exists on your system.
   > Type another login name.

   Login names should only be lowercase letters and they can not contain spaces or a ":". If you include one of these characters, the following message appears:

You can not use space or ":" characters in the user login name.

4. With the cursor resting on the "Full Name:" field, type the user's full name.

NOTE	You must provide a full name to add a user. You cannot use a ":" character in the user's full name.

5. A default HOME directory is generated when the user login name is given. You can change the HOME directory by positioning the cursor on the "HOME Directory:" field and typing a new directory name. Do not enter a ":" character in the HOME directory field.

   If you specify a HOME directory that already exists, you will receive the following message:

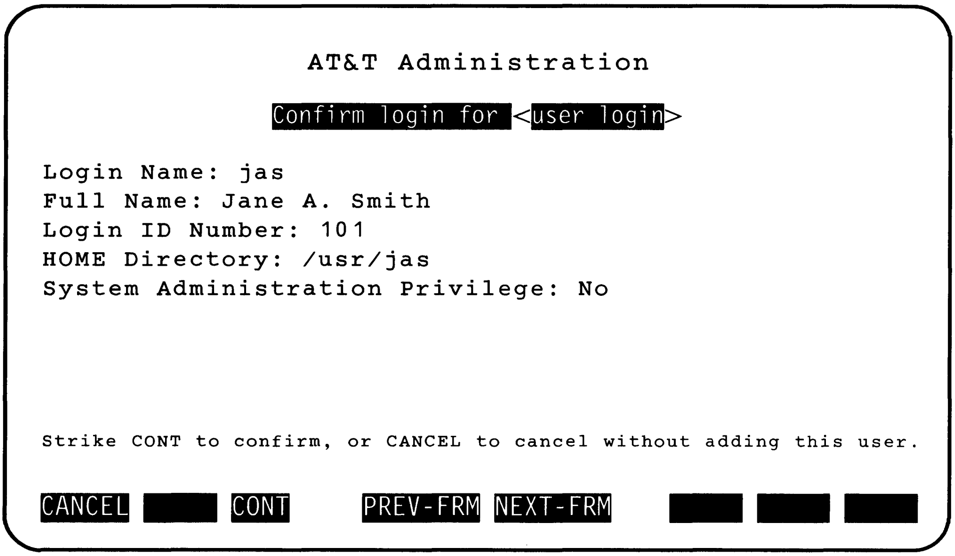
   ```
   <directory name> already exists.
   Type another HOME directory.
   ```

   If you delete the default HOME directory and strike SAVE without specifying another HOME directory, the default will be assigned.

6. System administration privileges always defaults to "No." With the cursor resting on the "System Administration Privileges" field, strike CHOICES. Toggle the system administration privileges to Yes or No as appropriate.

7. Strike SAVE when you complete the form.

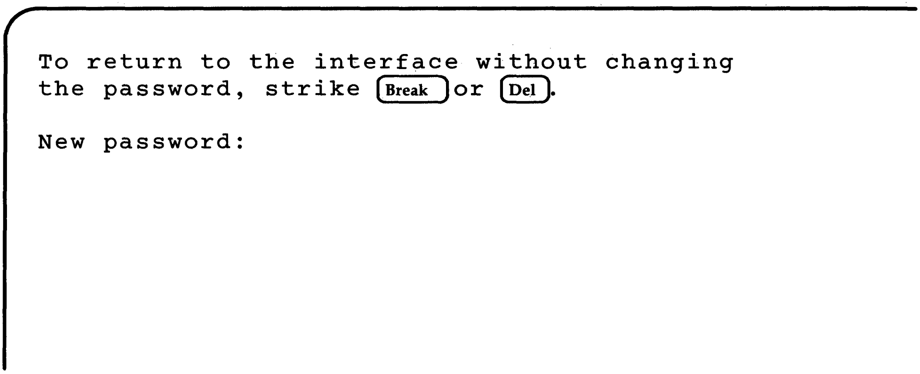
8. The user login information entered is displayed as follows. Please make sure the information is correct.



```
                    AT&T Administration

                 Confirm login for <user login>

Login Name: jas
Full Name: Jane A. Smith
Login ID Number: 101
HOME Directory: /usr/jas
System Administration Privilege: No




Strike CONT to confirm, or CANCEL to cancel without adding this user.
```

9. Strike CONT if the information is correct. If the information is not correct, strike CANCEL.

10. After you strike CONT, you will be prompted for a password. The screen clears and appears as follows:



```
To return to the interface without changing
the password, strike Break or Del.

New password:
```

Respond to the prompts to assign a password for the new user login name. When you have assigned a password, you will be prompted as follows:

Strike RETURN to continue.

Strike Enter.

NOTE A password is required for every login. If you do not provide a password, the users login will not be installed.

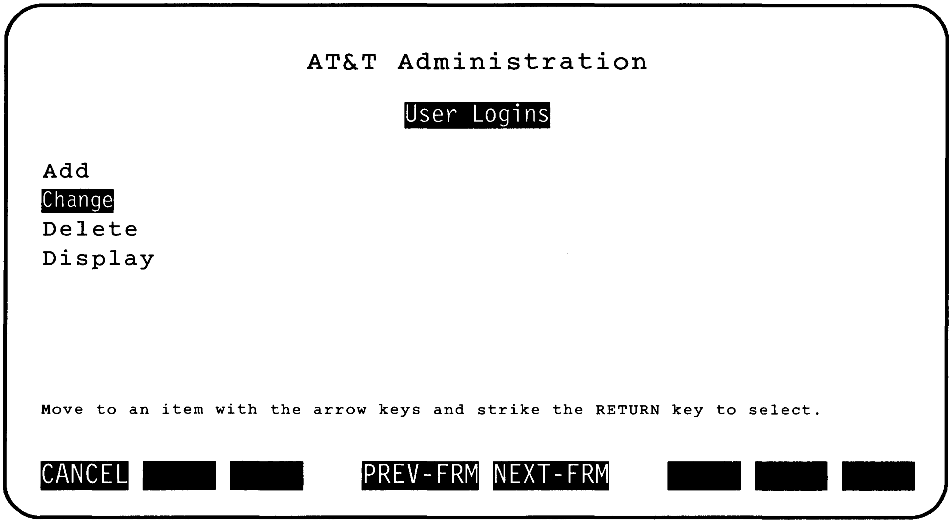
11. After you provide a password, you will return to the User Logins menu.

Change User Logins

To change user login information, use this procedure.

NOTE You must have system administation privileges to change a user login.

1. From the Administration menu, highlight User Logins and strike Enter. The User Logins menu appears as follows:

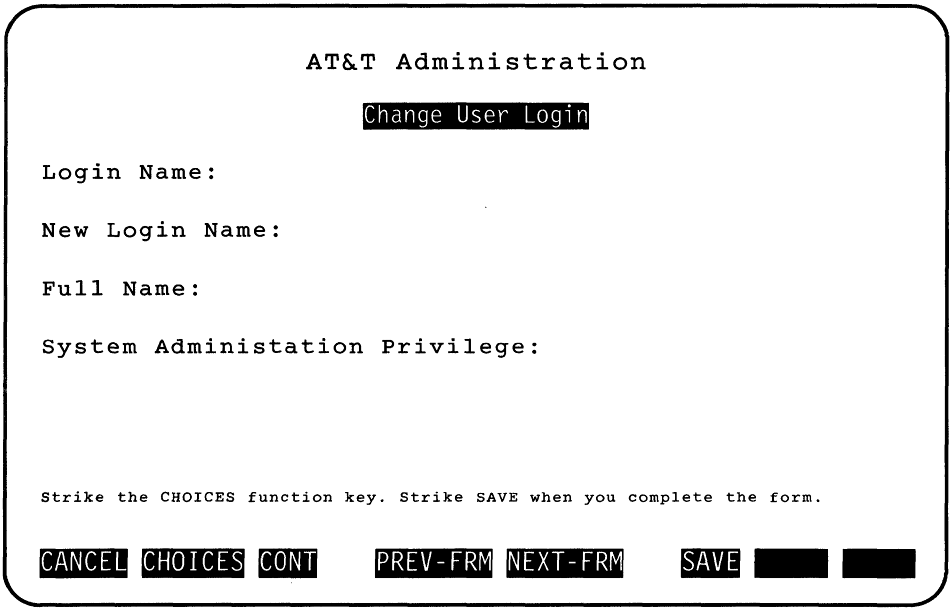


```
                    AT&T Administration

                        User Logins

Add
Change
Delete
Display



Move to an item with the arrow keys and strike the RETURN key to select.
```

2. Highlight Change from the User Logins menu and strike Enter.

NOTE A validation check determines if you have special administration privileges. If you do not, you will receive a warning message. You must have special administration privileges to change logins not belonging to you.

The Change User Login form appears as follows:



```
                    AT&T Administration

                     Change User Login

Login Name:

New Login Name:

Full Name:

System Administation Privilege:




Strike the CHOICES function key. Strike SAVE when you complete the form.
```

3. If you do not wish to change the login name, move the cursor to "Full Name:" and skip to Step 5. While the cursor is resting on the "Login Name:" field, strike CHOICES and select the desired login from the pop-up menu of user logins. If you are using the pop-up list, move to an item with the arrow keys and strike Enter. The login to be changed is now entered in the login name field.

NOTE — If there are fewer than four logins on the computer, the form will automatically cycle through the existing ones.

You could type the login name, rather than select it from CHOICES. If you type the login name incorrectly, the following message appears:

> This is not a valid login name.
> Strike CHOICES for valid choices.

When you provide a login name, the interface will fill in the remaining values on the form except for New Login name.

You can give new values for the Full Name and System Administration Privileges, but you can not use the Login Name field to change the Login name. You must use the New Login Name field.

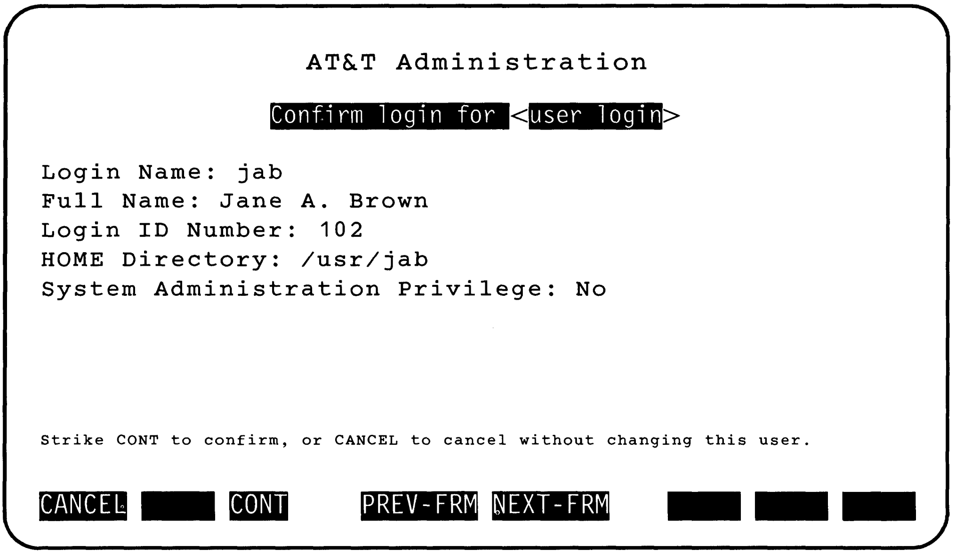
4. With the cursor resting on the "New Login Name:" field, type the new login name. If the new login name is not unique, you will recieve the following mesage:

```
That login name already exists on your system. Type another login name.
```

NOTE

The contents of the old HOME directory is retained. The new login name is automatically assigned as the owner of all the former login name's files. If you wish to change the name of the HOME directory to coincide with the new login name, you must create a new directory and move the files from the old to the new directory (**mvdir** *old-dir new-dir*) without the assistance of the interface.

5. If you do not wish to change the user's "Full Name:", move the cursor to "System Administration Privileges:" and skip to Step 6. With the cursor resting on the "Full Name:" field, type the user's new full name if any. If the user's full name is deleted, the old full name is used.
6. If you wish to change the "System Administration Privileges:", continue. With the cursor resting on the "System Administration Privileges:" field, strike CHOICES and toggle Yes or No until the appropriate choice is made.
7. Strike SAVE when you complete the form.
8. Check the Confirm form that appears to make sure the information is correct. A sample confirm form follows:



```
                    AT&T Administration

                 Confirm login for <user login>

Login Name: jab
Full Name: Jane A. Brown
Login ID Number: 102
HOME Directory: /usr/jab
System Administration Privilege: No




Strike CONT to confirm, or CANCEL to cancel without changing this user.
```

9. Strike CONT if the information is correct to close the Confirm Login Form and make the User Logins menu active. If the information is not correct, strike CANCEL.

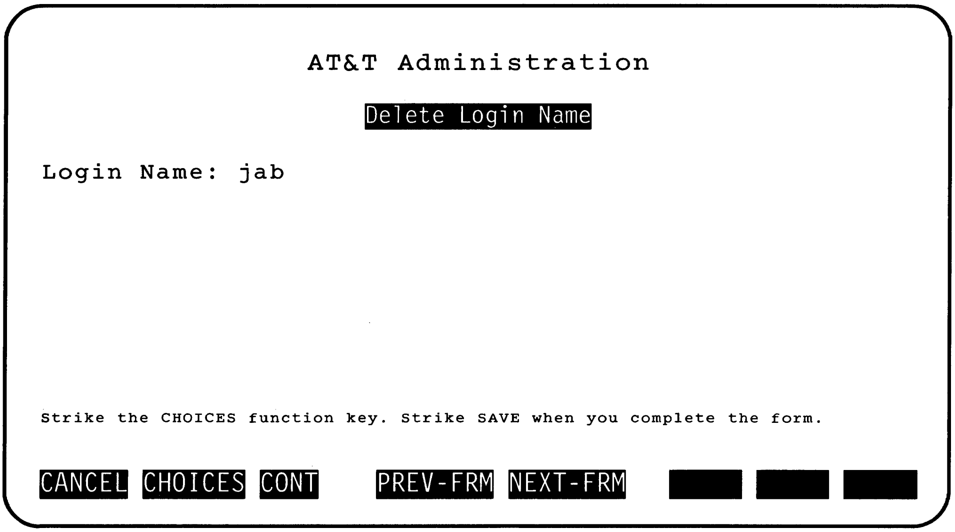
Delete User Logins

When you no longer need a login, it should be removed.

1. From the Administration menu, highlight User Logins and strike Enter.
2. From the User Logins menu, highlight Delete and strike Enter.

NOTE: A validation check determines if you have special system administration privileges. If you do not, you will receive a warning message. You must have special system administration privileges to delete a login even if it is your own.

The Delete Login Name form appears as follows:



AT&T Administration

Delete Login Name

```
Login Name: jab
```

Strike the CHOICES function key. Strike SAVE when you complete the form.

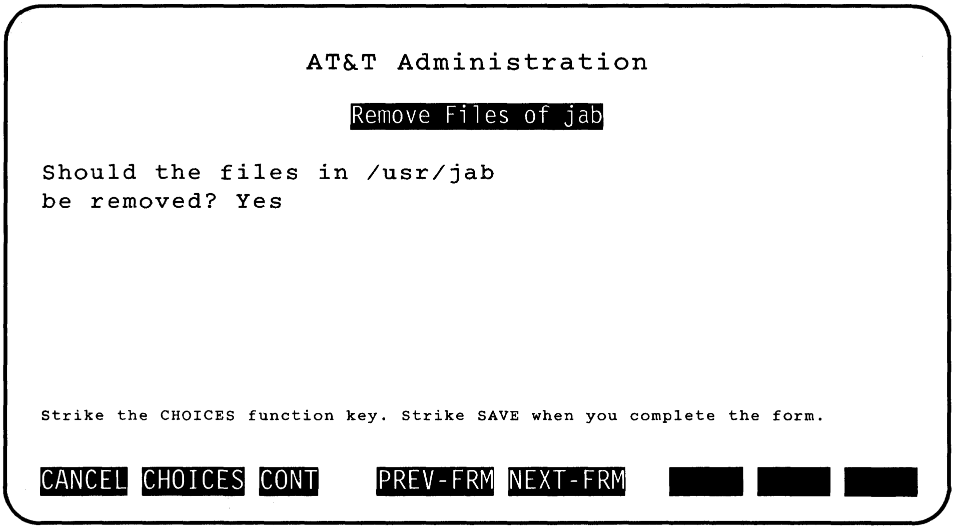
3. While the cursor is resting on the Login Name field, strike CHOICES and select the desired login from the pop-up menu of user logins. Move to an item with the arrow keys and strike the Enter key to select. The login to be deleted is now entered into the Login Name field.

You can also type in the login name that you want to delete. If you type the login name incorrectly, the following message appears:

```
That is not a valid login name.
Strike CHOICES for valid choices.
```

If the user you select is currently logged on, you will be warned. The user will be able to continue working if you delete the login, but the user will not be able to login again.

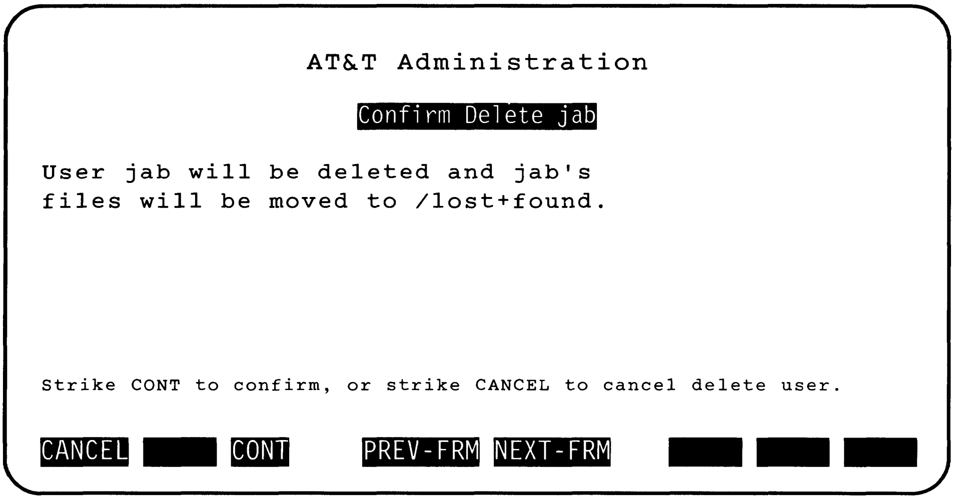
4. When you delete a user login name, you must decide if you want to remove all the files from the user's HOME directory. The Remove Files of <user login> form appears as follows:



```
                    AT&T Administration

                    Remove Files of jab

Should the files in /usr/jab
be removed? Yes






Strike the CHOICES function key. Strike SAVE when you complete the form.
```

5. The default decision is Yes. Strike CHOICES (Yes or No) until your decision appears (toggle). When you complete the form, strike SAVE.

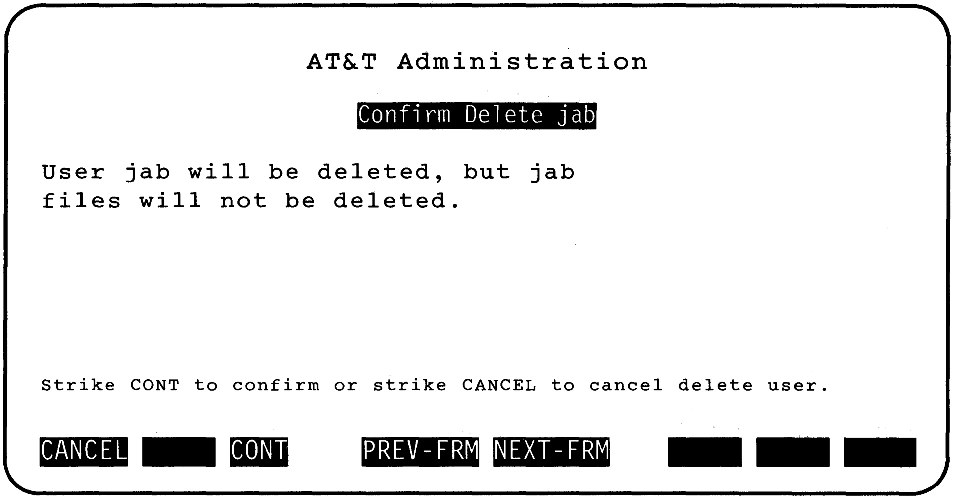
If you responded Yes to delete the files the following confirm frame appears:



```
                    AT&T Administration

                    Confirm Delete jab

User jab will be deleted and jab's
files will be moved to /lost+found.




Strike CONT to confirm, or strike CANCEL to cancel delete user.
```

The files will be stored temporarily in */lost+found*. Then they will be deleted.

Responding "No" to delete the files causes the following confirm frame to appear:



```
                     AT&T Administration

                     Confirm Delete jab

User jab will be deleted, but jab
files will not be deleted.




Strike CONT to confirm or strike CANCEL to cancel delete user.
```

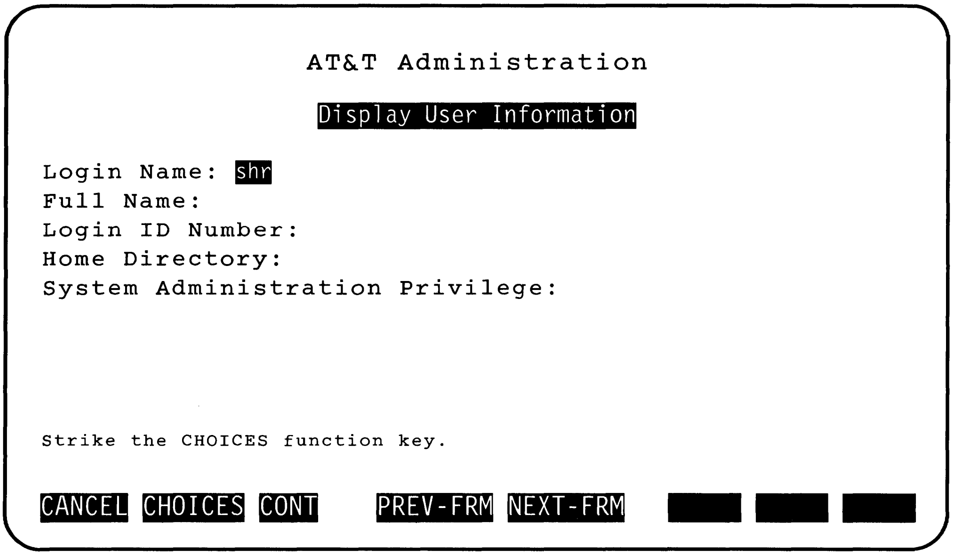
6. Strike CONT if you're sure you want to delete the login. If you do not want to delete the login, strike CANCEL.

   When you strike CONT the Confirm message closes and the User Logins menu becomes active. You can remove another login if desired or go to another menu.

Display User Logins

Any user can display login information for other user's logins. To display user login information, use this procedure.

1. From the User Logins menu, highlight Display and strike (Enter). The Display User Information form appears as follows:



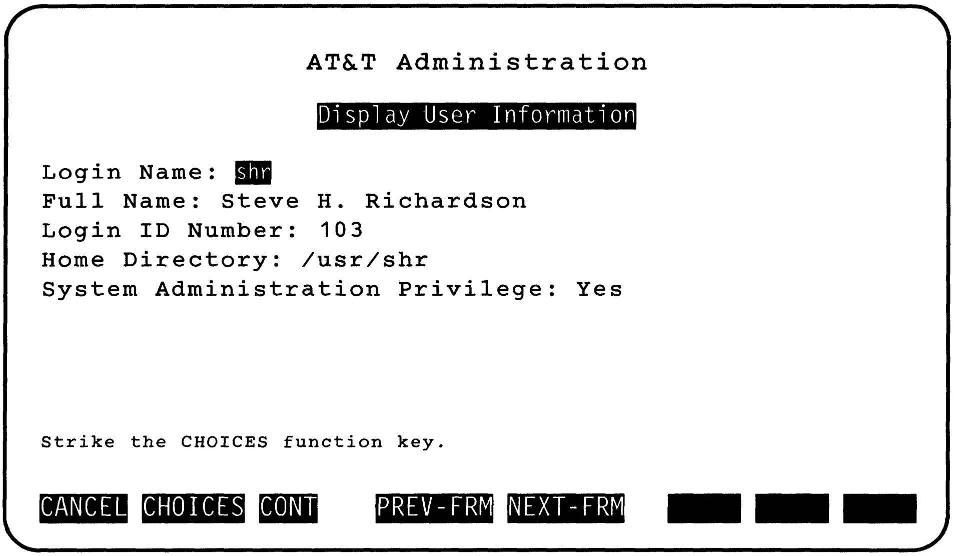
```
                AT&T Administration

              Display User Information

Login Name: shr
Full Name:
Login ID Number:
Home Directory:
System Administration Privilege:




Strike the CHOICES function key.

CANCEL CHOICES CONT    PREV-FRM NEXT-FRM
```

2. While the cursor is resting on the Login Name field, strike CHOICES and select the desired login from the pop-up menu of user logins. If you are using the pop-up menu, move to an item with the arrow keys, and strike the (Enter) key to select. The Display User Information form will display the information for the login name selected.

   You can type the login name, but if you type it incorrectly, the following message appears:

   > This is not a valid login name.
   > Strike CHOICES for valid choices.

   After you select the login name, the other fields of the form will display the information for that login as follows:



```
                     AT&T Administration

                   Display User Information

Login Name: shr
Full Name: Steve H. Richardson
Login ID Number: 103
Home Directory: /usr/shr
System Administration Privilege: Yes




Strike the CHOICES function key.
```

3. Strike CONT to make the User Logins menu active, CANCEL to return to the Administration menu, or repeat selection of Login Names to display other logins.

Printer Operations

After you set up a printer through Peripheral Setup, you use the Printer Operations entry from AT&T Administration to:

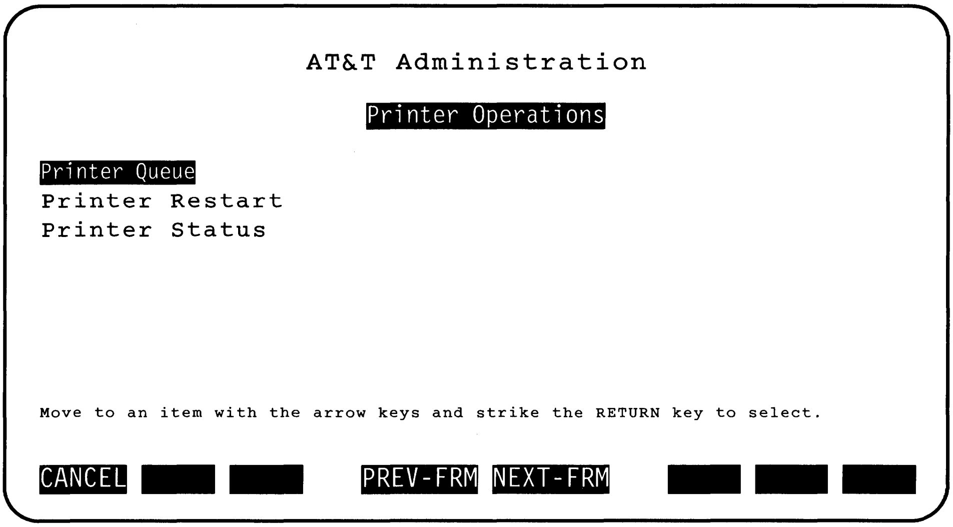
- display a list of current jobs queued to the printer(s),
- restart the printer(s), and
- display status information for the printer(s).

NOTE If your parallel or serial printer has not been set up in advance, you are notified with a warning message.

To display printer information, use the following procedure.

1. From the AT&T Administration main menu, select `Printer Operations` and strike Enter.

   The Printer Operations menu appears as follows:



```
                    AT&T Administration

                     Printer Operations

Printer Queue
Printer Restart
Printer Status




Move to an item with the arrow keys and strike the RETURN key to select.
```

The Printer Operations menu has three selections:

- Printer Queue: Shows a list of print jobs in the order in which they will print. From the printer queue, you can also cancel a job.
- Printer Restart: Permits restart of the printer scheduler and enables the printer.
- Printer Status: Shows the status of the printer and any print jobs that are queued.

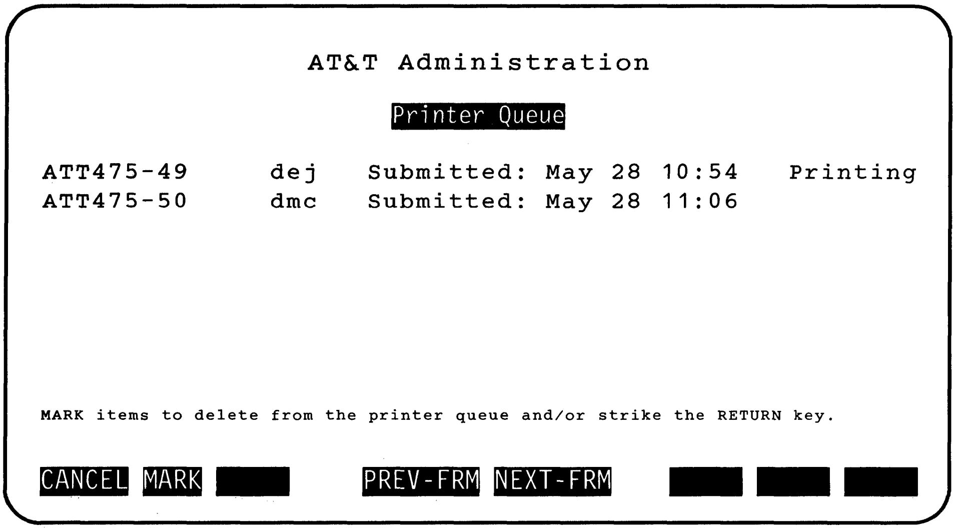
Printer Queue—Displaying Queued Jobs

A list of jobs queued to the printer(s) can be displayed by using the following procedure.

1. From the Printer Operations menu, select Printer Queue and strike [Enter].

   The Printer Queue menu appears as shown in the following screen. The list includes the following.

   - printer name,
   - a job ID number,
   - the user login of the user that queued the job,
   - a time stamp of when the job was submitted, and
   - which job is currently printing.



```
                        AT&T Administration

                           Printer Queue

ATT475-49     dej   Submitted: May 28 10:54    Printing
ATT475-50     dmc   Submitted: May 28 11:06







MARK items to delete from the printer queue and/or strike the RETURN key.
```

2. Once you have seen the print jobs in the queue, strike CANCEL to close this frame and return to the Printer Operations menu without deleting any jobs.

NOTE Only 30 print jobs can be confirmed at one time.

Printer Queue—Canceling a Print Job

You can use the Printer Queue menu to cancel a job that is queued to print. To cancel a print job, use the following procedure.

1. From the Printer Operations menu, select `Printer Queue` and strike [Enter].

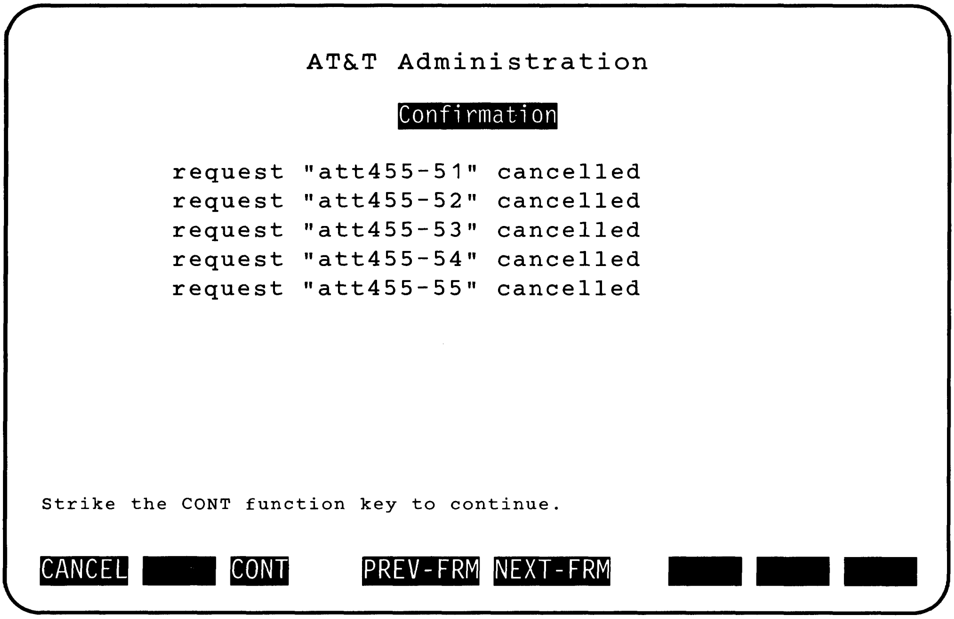
   The printer queue appears.

2. Highlight the print job(s) that you want to cancel by striking MARK for each selection.

   NOTE Striking [Enter] while the cursor rests on a print job display will not cause that print job to be marked for deletion, but it will cause you to leave the print job display. Use the cursor-control keys to move to the print job you want to delete. Then you must use the MARK function key to mark jobs to be deleted.

3. Repeat Step 2 until you have selected all the print jobs that you want to cancel.

4. After you have marked the items to delete from the printer queue (this prevents the specified jobs from being printed) and striked [Enter], you will receive the following confirmation form:



```
                AT&T Administration

                    Confirmation

      request "att455-51" cancelled
      request "att455-52" cancelled
      request "att455-53" cancelled
      request "att455-54" cancelled
      request "att455-55" cancelled




Strike the CONT function key to continue.
```

Printer Restart

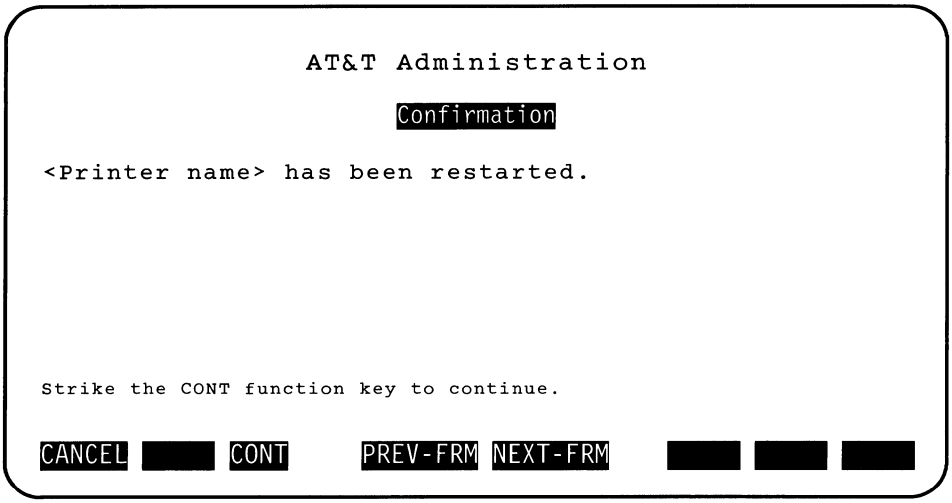
When you set up a printer, the software for it is automatically started. If you're having trouble getting something to print on your printer, you may need to restart the parallel or serial printer scheduler.

NOTE If you have trouble with printer output, be sure the power is turned on for the printer. Also check to see if the printer cable is plugged in securely and the printer is "On Line" or "Ready."

NOTE When the same printer is restarted, the jobs queued to this printer will resume printing.

To restart the printer:

1. Highlight `Printer Restart` from the Printer Operations menu and strike [Enter]. The Printer Name form appears.
2. Strike the CHOICES function key to obtain a list of the printers you have previously set up via Printer Setup. Select the printer to restart and strike SAVE. You will receive the confirmation form as follows:



```
                    AT&T Administration

                        Confirmation

<Printer name> has been restarted.




Strike the CONT function key to continue.
```

3. Strike CONT to erase this frame and display the Printer Operations menu. Striking CANCEL closes this frame and makes the Printer Operations menu active. The printer resumes printing or is ready to print.

4. Strike CONT or CANCEL to close this frame and display the Printer Setup menu.

Printer Status

You can use the printer status form to display a list of the printer(s) currently set up and information that includes the following:

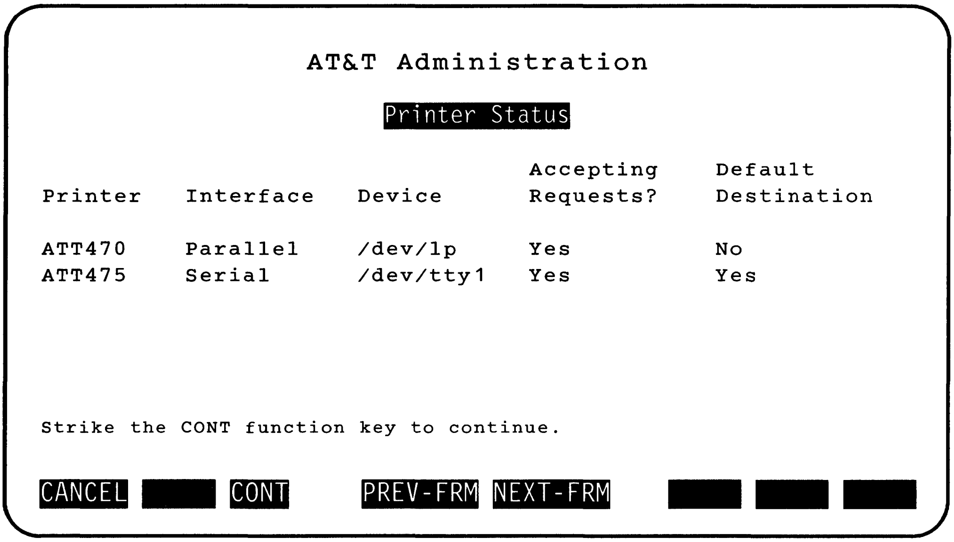
- the printer name,
- the interface connection (i.e., parallel or serial),
- the UNIX system port,
- whether the printer(s) is currently accepting requests, and
- which printer has been set up as the default destination.

NOTE

If you have trouble with printer output, be sure the power is turned on for the printer. Also check to see if the printer cable is plugged in securely and the printer is "On Line" or "Ready."

1. From the Printer Operations menu, select Printer Status and strike Enter.

   The sample output appears as follows:



```
                    AT&T Administration

                       Printer Status

                                          Accepting      Default
Printer     Interface    Device           Requests?      Destination

ATT470      Parallel     /dev/lp          Yes            No
ATT475      Serial       /dev/tty1        Yes            Yes




Strike the CONT function key to continue.
```

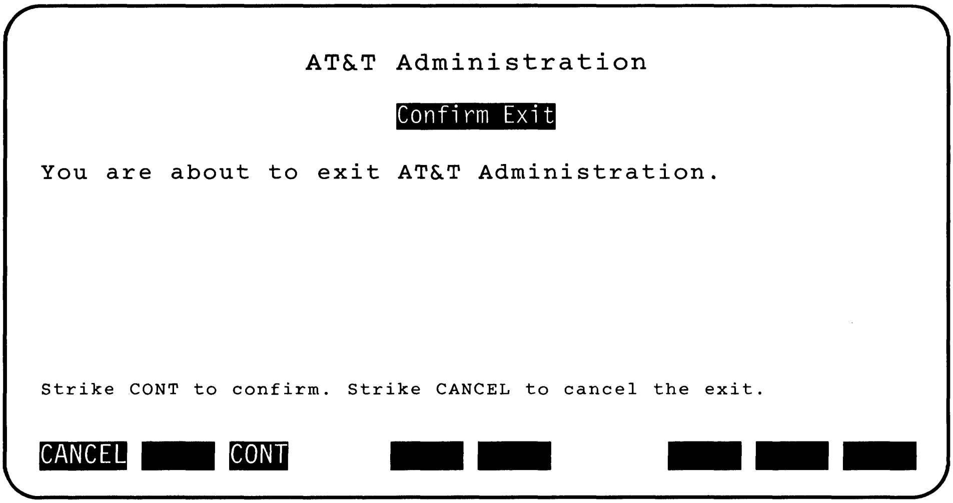
2. Once you have seen the information you requested, strike CONT or CANCEL to close the frame and make the Printer Operations menu active.

Exit

You can use the Exit selection to exit the interface and return to the UNIX system shell. To exit the interface, use the following procedure:

1. From the AT&T Administration main menu, select Exit and strike Enter.

   To make sure that you really want to exit, you are asked if you want to continue as follows:



```
                     AT&T Administration

                         Confirm Exit

You are about to exit AT&T Administration.




Strike CONT to confirm. Strike CANCEL to cancel the exit.
```

2. Strike the CONT key to confirm and exit the interface so that you can return the UNIX system $ prompt. Strike CANCEL to cancel the exit.

UNIX System

Selecting the UNIX System from the AT&T Administration main menu is another way of returning to the UNIX system shell. This creates a sub-shell that takes up the entire screen. At the top of the screen it displays the current directory and gives you the following directions

type exit or control d to return to AT&T Administration.

The prompt for this sub-shell is `$$` instead of the `$` prompt so that you will know when you are in a sub-shell.

To return to the interface, use the following procedure.

1. Type

```
$$ exit Enter

        or

Ctrl d
```

2. When you are prompted to press RETURN to continue, strike the Enter key to return to AT&T Administration.

You can run an application or execute commands in the sub-shell, but when you return to the interface, the sub-shell is terminated as are all processes running within it. More about the UNIX system shell is described in Chapter 3, Using the UNIX System Shell.

Customizing Your Computer

Customizing Your Computer

Chapter 5: Customizing Your Computer

Tailoring Your Environment

This chapter describes how to tailor or customize your computer environment to perform the duties unique to you and the other people who use it.

The System Default Profile

When you first log in, the UNIX system establishes your working environment as defined by the default system *profile*. The default system *profile,* located in the */etc* directory (*/etc/profile*), contains the commands needed to initialize your environment and commands common to all users. You must be logged in as **root** to modify the default system *profile*. However, the *.profile* (located in your login directory) contains the commands specific to you to set up your own environment. The variable MAIL, PATH, T2, erase character, etc, are set up here. So, your *.profile* is used to taylor your environment to your needs. For example, if you frequently log on with a specific remote terminal, you can set TERM to that terminal. You can use your own *.profile* (just like a regular file) to include other initialization commands.

After creating or making changes to the *.profile,* you can initiate the changes without logging off and logging back in again by typing the following:

```
. .profile Enter
```

The shell will reinitialize your environment. The dot (**.**) is a special shell command used to execute commands in the *.profile*.

The following is an example of a *.profile*. The lines that start with a # sign are comment lines. The comment lines are only in this example to explain what some of the commands are.

```
# Set command search path.
PATH=$PATH:$HOME/bin:
# Set the shell prompt to something other than $.
PS1=prompt
# Have mail printed when you log in.
mail
# Output the system date and time.
date
```

During login, the */etc/profile* file will be read to initialize your environment, and then your *.profile* will be read to execute any other commands you may have specified.

Some commands are executed based on the variables defined in the default environment. The following is a list of those variables:

CDPATH Defines the paths to be searched for an argument to the **cd** command. By default, the current directory is searched.

LOGNAME Defines your login name and is set when you log in to the system. This variable is often referred to by shell programs. **LOGNAME** is specified in */etc/passwd.*

HOME Defines pathname of your login directory. The value of **HOME** is set when you log in and should not be changed. **HOME** is specified in */etc/passwd.*

IFS Defines the internal field separator characters. The shell initially sets these characters to include the space (blank), tab, and new-line characters.

MAIL Defines the full pathname where you will receive mail from other users. **MAIL** is usually kept in */usr/mail* (i.e., **/usr/mail/***LOGNAME*).

PATH Defines the directory search path for commands. By default, this variable includes the current directory, the **/bin, and /usr/bin** directories.

PS1 Defines the primary shell prompt. By default, the shell prompt is set to **$** for regular UNIX system users and **#** for users that log in as **root**.

PS2 Defines the secondary shell prompt. By default, the secondary shell prompt is set to >. This prompt means that additional information (input) is needed for the command to run.

TERM Defines your terminal type for certain programs (such as screen editors). By default, **TERM** is set to **AT386**.

TZ Defines the time zone.

The following are a few hints to use when you're working with .profile(s):

- To change a variable, type the following:

```
variable=new_variable Enter

export variable Enter
```

For example, to change your secondary shell prompt (PS2) to the word "MORE", you would enter the following:

```
PS2=MORE Enter

export PS2 Enter
```

- To look at the variables in your environment, type the following:

```
env Enter
```

- To reinitialize your environment, type the following:

```
. .profile Enter.
```

Changing the News

News is a type of news brief displayed to users as they log in. The **news** directory is located in */usr/news* and contains news files for people to read. There is one **news** file per news item. To add or change **news**, create a file in the **news** directory and enter the news you want system users to read. To read the news after logging in, just type **news** Enter. See **news**(1) in the *User's/System Administrator's Reference Manual.*

Changing Message of the Day

The message of the day is a brief item of information displayed to all users as they log in. If you are going to update the message of the day, you must use the **root** login. The message is contained in the */etc/motd* file and can be changed using any text editor. You should try to keep the message of the day relatively short and to the point. Save your longer messages for the */usr/news* file(s).

Changing the Path

You can add your own directory of commands to your path by adding to your *.profile*:

```
    PATH=$PATH:$HOME/bin; export PATH
```

where **bin** is the name of your command directory.

Automatic Program Execution

The UNIX system allows you to have programs run automatically at specified times. This is done with the **cron** program. The **cron** program and, more specifically, the **crontab** command allows you to run programs during off-hours such as:

- File system administration
- Long-running, user-written shell procedures

- Cleanup procedures.

Any task that needs to be done repeatedly at a specified time is a candidate for your *cron* file located in the */usr/spool/cron/crontabs* directory. You can use the **crontab** command to establish the entries you want.

The **crontab** command is used as follows:

crontab *file* Enter

crontab *-r* Enter

crontab *-l* Enter.

The **crontab** command copies the specified *file* or standard input if no file is specified into a directory that holds all users' crontabs. The *-r* option removes a user's crontab from the *crontab* directory. The *-l* option will list the *crontab* file for the invoking user. See the **crontab**(1) command in the *User's/System Administrator's Reference Manual* for additional information.

Each line in the *crontab* file defines one procedure. The line entry format looks like the following:

minute hour day month day-of-week command

Each field is defined as follows:

minute (0-59),
hour (0-23),
day of the month (1-31),
month of the year (1-12),
day of the week (0-6 with 0=Sunday)
command - the command to be executed at the time specified.

The following rules apply to the first five fields:

- Two numbers separated by a hyphen indicate a range of numbers between the two specified numbers.
- A list of numbers separated by commas indicates only the numbers listed will be used.
- An asterisk specifies all legal values.

For example, 0 0 1,14 * 2 indicates a command will be run on the first and fourteenth of each month, as well as on every Tuesday. If a percent sign (%) is placed in the command field (sixth field), the UNIX system will translate it as a new-line character. Only the first line of a command field (character string up to the percent sign) is executed by the shell. Any other lines are made available to the command as standard input.

For example, let a file called "anyfile" contain the following **cron** entry:

```
0 0 1 * * mailx $LOGNAME % Subject: Call Mom! % now
```

When the command line **crontab anyfile** Enter is executed, the user whose login is $LOGNAME will get a reminder mail message with "Call Mom!" as the subject the first of every month.

Automatic System Cleanup

The UNIX system has to be cleaned up occasionally. Fortunately, you can get out of some cleaning with the help of the **crontab** command and the **crontab** file. You can specify cleanup jobs (e.g., remove aged files) and the time you want them to execute in the *crontab* file.

Your computer comes with some default cleanup procedures already defined. These cleanup procedures are done by the **root** login under the control of **crontab** each Sunday morning at 5:17. The file */etc/cleanup* defines what cleanup procedures are done.

Some of the files cleaned up each Sunday morning are as follows:

- */etc/wtmp*: This file contains a history of system logins. Every time a user logs in, a record is made in this file. As you can see, the size of this file grows forever, and it needs to be limited. Instead of cleaning it up manually, you can have **cron** do it for you.
- */usr/adm/sulog*: This file contains a history of users that use the **su** command to switch logins. As a security measure, this file should not be readable by other users. See the **su**(1) manual pages in the *User's/System Administrator's Reference Manual* for additional information.
- */usr/adm/cronlog*: This file contains a history of all actions taken by **cron**.

By logging in as **root** and executing `crontab` *-l,* you will see the **crontab** entry that executes */etc/cleanup* as well as other cleanup routines for UUCP (Basic Networking). By examining */etc/cleanup,* you will see the default routines done each Sunday morning. You can edit */etc/cleanup* and modify the **root crontab** (the **root crontab** is stored in */usr/spool/cron/crontabs*) to do cleanup jobs differently if you wish.

Administering System Security

Security should be given special consideration on a multiuser system. The information you have stored on your computer belongs to you and no one else. Your information is a valuable resource that requires protection. The computer provides some of the most sophisticated security features available among personal computers. This section will point out the ways you can help keep the information in your computer secure. Four security features provided for your computer are:

- System backups
- Access permissions
- Passwords
- Data encryption.

Sources of Potential Damage

To provide adequate security for your system, you first need to consider what kinds of problems might arise. It is important to consider every possible contingency: theft of the hardware, breakdowns in the hardware or software, tampering with data in the computer by unauthorized people, theft of data, and simple human error. To determine the types of problems to which your system may be vulnerable, ask yourself the following questions.

- Are you using your computer in a large or a small office? Is it possible for someone to steal your computer?

  Prevent theft by setting up your computer in a secure place.

- How many people have ready access to your system?

  Restrict the number of people with access to the computer by using passwords.

- How sensitive or valuable is the information you are storing?

  Restrict the number of people who have access to sensitive data with access permissions. Protect your data by encryption.

- Do you transmit data over telephone lines?

  Protect your data by encryption. Limit access to your computer telephone numbers.

The way you answer these questions will help you decide the measures you want to take to safeguard your data.

Basic Precautions

The following safeguards are recommended to everyone for ensuring the security of their computer and data.

- Set up your computer in an area that can be secured when you are not using it.
- Never leave your computer terminal logged in and unattended.
- If you are using your computer in an office environment, restrict the number of people who have access to the computer.

System Backups

If a breakdown occurs in the system, you may lose information that you had stored in the computer. You can avoid this type of loss if you have copies of your files stored on removable storage devices. Copy your data periodically onto floppy diskettes (or cartridge tapes) and store the diskettes (or tapes) in a safe place, away from the computer site.

This procedure is known as "backing up the system." With your files backed up regularly and stored safely off site, you won't have to worry about losing your information, even if your files are damaged.

There are two types of backups, system and incremental. In a system backup, you copy all mounted file systems that have been modified or created since the system was installed. In an incremental backup, you make copies of only those files that have been modified since the last backup.

Access Permissions

If you are sharing your computer with other users, you may want to share some information while keeping other information private. The UNIX Operating System allows you to satisfy both these needs by letting you define the following:

- The users that have permission to access data
- The types of permission they have (that is, how they are allowed to use the data).

The UNIX Operating System recognizes three categories of users of stored data: the owner of the data, the group to which the owner belongs, and all other people who use the system. Users can be given permission to use data in any or all of three ways: to read, write, and/or execute it.

Whenever you create a file (or directory), the system automatically identifies the users who will be allowed to read, write, and/or execute the file. However, as the owner of the file, you have the ability to change the access permission after it has been created. The only person besides you who can change the permission on your file is the holder of "super user" privileges, usually the system administrator.

One way to restrict access to your files is to change the permission modes. There are three sets of permission modes assigned to a file or directory: you (the user), the group, and everybody else. The permission modes are displayed every time you use the long list (**ls -l**) command. See the **ls**(1) and **chmod**(1) manual pages in the *User's/System Administrator's Reference Manual* for additional information.

The following is an example of the permission modes of a file or directory:

```
-rwx rwx rwx (file)
drwx rwx rwx (directory).
```

Note that the first set of **rwx** refers to the permission mode for you (the owner), the second for the group, and the last for everybody else.

r Allows you to read a file or to copy its contents.

w Allows you to write changes into a file or copy a file to a directory.

x Makes the file executable, also allows a directory to be searched.

Whenever a file or directory is created, the permission modes are automatically set for everyone to have access. Each permission mode is based on a number:

4 = read
2 = write
1 = execute.

You can modify the permission modes of your file with the **chmod** command.

For example, to change the file *mycmd* so everyone can only execute it, enter the following command:

```
chmod 751 mycmd [Enter]
```

The permission modes for the file *mycmd* are now changed to rwxr-x--x. The 7 assigns the owner read, write, and execute permissions [4 (read) + 2 (write) + 1 (execute) = 7]. The 5 assigns the group read and execute permissions [4 (read) + 1 (execute) = 5]. The 1 assigns everybody else execute permission. See **chmod**(1) manual pages for additional information.

Also, to keep people from looking at the contents of your directory, you can deny execute permission to that directory. This has the effect of preventing others from visiting your directory.

You can also use the **umask** command to change the default permission modes of a file (for example 777) that you will be creating. The **umask** command can be placed in your *.profile* or the system default *profile*. The */etc/profile* is the system default profile. The **umask** command is specified as the following:

```
umask 000
```

The three octal digits (000) refer to read, write, and execute permission for owner, group, and other. The value of each permission digit is subtracted from the corresponding permissions digit. For example, if you entered **umask 022** in your system default */etc/profile,* a file normally created with 777 permission will be created with 755, and a file created with 666 will be created with 644. This is a way of ensuring that only you can write the file. See the **umask**(1) manual page in the *User's/System Administrator's Reference Manual* for additional information.

Password Administration

Whether or not users should have passwords is up to the administrator. For security reasons, it's a good idea to have a password. When a login is first established, a password can be assigned, or one can be assigned later. The person using the **root** login can assign or change a password for any login. See Appendix C in this document.

You can add, change, or delete your own password from the Change Password menu.

You can also change your password by using the **passwd** command. See the *User's/System Administrator's Reference Manual.* While in the shell, just type

```
$ passwd Enter
```

The system will prompt you for your old password, then the system will prompt you to enter the new password twice. If you are super–user, you will not be prompted for the old password. New passwords should be at least six characters long and must contain at least one numeric or special character. The space character is considered to be a special character. Other special characters include:

```
< > * ? ¦ & $ ; \ " ′ ′ ˆ ( ) [ ]
```

The Password Files

The file */etc/passwd* identifies each user to the system. Every time a login is established, a new entry is added to this file. Each entry is one line that has seven fields separated by colons. There are two password files on your UNIX system, */etc/password* and */etc/shadow*. The */etc/password* contains information for each user's login id, user id number, group id number, a comment on the user, the default program that is executed when the user logs in (usually **/bin/shell**, and the home directory. The */etc/shadow* file contains each user's encrypted password and password aging information.

All modifications to the *password* file should be done through the user interface or the **passwd** and **passmgmt** shell level commands. The *password* files should **never** be edited directly.

Password Aging

Password aging allows you to set time requirements on passwords. After a specified period of time, your password will expire, and you will be required to enter a new one. This forces you to change your password periodically. Provisions are made to prevent you from changing a new password before a specified time.

Normally, password aging is assigned by using the main interface menu. The System Administrator can assign password aging with the **root** login by using the **passwd**(1M) command in the *User's/System Administrator's Reference Manual.*

When a login is assigned, there is no aging. Password aging has to be assigned for passwords to expire. The password aging information consists of the following:

- The duration of the password—how often the password must be changed.

  max is the duration of the valid password in days.
  See **passwd**(1) in the *User's/System Administrator's Reference Manual.*

  min is the minimum number of days before a change can
  be made to a new password.
  See **passwd**(1) in the *User's/System Administrator's Reference Manual.*

- The minimum time interval between password changes.
- The day when the password was last changed. You do not enter this information. The system automatically manages this information for each user.

When establishing the password aging information, there are two variables to keep in mind:

Use the administration interface to enable password aging.

Data Encryption—Commands and Descriptions

If you have sensitive data that requires greater protection than that provided by access permission, you can encrypt the data. The encrypted file can not be read without a password. If somebody tried to read the encrypted file without a password, it could not be understood. The computer would output information in such a strange way no one could understand it.

NOTE You will only have data encryption capabilities if you have installed the security package floppy disk called *Security Administration Package* containing data encryption. Refer to Chapter 2 on installing optional add–on packages.

There are seven different commands used in data encryption. A brief summary of these commands appears in the table on the next page.

COMMAND LINE	DESCRIPTION
crypt	This command is used to encode and decode files. The **crypt** commands reads from the standard input or keyboard, and writes to the standard output or the terminal.
makekey	This command is used by the system to generate an encryption key.
ed -*x*	This command line is used to edit a file that has already been encrypted or to create a new encrypted file using the **ed** editor.
vi -*x*	This command line is used to edit a file that has already been encrypted or to create a new encrypted file using the **vi** editor.
ex -*x*	This command line is used to edit a file that has already been encrypted or to create a new encrypted file using the **ex** editor.
edit -*x*	This command line is used to edit a file that has already been encrypted or to create a new encrypted file using the **edit** editor.
X	This command is used to encrypt a file while in the (**ed**, **ex**, or **edit**) editor mode.

Crypt—Encode/Decode Files

The **crypt** command is used to encode and decode files for security. When using **crypt**, you have to assign a password (key) to encode the file. The same password is used to decode the file. An encrypted file cannot be read unless the correct password is used to decode it.

If no password is given with the **crypt** command, the system will prompt you for one. For security, the screen does not display the password as you type it in.

Password security is the most vulnerable part of the **crypt** command. Anyone who figures out your password can look at your files. The best way to ensure your security is to select an uncommon group of characters. As with your login password, the password should be no more than eight letters or numbers long.

A file can be encrypted in the shell mode using **crypt** or in the edit mode using the *-x* or X option. When you are ready to decrypt the file, you can use the **crypt** command in the shell mode. The following is the command format to encrypt a file:

crypt < *oldfile* > *newfile* Enter

Before removing the unencrypted *oldfile*, make sure the encrypted *newfile* can be decrypted using the appropriate password. The *oldfile* is the file to be encrypted. The *newfile* is the name of the destination file for the encrypted text. The *oldfile* should now be removed. The system will prompt you for a password.

NOTE

Always remember to remove the file (*oldfile*) you're encrypting from, because it will not be encrypted. Only the *newfile* will be encrypted.

Without any arguments, the crypt command takes standard input from the keyboard and encodes it before directing it to the standard output (the display). To encode an existing file, you must tell crypt to take its input (<) from a file instead of the keyboard. Similarly, you must tell crypt to send its output (>) to a new file instead of the display.

To decrypt a file, redirect the crypted file to a new file you can read. The command to decrypt a file is as follows:

crypt < *crypted_file* > *new_filename* Enter

NOTE

Always encrypt and decrypt files separately.

Encrypting and Decrypting With Editors

The editors (**ed**, **edit**, **ex**, or **vi**) can be used to either edit an existing file that has been encrypted or to create a new encrypted file by using the *-x* option. When encrypting a file, you have to assign a password to encode the file. The same password is used to decode the file. An encrypted file cannot be read unless the correct password is used to decode it.

Select an uncommon group of characters for the password. It should be no more than eight characters long.

The following is the command format for the editors (**ed**, **edit**, **ex**, or **vi**) using the *-x* option:

```
$ ed [-x] [filename] Enter

$ edit [-x] [filename] Enter

$ ex [-x] [filename] Enter

$ vi [-x] [filename] Enter
```

The *-x* option is used to either edit an existing file that has been encrypted or to create a new encrypted file. The *filename* argument is the name of the file that is being created or edited. The system will prompt you for a password.

When you get ready to decrypt the file, you must use the **crypt** command from the shell.

The editor **X** command is another way to encrypt a file while in the editor mode. The **X** command will only work with the **ed**, **edit**, or **ex** editors. (For the **vi** editor, type: **X**.) This command also needs a password to encrypt and decrypt files.

After you have edited the file, you can easily encrypt it again by using the **X** command as follows:

1. While still in the editor, enter **X** on a line by itself.
2. The system will prompt you for a password.
3. Quit the file.

Helpful Hints

When working with a single-user computer or in a multiuser environment, there are a few things to keep in mind about system security. Here are some suggestions that may help you:

- Check your files and directories to be sure the permission modes are the way you want them. Set the permission modes to allow only the necessary permissions for owner, group, and others.
- Encrypt sensitive files. The **crypt** command can be used with different editors. Be sure to remember the passwords to your encrypted files.
- Assign passwords to all logins. Change your login password regularly. Don't make your passwords obvious; use a combination of numbers and letters instead of standard names.
- Remove or lock logins that are not needed (to keep others from using them). The **passwd -l <user login>** command can be used to do this. See **passwd**(1) in the *User's/System Administrator's Reference Manual.*
- Any system with dial-up ports is less secure than one without dial–up ports. If your system has dial-up ports, you should turn on the **login logging** feature as described later in this section. The feature allows you to monitor unsuccessful login attempts.
- Check the */usr/adm/sulog* file to monitor the use of the **su** command. The **su** command (changing to another login) can be dangerous since the knowledge of another login/password is required by the user. When more users know the login and password, the information in the system

becomes less secure. Because of this, a log is kept on the use of the **su** command. Always enter the complete **/bin/su** path name when changing to another login.

- Check the */usr/adm/loginlog* file to monitor login attempts that have failed more than five times. By default, the feature login logging is not turned on. To start it, enter:

  ```
  # >/usr/adm/loginlog
  Enter
  ```

  This file (*/usr/adm/loginlog*) grows forever. Since it is not automatically cleared up, you must manually clear it or write a shell script that is called by **cron** to do it for you.

- Be sure any files you have in */bin, /usr/bin,* or */etc* directories are write secured.
- Log off the system if you're going to be away from the terminal. Don't leave a logged in terminal unattended, especially if you're logged in as **root**.
- Keep your computer in an area that can be secured when you're not using it.
- Make backup copies of your files on Floppy Disks. Store your floppies in a safe place. If your files on the computer should be destroyed, you will always have a backup copy. See " Back Up To Removable Media" in Chapter 4, "System Administration", for more information.
- Keep your boot diskette in a secure place (under lock and key).

Configuring Your Computer With Additional Terminals

Your computer can support multiple users. A terminal can be connected to the built-in serial port on the back of the computer. This allows two people to use the computer at one time. Of course, more than two people can have logins on the computer.

When the computer is powered up or rebooted, it will automatically be initialized to support both the console and the remote terminal if the serial port is appropriately set up as described in "Peripheral Setup", Chapter 4, "System Administration".

Requirements for Multiuser Operation

Follow the procedures in "Peripheral Setups" in Chapter 4, "System Administration", for administering the serial port. The following are requirements for the computer to support a remote terminal.

- Set the shell variable TERM to match the type of terminal attached to your computer. This will ensure correct operation of screen-oriented applications, such as **vi**. The default profile (*/etc/profile*) automatically sets TERM to AT386, corresponding to the computer console. At each remote terminal, you should set TERM as follows:

  ```
  TERM=terminal_name
  export TERM
  ```

  If each terminal is used frequently, you might want to set up the TERM command line in your *.profile*. If you frequently use several different types of remote terminals, you may want to modify your *.profile* to handle this.

- The terminal must have the correct baud rate set.
- The remote terminal must have a null modem cable.

Stopping a Command From a Remote Terminal

By default at login, the DEL character is the interrupt character. You should use the Del key on the remote terminal to stop execution of a command or program.

Installing Software Support for Additional Terminals

Part of the software included in the Foundation Set is to define the terminals you can add to your computer. This group of terminal characteristics is known as the Terminal Information Library (terminfo). Each entry in the library represents one supported terminal.

The following are the terminals the computer will support.

AT&T Terminals:	**TERM name:**
Personal Terminal (510)	510
BCT 513	513
UNIX PC	unix_pc
6386WGS	AT386
4410/5410.	4410

Other Terminals:	
ANSI X3.64	ansi
HP 262 series	2621
DEC VT 100.	vt100

If you're adding a terminal to your computer that is not on this list and you want the software support, you must add the additional terminal information. These entries are packaged separately and called the Software Installation/Remote Terminal Package. The Remote Terminal Package is used to install the "terminfo" data base. The Remote Terminal Package installation has a separate procedure that involves selecting a terminal type from a list. Also refer to the "Peripherals Setup" section in Chapter 4, "System Administration", for information on enabling the serial port.

Tunable System Parameters

The tunable parameter files contain kernel tunable parameters. These files are */etc/conf/cf.d/mtune* and */etc/conf/cf.d/stune*. These files can have a profound effect on system performance, and occasionally an add–on driver or kernel software module may have to modify an existing parameter or define a new tunable parameter which is accessible by other add–on drivers.

The *User's/System Administrator's Reference Manual* contains manual pages for *mtune* and *stune*. The *mtune* manual page defines a default value along with a minimum and maximum value for each kernel parameter. An add–on package should never modify a predefined system parameter in the *mtune* file.

Before proceeding, it may be useful to become familiar with the following manual pages for the ID/TP commands relating to tunable parameters.

- **idbuild**(1M) - A shell script that does the complete system reconfiguration
- **idtune**(1M) - A shell script to specify system tunable parameters
- **idspace**(1M) - A command to interrogate free space in one or more file systems
- **mtune**(4) - Tunable parameter master file
- **stune**(4) - Tunable parameter system file.

The remainder of this section describes procedures for modifying system tunable parameters for the computer. Adjusting these parameters can have a significant impact on system performance, and certain tunable parameters are normally adjusted upward when additional memory is installed to allow the system to support more users. For a computer used as a high-powered personal computer or dedicated processor, however, it may not be necessary to increase kernel tunable parameters when additional memory is installed. In fact, tuning certain parameters normally associated with adding additional memory to support more users (NBUF, NCLIST, etc.) can actually decrease overall performance since these parameters increase kernel data space requirements thus making less of the new memory available for user processes. Simply stated, the intended use of your computer and your observations on how well it is performing should be used as a guide in determining the need to adjust tunable parameters.

Your computer runs the System V Release 3.1 UNIX system. The UNIX system uses a Installable Drivers/Tunable Parameter (ID/TP) scheme ported from the PC 6300 PLUS.

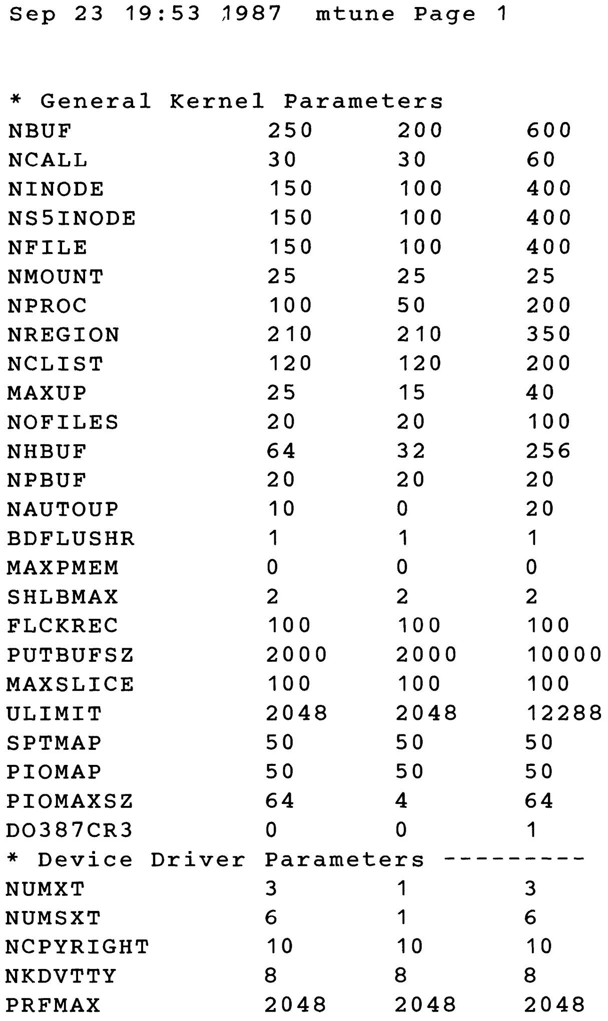
Additional information can be found in the Integrated Software Development Guide for the UNIX system.

In addition to the information provided here there are other aspects of system configuration that can have a dramatic affect on system performance as well. For example:

- Hard Disk Interleave
- File System Organization
- Use of Text-Bits (Sticky Bits)
- Directory Organization
- User $PATH Efficiency
- Use of Larger File System Block Sizes (2K File System)
- Use of "**ps**", "**sar**", process accounting, kernel profiling and other system utilities to determine system utilization.

Tunable system parameters are used to set various table sizes and system thresholds to handle the expected system load. For the most part, the initial tunable parameter values for your new computer are acceptable for most configurations and applications. If your application has special performance needs, you may have to experiment with different combinations of parameter values to find an optimal set. In order to modify kernel parameters, the UNIX system kernel will have to be reconfigured, and the system rebooted.

The computer has an Installable (Device) Driver/Tunable Parameter (ID/TP) scheme which places all system tunable parameters in a file *mtune* in the kernel configuration directory */etc/conf/cf.d.* The format of *mtune* is defined in the **mtune**(4) manual page. A sample *mtune* file delivered with the initial UNIX system is shown in Figure 5-1.



```
Sep 23 19:53 1987  mtune Page 1


* General Kernel Parameters
NBUF             250      200      600
NCALL            30       30       60
NINODE           150      100      400
NS5INODE         150      100      400
NFILE            150      100      400
NMOUNT           25       25       25
NPROC            100      50       200
NREGION          210      210      350
NCLIST           120      120      200
MAXUP            25       15       40
NOFILES          20       20       100
NHBUF            64       32       256
NPBUF            20       20       20
NAUTOUP          10       0        20
BDFLUSHR         1        1        1
MAXPMEM          0        0        0
SHLBMAX          2        2        2
FLCKREC          100      100      100
PUTBUFSZ         2000     2000     10000
MAXSLICE         100      100      100
ULIMIT           2048     2048     12288
SPTMAP           50       50       50
PIOMAP           50       50       50
PIOMAXSZ         64       4        64
DO387CR3         0        0        1
* Device Driver  Parameters ---------
NUMXT            3        1        3
NUMSXT           6        1        6
NCPYRIGHT        10       10       10
NKDVTTY          8        8        8
PRFMAX           2048     2048     2048
```

Figure 5-1: Sample *mtune* File (Sheet 1 of 4)

```
* Paging Parameters ----------------
VHNDFRAC       16     8     32
VHANDR          1     1     300
VHANDL         10     10     10
GPGSLO         25     0     25
GPGSHI         40     1     40
GPGSMSK        0x00000420     0x00000420     0x00000420
MAXSC          1      1      1
MAXFC          1      1      1
MAXUMEM        2560    2560    8192
MINARMEM       25      25      40
MINASMEM       25      25      40
BMAPFLMIN      80      40      1024
AUTOBMAPFL      30      0      300
BMAPFLRATE     3      1      20
* STREAMS Parameters ---------------
NQUEUE         96     0     384
NSTREAM        32     0      64
NSTRPUSH        9     9     9
NSTREVENT      256     256     512
MAXSEPGCNT      1      0      2
NMUXLINK       87      0      87
STRMSGSZ       4096    4096    4096
STRCTLSZ       1024    1024    1024
NBLK4096       0      0      0
NBLK2048       20      0      60
NBLK1024       20      0      32
NBLK512        8      0      18
```

Figure 5-1: Sample *mtune* File (Sheet 2 of 4)

```
Sep 23 19:53 1987  mtune Page 2


NBLK256        8     0     48
NBLK128        8     0     128
NBLK64         40    0     256
NBLK16         40    0     256
NBLK4          40    0     256
STRLOFRAC      80    0     80
STRMEDFRAC     90    80    100
NLOG           3     3     3
NUMSP          32    5     50
NNTTY          10    0     20
NLD            10    10    10
NUMTIM         16    0     32
NUMTRW         16    0     32
* Message Parameters ---------------
MSGMAP         100    10     200
MSGMAX         2048   512    8192
MSGMNB         4096   4096   4096
MSGMNI         50     50     50
MSGSSZ         8      8      8
MSGTQL         40     40     40
MSGSEG         1024   1024   1024
* Semaphore Parameters -------------
SEMMAP         10     10     10
SEMMNI         10     10     10
SEMMNS         60     60     60
SEMMNU         30     30     30
SEMMSL         25     25     25
SEMOPM         10     10     10
SEMUME         10     10     10
SEMVMX         32767  32767  32767
SEMAEM         16384  16384  16384
```

Figure 5-1: Sample *mtune* File (Sheet 3 of 4)

```
* Shared Memory Parameters ----------
SHMMAX        524288 131072 524288
SHMMIN        1      1      1
SHMMNI        100    100    100
SHMSEG        6      6      15
SHMALL        512    256    512
* RFS Parameters ---------------------
NRCVD         150    40     500
NSNDD         100    100    350
NSRMOUNT       10     0      50
NADVERTISE     25     0      25
MAXGDP        24     10     32
MINSERVE      3      3      3
MAXSERVE      6      3      6
NRDUSER       250    0      700
RFHEAP        3072   1024   3072
NLOCAL        0      0      0
NREMOTE       0      0      0
RCACHETIME     10     -1     10
RFS_VHIGH     1      1      1
RFS_VLOW      1      1      1
* STARLAN Parameters ---------------
N_SRMPORTS     3      3      3
N_DRIVERS     20     0      20
N_CIRCUITS    20     0      20
* S52K (2K file system) Parameters --
S52KNBUF      100    100    400
S52KNHBUF      32     32     256
```

Figure 5-1: Sample *mtune* File (Sheet 4 of 4)

As noted in the **mtune** manual page, each system tunable parameter is assigned a default value along with a minimum and maximum value. You can examine *mtune* to determine the tunable parameter settings for your computer, however, you should never modify the *mtune* file. A second file in the configuration directory *stune* is used to modify a parameter to any value between the minimum and maximum value defined in the *mtune* file. See Section "Reconfiguring the Kernel to Enable New Parameters" for step-by-step procedures to use when modifying system tunable parameters.

When to Tune and What to Tune

Some UNIX system running on superminicomputers or mainframe computers support dozens of users simultaneously. As additional users are added to those systems often additional memory hardware is added, and system parameters are adjusted to allow the UNIX system kernel to operate more efficiently. That often includes allocating more memory to kernel data space by increasing the size of kernel data structures. This generally will allow the system to support more users. However, as these kernel data structures are increased in size, it takes more time for the kernel to scan those structures, and in fact, increasing certain parameters unnecessarily can actually slow the system down. For example, increasing the parameter NPROC will allow the system to maintain a larger list (the proc table) of active processes. This can have an adverse affect on the kernel scheduler since it now must repeatedly scan this larger table every time it is check to see which process to run next. Additionally, since the kernel data space requirements will increase when table sizes are increased, there will be less memory space available for user processes which can also lower overall performance.

Special Case Needs

Often your system usage will present you with the need to tune certain parameters for particular circumstances. A common need is the ability to create very large files. This can be accomplished by becoming superuser, and modifying the 'ulimit' for the particular shell process that you are running as superuser. An alternate solution is to modify the system ULIMIT for all users. The ULIMIT parameter and other commonly encountered limits are summarized in Figure 5-2.

Desired Improvement	Parameters
Improve System Performance(*) when additional memory installed.	NBUF, NHBUF
Other Performance related system parameters.	NAUTOUP, MAXSLICE, BDFLUSHR (also see paging parameters).
Increase system limits when additional memory installed (support more users; reduce chances of system problems at times of heavy load, etc.).	NCALL, NINODE, NSINODE, NFILE, NPROC, NREGIONS, NCLIST (Also see message, semaphore, and shared memory parameters).
Users need to create bigger files.	ULIMIT
Each user needs to open more files.	NOFILES
Each user needs to run more processes.	MAXUP
Other system limits that may be encountered.	SHLBMAX, FLCKREC, SPTMAP, NUMXT, NUMSXT, PRFMAX (also see STREAMS, and RFS parameters.)
Miscellaneous	PUTBUFSIZE, DO387CR3

(*) Note that increasing the size of the buffer cache will increase the chances that data frequently read will be found in memory rather than have to read it from disk. Depending on your system usage, an increase in the chance of reusing a data block may not yield an overall system performance improvement. In some system usage scenarios, it can provide a significant performance improvement. See Section "Buffer Cache".

Figure 5-2: Special Case Tuning Needs.

Kernel Messages That System Limits Are Being Exceeded

There are cases when the UNIX system kernel will advise you that system limits are being exceeded. These messages are in the form of console printouts. Some of the messages are advisory only, others precede a system panic in which case additional diagnostic messages are printed, and the system will "hang" requiring you to reboot. If you encounter any of the messages listed in Figure 5-3, refer to the appropriate tunable parameter for additional information.

Kernel Console Message	Parameter
iget - inode table overflow	NINODE
Timeout table overflow	NCALL
File table overflow	NFILE
mfree map overflow *n*(*)	SPTMAP
Region table overflow	NREGION
Configured value of NOFILES *n*(*) is less than minimum (greater than maximum).	NOFILES
stropen: out of streams	NSTREAM
swapdel - too few free pages	MINASMEM
stropen: out of streams	NSTREAM
stropen: out of queues	NQUEUE

(*) The value *n* indicates the actual value encountered by the kernel.

Figure 5-3: Kernel Messages and Associated Tunable Parameter

Buffer Cache

The NBUF parameter specifies the number of 1K buffers in the system buffer cache. These buffers hold recently-used data on the chance that it will be needed again. If a read or a write can be satisfied using the buffer cache instead of the disk, system performance improves, because memory operations are much faster than disk operations. NHBUF specifies the number of hashing buckets in the buffer cache. The more buffers, the greater chance that data can be found in the buffers without the system having to do a time-consuming disk read. The read and write cache hit ratios listed by the **sar -b** command indicate how effective the system buffers are. The value for NHBUF is a power of 2 that is roughly one-quarter the value of NBUF.

The values of NBUF and NHBUF given in the *mtune* file are a good starting point for the buffer cache. These values come close to optimum for most system workloads. Increasing NBUF and NHBUF, up to a point, may improve system performance. A system with 2 megabytes of memory can typically devote roughly 250K bytes of memory to buffers while a system with 4 megabytes of memory can devote 400K bytes of memory to buffers. However, if too many buffers are allocated, there may not be enough memory space for efficient operation of user processes, and the amount of swapping done by the system will increase. The swapping activity usually costs more in system efficiency than is gained by having a large amount of buffer space. If the **sar -w** command shows that your system has **swpot/s** greater than 1.0, adding buffers may not be beneficial. Additionally, by increasing the number of buffers the kernel must manage, the kernel routines dealing with managing the allocation and freeing of buffers may take longer to execute.

If you choose to modify the number of buffers, after the UNIX system has run for a day or so, check system performance particularly excessive swapping activity. If such activity is found, reduce the number of buffers.

The parameters S52KNBUF and S52KNHBUF perform analogous functions for 2K buffers. When you increase 2K buffers, you may want to lower the number of 1K buffers, so that you maintain the memory available for user processes.

What to Do When You Add More Memory

In the past, most UNIX system administrators would routinely increase all system tunable parameters when additional memory was installed in mini and supermini computers. This would usually allow the UNIX system to support more users without encountering system limits during heavy system activity. For a single-user PC environment, however, there may be no need to increase kernel tunable parameters at all. And, for the reasons previously stated, keeping system limits at the default value may deliver optimum performance even when additional memory is installed.

As Figure 5-4 ́shows, the default parameters defined in the base system *mtune* file are intended for a 2-megabyte memory system. This is the minimum memory size recommended for a computer running the UNIX operating system. If your system will be used in a multiuser configuration, that is, five users or more, you may wish to add more memory and increase selective parameters to be sure system limits are not reached, and to increase the buffer cache hit ratio. The values for selective parameters are given for a 3-megabyte and 4-megabyte memory configuration. You should try to establish a performance baseline before making these changes, modify your system parameters, and again determine your system's performance. This is the best approach to see if parameter changes increase or decrease your systems performance.

	Memory Size		
Parameter	2 Meg	3 Meg	4 Meg
NBUF	250	300	400
NHBUF	64	64	128
NCALL	30	40	50
NINODE	150	200	300
NS5INODE	150	200	300
NFILE	150	200	300
NREGION	210	250	300
NCLIST	120	140	170
NPROC	100	120	150

Figure 5-4: Suggested Parameter Values Based on Memory Size

Parameter Descriptions

The following sections provide a breakdown of system tunable parameters defined in the file */etc/conf/cf.d/mtune*. The parameter categories are as follows:

General Kernel Parameters
Device Driver Parameters

Paging Parameters
Streams Parameters
Message Parameters
Semaphore Parameters
Shared Memory Parameters
RFS Parameters
STARLAN Parameters
S52K (2K File System) Parameters

Note that the Streams parameters determine configuration of the Network Support Utilities (NSU) add-on package, the RFS Parameters determine configuration of the Remote File Sharing add-on, the STARLAN Parameters control configuration of the STARLAN add-on, and the S52K (2K File System) Parameters control configuration of the S52K add-on. If these packages are not installed, adjusting the parameter values will have no effect upon you system's configuration.

General Kernel Parameters

NBUF

Specifies how many 1K system buffers to allocate. The UNIX system buffers form a data cache. The data cache is a memory array containing disk file information. Cache hit rate increases with the number of buffers. Cache hits reduce the number of disk accesses and thus may improve overall performance. The entries are normally in the range of 100 to 600. Each buffer contains 1076 bytes. 1K hash buffers (NHBUF) should be increased along with system buffers (NBUF) for optimal performance.

NCALL

Specifies how many call-out table entries to allocate. Each entry represents a function to be invoked at a later time by the clock handler portion of the kernel. This value must be greater than 2 and is normally in the range of 10 to 70. The default value is 30. Each entry contains 16 bytes.

Software drivers may use call entries to check hardware device status. When the call-out table overflows, the system crashes and outputs the following message on the system console:

```
PANIC: Timeout table overflow
```

NINODE

Specifies how many i-node table entries to allocate. Each table entry represents an in-core i-node that is an active file. For example, an active file might be a current directory, an open file, or a mount point. The file control structure is modified when changing this variable. The number of entries used depends on the number of opened files. The entries are normally in the range of 100 to 400. The value for NINODE pertains directly to the NFILE value. (NINODE is equal to or greater than NFILE). NINODE must always be less than or equal to NS5INODE. NINODE greater than NS5INODE results in an unusable system. When the i-node table overflows, the following warning message is output on the system console:

```
WARNING: i-node table overflow
```

NS5INODE

NS5INODE must always be equal to or greater than NINODE.

NFILE

Specifies how many open file table entries to allocate. Each entry represents an open file. The entry is normally in the range of 100 to 400. Each entry contains 12 bytes. The NFILE entry relates directly to the NINODE entry. (NFILE is less than or equal to NINODE). The NFILE control structure operates in the same manner as the NINODE structure. When the file table overflows, the following warning message is output on the system console.

```
NOTICE: file table overflow
```

As a reminder, this parameter does not affect the number of open files per process (see the NOFILES parameter).

NMOUNT

Specifies how many mount table entries to allocate. Each entry represents a mounted file system. The root (/) file system is always the first entry. When full, the **mount**(2) system call returns the error EBUSY. Since the mount table is searched linearly, this value should be as low as possible.

NPROC

Specifies how many process table entries to allocate. Each table entry represents an active process. The swapper is always the first entry and **/etc/init** is always the second entry. The number of entries depends on the number of terminal lines available and the number of processes spawned by each user. The average number of processes per user is in the range of 2 to 5 (also see MAXUP, default value 25). When full, the **fork**(2) system call returns the error EAGAIN. The NPROC entry is in the range of 50 to 200.

NREGION

Specifies how many region table entries to allocate. Each NREGION entry contains 36 bytes. Most processes have 3 regions: text, data, and stack. Additional regions are needed for each shared memory segment and shared library (text and data) attached. However, the region table entry for the text of a "shared text" program will be shared by all processes executing that program. Each shared memory segment attached to one or more processes uses another region table entry. A good starting value for this parameter is about 3.5 times NPROC. If the system runs out of region table entries, the following message is output on the system console.

```
Region table overflow
```

NCLIST

Specifies how many character list buffers to allocate. Each buffer contains up to 64 bytes. The buffers are dynamically linked to form input and output queues for the terminal lines and other slow speed devices. The average number of buffers needed per terminal is in the range of 5 to 10. Each entry (buffer space plus header) contains 72 bytes. When full, input and output characters dealing with terminals are lost, although echoing continues.

MAXUP

Specifies how many concurrent processes a non-superuser is allowed to run. The entry is normally in the range of 15 to 40. This value should not exceed the value of NPROC (NPROC should be at least 10% more than MAXUP). This value is per user identification number, not per terminal. For example, if 12 people are logged in on the same user identification, the default limit would be reached very quickly.

NOFILES

Specifies the maximum number of open files per process. Default is 20. Values higher than 20 are accessible only to processes using system calls (**open**(2) and **creat**(2), for example). Processes using standard Input/Output (I/O) subroutines are limited to 20, independent of the value of NOFILES. Unless an application package recommends that NOFILES be changed, the default setting of 20 should be left as is.

/bin/sh uses 3 file table entries: standard input, standard output, and standard error (0, 1, and 2 are normally reserved for stdin, stdout, and stderr, respectively). This leaves the value of NOFILES minus 3 as the number of other open files available per process. If a process requires up to three more than this number, then the standard files must be closed. This practice is NOT recommended, and must be used with caution, if at all.

If the configured value of NOFILES is greater than the maximum (100) or less than the minimum (20), the configured value is set to the default (20), and a NOTICE message is sent to the console.

NHBUF

Specifies how many "hash buckets" to allocate for 1K buffers. These are used to search for a buffer given a device number and block number rather than a linear search through the entire list of buffers. **This value must be a power of 2**. Each entry contains 12 bytes. The NHBUF value must be chosen so that the value NBUF divided by NHBUF is approximately equal to 4.

NPBUF

Specifies how many physical I/O buffers to allocate. One I/O buffer is needed for each physical read or write active. Each entry contains 52 bytes. The default value is 20.

NAUTOUP

The NAUTOUP entry specifies the buffer age in seconds for automatic file system updates. A system buffer is written to the hard disk when it has been memory-resident for the interval specified by the NAUTOUP parameter. Specifying a smaller limit increases system reliability by writing the buffers to disk more frequently and decreases system performance. Specifying a larger limit increases system performance at the expense of reliability. This parameter controls behavior of the **bdflush** daemon process.

BDFLUSHR
Specifies the rate in seconds for checking the need to write the file system buffers to disk. The default is 1 second. This parameter controls behavior of the **bdflush** daemon process.

MAXPMEM
Specifies the maximum amount of physical memory to use in pages. The default value of 0 specifies that all available physical memory be used.

SHLBMAX
Specifies the maximum number of shared libraries that can be attached to a process at one time.

FLCKREC
Specifies the number of records that can be locked by the system. The default value is 100. Each entry contains 28 bytes.

PUTBUFSZ
Specifies the size of a circular buffer, **putbuf**, that is used to contain a copy of the last PUTBUFSZ characters written to the console by the operating system. The contents of **putbuf** can be viewed using **crash**(1M).

MAXSLICE
Specifies in clock ticks the maximum time slice for user processes. After a process executes for its allocated time slice, that process is suspended. The operating system then dispatches the highest priority process and allocates to it MAXSLICE clock ticks. MAXSLICE, is normally one second (100 clock ticks on the WGS 6836).

ULIMIT
Specifies in 512-byte blocks the size of the largest file that an ordinary user may write. The default value is 2048; that is, the largest file an ordinary user may write is one megabyte. The super-user may write a file as large as the file system can hold. The ULIMIT parameter does not apply to reads: any user may read a file of any size.

SPTMAP Determines the size of the map entry array used for managing kernel virtual address space. Users should not modify this parameter.

PIOMAP Determines the size of the map entry array used by the kernel programmed I/O (PIO) breakup routine. This routine allows device drivers to do programmed I/O of large data blocks at interrupt level by breaking the data blocks into smaller data units. Users should not modify this parameter.

PIOMAXSZ Maximum number of pages to use at one time for programmed I/O. Users should not modify this parameter.

DO387CR3 Controls setting of high order bits of Control Register 3 (CR3) when an 80387 chip is installed.

Device Driver Parameters

The following parameters control various data structure sizes and other limits in base system device drivers.

NUMXT Determines number of xt layers sub devices configured to support bitmapped display devices such as the BLIT or the AT&T 5620 terminal.

NUMSXT Determines number of shell layers sub devices configured.

NCPYRIGHT Defines the size of a kernel data structure used to print console initialization messages. Users should not modify this parameter.

NKDVTTY Determines the number of virtual terminals (ttys) supported by the console keyboard driver. Users should not modify this parameter.

PRFMAX Maximum number of text symbols that the kernel profiler (/dev/prf) will be able to properly process.

Paging Parameters

There exists in the system a paging daemon, **vhand**, whose responsibility is to free up memory as the need arises. It uses a "least recently used" algorithm to approximate process working sets, and it writes those pages out to disk that have not been touched during some period of time. The page size is

2048 bytes. When memory is exceptionally tight, the working sets of entire processes may be swapped out. A second daemon **bmapflush** controls the rate at which block maps are freed.

The following tunable parameters determine how often **vhand** and **bmapflush** run and under what conditions. The default values should be adequate for most applications.

VHNDFRAC Used to determine the initial value for the system variable VHANDL. VHANDL is set to the maximum user-available memory divided by VHNDFRAC or the value of GPGSHI, whichever is larger. The value of VHANDL determines when the paging daemon **vhand** runs. The amount of available free memory is compared with the value of VHANDL every VHANDR seconds. If free memory is less than VHANDL, then the paging daemon **vhand** is awakened.

The default for VHNDFRAC is 16. Decrease the value to make the daemon more active; increase the value to make the daemon less active (must be > 0 and < 25 percent of available memory).

VHANDR Specifies in seconds the maximum rate at which **vhand** can run. **vhand** will only run at this rate if free memory is less than VHANDL, as explained above for VHNDFRAC. The default is 1. Increase the value to make the daemon less active (must be an integer > 0 and ≤ 300). If you have set the value higher, decreasing it makes the daemon more active.

VHANDL See VHNDFRAC above.

GPGSLO Specifies the low water mark of free memory in pages for **vhand** to start stealing pages from processes. The default is 25. Increase the value to make the daemon more active; decrease the value to make the daemon less active (must be an integer ≥ 0 and < GPGSHI).

GPGSHI Specifies the high water mark of free memory in pages for **vhand** to stop stealing pages from processes. The default is 40. Increase the value to make the daemon more active; decrease the value to make the daemon less active

(The value must be an integer > 0, > GPGSLO and < 25 percent of the number of pages of available memory).

GPGSMSK — Mask used by the paging daemon The default is 0x00000420. This value should not be changed.

MAXSC — Specifies the maximum number of pages which will be swapped out in a single operation. The default value is 1.

MAXFC — Specifies the maximum number of pages that will be added to the free list in a single operation. The default value is 1.

MAXUMEM — Specifies the maximum size of a user's virtual address space in pages. This value cannot be greater than 8192. The default is 2560.

MINARMEM — Specifies the minimum number of memory pages reserved for the text and data segments of user processes.

MINASMEM — Threshold value that specifies the number of memory and swap pages reserved for system purposes (unavailable for the text and data segments of user processes).

BMAPFLMIN — Determines available memory threshold at which the block map flush daemon **(bmapflush)** starts discarding block map lists for reuse. Users should not modify this parameter.

AUTOBMAPFL — Determines age at which block map lists are discarded by **bmapflush**. Default is 30 seconds.

BMAPFLRATE — Number of seconds between calls to the block map flush daemon. Default is 3 seconds.

Streams Parameters

The following tunable parameters are associated with Streams processing. The values have no affect on the system unless the Network Support Utilities (NSU) package is installed.

NQUEUE — The number of Streams queues to be configured. Queues are always allocated in pairs, so this number should be even. A minimal Stream contains four queues (two for the Stream head, two for the driver). Each module pushed on a Stream requires an additional two queues. A typical configuration value is 4*NSTREAM.

NSTREAM The number of "Stream-head" (stdata) structures to be configured. One is needed for each Stream opened, including both Streams currently open from user processes and Streams linked under multiplexers. The recommended configuration value is highly application-dependent, but a value of 32-40 usually suffices on a computer for running a single transport provider with moderate traffic.

NSTRPUSH The maximum number of modules that may be pushed onto a Stream. This is used to prevent an errant user process from consuming all of the available queues on a single Stream. By default this value is 9, but in practice, existing applications have pushed at most four modules on a Stream.

NSTREVENT The initial number of Stream event cells to be configured. Stream event cells are used for recording process-specific information in the **poll**(2) system call. They are also used in the implementation of the STREAMS I_SETSIG **ioctl** and in the kernel **bufcall()** mechanism. A rough minimum value to configure would be the expected number of processes to be simultaneously using **poll**(2) times the expected number of Streams being polled per process, plus the expected number of processes expected to be using Streams concurrently. The default is 256. Note that this number is not necessarily a hard upper limit on the number of event cells that will be available on the system (see MAXSEPGCNT).

MAXSEPGCNT The number of additional pages of memory that can be dynamically allocated for event cells. If this value is 0, only the allocation defined by NSTREVENT is available for use. If the value is not 0 and if the kernel runs out of event cells, it will under some circumstances attempt to allocate an extra page of memory from which new event cells can be created. MAXSEPGCNT places a limit on the number of pages that can be allocated for this purpose. Once a page has been allocated for event cells, however, it cannot be recovered later for use elsewhere. It is recommended that the NSTREVENT value be set to accommodate most load conditions, and that MAXSEPGCNT be set to 1 to handle exceptional load cases should they arise.

NMUXLINK The maximum number of multiplexer links to be configured. One link structure is required for each active multiplexor link (STREAMS I_LINK **ioctl**). This number is application dependent; the default allocation guarantees availability of links.

STRMSGSZ The maximum allowable size of the data portion of any Streams message. This should usually be set just large enough to accommodate the maximum packet size restrictions of the configured Streams modules. If it is larger than necessary, a single **write**(2) or **putmsg**(2) can consume an inordinate number of message blocks. The recommend value of 4096 is sufficient for existing applications.

STRCTLSZ The maximum allowable size of the control portion of any Streams message. The control portion of a **putmsg**(2) message is not subject to the constraints of the min/max packet size, so the value entered here is the only way of providing a limit for the control part of a message. The recommended value of 1024 is more than sufficient for existing applications.

NBLK*n* The number of Streams data blocks and buffers to be allocated for each size class are controlled by the parameters NBLK4 through NBLK4096. Message block headers are also allocated based on these numbers: the number of message blocks is 1.25 times the total of all data block allocations. This provides a message block for each data block, plus some extras for duplicating messages (kernel functions **dupb**(), **dupmsg**()). The optimal configuration depends on both the amount of primary memory available and the intended application. The default values provided in the NSU package are intended to support a moderately loaded configuration using Remote File Sharing (RFS) and UUCP/CU over STARLAN.

STRLOFRAC

The percentage of data blocks of a given class at which low-priority block allocation requests are automatically failed. For example, if STRLOFRAC is 40 and there are 48 256-byte blocks, a low-priority allocation request will fail when more than 19 256-byte blocks are already allocated. The parameter is used to help prevent deadlock situations by starving out low-priority activity. The recommended value of 40 works well for current applications. STRLOFRAC must always be in the range 0 <= STRLOFRAC <= STRMEDFRAC.

STRMEDFRAC

The percentage cutoff at which medium priority block allocations are failed (see STRLOFRAC discussion above). The recommended value of 90 works well for current applications. STRMEDFRAC must always be in the range STRLOFRAC <= STRMEDFRAC <= 100.

NOTE There is no cutoff fraction for high-priority allocation requests; it is effectively 100.

NLOG

The number of minor devices to be configured for the log driver; the active minor devices will be 0 through (NLOG–1). The recommended value of 3 services an error logger (**strerr**(1M)) and a trace command (**strace**(1M)), with one left over for miscellaneous usage. If only an error logger and a tracer are to be supported, this number can be set to 2. If there are several daemons for an application that may be submitting log messages, this number can be increased to accommodate the extra users.

NUMSP

Determines number of Streams Pipe devices (/dev/sp) supported by the system. Users should not modify this parameter.

NNTTY

Determines number of Streams tty devices supported. Users should not modify this parameter.

NLD Determines number of tty Line Discipline data structures to allocate in kernel data space for use by Streams tty devices. Users should not modify this parameter.

NUMTIM Maximum number of Streams modules that can be pushed by the Transport Library Interface (TLI). This value controls the number of data structures used to hold pushed streams modules configuration data. Users should not modify this parameter.

NUMTRW Number of Transport Library Interface (TLI) read/write data structures to allocate in kernel data space. Users should not modify this parameter.

Message Parameters

The following tunable parameters are associated with interprocess communication messages.

MSGMAP Specifies the size of the control map used to manage message segments. Default value is 100. Each entry contains 8 bytes.

MSGMAX Specifies the maximum size of a message. The default value is 2048. Although the maximum possible size the kernel can process is 64 kilobytes –1, the *mtune* limit is 8192.

MSGMNB Specifies the maximum length of a message queue. The default value is 4096.

MSGMNI Specifies the maximum number of message queues system-wide (id structure). The default value is 50.

MSGSSZ Specifies the size, in bytes, of a message segment. Messages consist of a contiguous set of message segments large enough to fit the text. The default value is 8. The value of MSGSSZ times the value of MSGSEG must be less than or equal to 131,072 bytes (128 kilobytes).

MSGTQL Specifies the number of message headers in the system and, thus, the number of outstanding messages. The default value is 40. Each entry contains 12 bytes.

MSGSEG Specifies the number of message segments in the system. The default value is 1024. The value of MSGSSZ times the value of MSGSEG must be less than or equal to 131,072 bytes (128 kilobytes).

Semaphore Parameters

The following tunable parameters are associated with interprocess communication semaphores.

SEMMAP Specifies the size of the control map used to manage semaphore sets. The default value is 10. Each entry contains 8 bytes.

SEMMNI Specifies the number of semaphore identifiers in the kernel. This is the number of unique semaphore sets that can be active at any given time. The default value is 10. Each entry contains 32 bytes.

SEMMNS Specifies the number of semaphores in the system. The default value is 60. Each entry contains 8 bytes.

SEMMNU Specifies the number of undo structures in the system. The default value is 30. The size is equal to 8 x (SEMUME + 2) bytes.

SEMMSL Specifies the maximum number of semaphores per semaphore identifier. The default value is 25.

SEMOPM Specifies the maximum number of semaphore operations that can be executed per **semop**(2) system call. The default value is 10. Each entry contains 8 bytes.

SEMUME Specifies the maximum number of undo entries per undo structure. The default value is 10. The size is equal to 8*(SEMMNU) bytes.

SEMVMX Specifies the maximum value a semaphore can have. The default value is 32767. The default value is the maximum value for this parameter.

SEMAEM Specifies the adjustment on exit for maximum value, alias **semadj**. This value is used when a semaphore value becomes greater than or equal to the absolute value of **semop**(2), unless the program has set its own value. The default value is 16384. The default value is the maximum value for this parameter.

Shared Memory Parameters

The following tunable parameters are associated with interprocess communication shared memory.

SHMMAX	Specifies the maximum shared memory segment size. The default value is 524288.
SHMMIN	Specifies the minimum shared memory segment size. The default value is 1.
SHMMNI	Specifies the maximum number of shared memory identifiers system wide. The default value is 100. Each entry contains 52 bytes.
SHMSEG	Specifies the number of attached shared memory segments per process. The default value is 6. The maximum value is 15.
SHMALL	Specifies the maximum number of in-use shared memory text segments. The default value is 512.

Remote File Sharing Parameters

There are several parameters you can tune to best suit the way you use Remote File Sharing. Remote File Sharing parameters control the amount of system resources you devote to Remote File Sharing service. Each network transport provider may also have some tunable parameters that may affect performance characteristics of that particular network. See the network documentation for your network for more details.

All parameters have set default values that should work well for an average system, however, if the values are too small, you may not be providing enough resources to properly handle your Remote File Sharing load. Requests for mounts, advertises, or even a file could fail if either of those values reach the maximum number allowed for your machine. If these parameters are too large, you could be allocating more system resources than you need to use.

Note that these parameters have no affect on your system unless the RFS add-on package is installed.

NRCVD (maximum number of receive descriptors)
Your system creates one receive descriptor for each file or directory being referenced by remote users and one for each process on your machine awaiting response to a remote request. If you limit the number of receive descriptors, you limit the number of local files and directories that can be accessed at a time by remote users. The result of exceeding the limit would be error messages for remote user commands.

NSNDD (maximum number of send descriptors)
For each remote resource (file or directory) your users reference, your system creates a send descriptor. A send descriptor is also allocated for each server process and each message waiting on the receive queue. You can change this value to limit how many remote files and directories your machine can access at a time. This would, in effect, limit the amount of Remote File Sharing activities your users can perform. The result of exceeding the limit would be error messages for user commands.

NSRMOUNT (server mount table entries)
Each time a remote machine mounts one of your resources, an entry is added to your server mount table. This number limits the total number of your resources that can be mounted at a time by remote machines.

NADVERTISE (advertise table)
An entry is placed in your advertise table for each resource you advertise. This parameter sets the maximum resources you can advertise.

MAXGDP (virtual circuits)
There are up to two connections (virtual circuits) set up on the network between you and each machine with which you are currently sharing resources. There is one for each computer whose resources you mount and one for each computer that mounts your resources. A virtual circuit is created when a computer first mounts a resource from another, and it is taken down when the last resource is unmounted.

This parameter limits the number of Remote File Sharing virtual circuits your computer can have open on the network at a time. It limits how many remote computers you can share resources with at a time. Note that a given network may have a limited number of circuits on any one computer, so this parameter influences the maximum percentage of those that might be used for Remote File Sharing.

MINSERVE (minimum server processes)

Your system uses server processes to handle remote requests for your resources. This parameter sets how many server processes are always active on your computer. (See the **sar -S** command for information on monitoring server processes.)

MAXSERVE (maximum server processes)

When there are more remote requests for your resources than can be handled by the minimum servers, your computer can temporarily create more. This parameter sets the maximum total server processes your system can have (MINSERVE plus the number it can dynamically create).

NRDUSER

This value specifies the number of receive descriptor **user** entries to allocate. Each entry represents a client machine's use of one of your files or directories. While there is one receive descriptor allocated for each file or directory being accessed remotely (NRCVD), there can be multiple receive descriptor **user** entries for each client using the file or directory (NRDUSER). These entries are used during recovery when the network or a client goes down. This value should be about one and one-half times the value of NRCVD.

RFHEAP

This value specifies the size in bytes of an area of memory set aside for RFS information. It contains the following information:

- The user and group ID mapping tables and the domain name of each machine currently sharing a resource(s) with your machine.

- A list of machine names supplied as a client list when you advertise resources.

The appropriate size for RFHEAP depends on:

□ UID/GID tables (size and number).

There will always be two global tables, one UID and one GID. Also, any machine with a **host** entry in **uid.rules** or **gid.rules** files will have a table corresponding to each of these entries while it is connected to this machine. Machines that do not have separate entries in one of these files do not take any extra space.

To estimate the size on an individual table, type **idload -n**. There will be one 4-byte table entry per line of output from **idload**, plus up to 24 bytes of overhead per table.

□ Adv client lists (size and number).

Each advertise may have a list of authorized clients attached to it. This list is stored in this area, with its size unchanged, until the resource is unadvertised.

□ Currently connected resources.

Each connection will use a maximum of 64 bytes to store the name of the connected resource. This memory is allocated dynamically, so some additional space is required to account for possible fragmentation as space is allocated and de-allocated. Since the total size is likely to be relatively small, 1 to 4 kilobytes, it is best to allow too much rather than too little space.

NLOCAL (local access buffers)

This parameter sets the minimum number of local buffers, available from the common buffer pool, reserved for local access. RFS client caching shares the common buffer pool with the local accesses (usually disk or tape). This value, therefore, protects local data from adverse effects of competition with RFS buffer use.

When this threshold is turned off (set to 0), it defaults to the recommended value of one third of the entire buffer pool (NBUF). A non-zero value of NLOCAL overrides this default.

Note that if RFS is not running or has had no recent activity, the entire buffer pool will be available to local access.

NREMOTE (remote access buffers)

This parameter sets the minimum number of local buffers, available from the common buffer pool, reserved for remote resource read data. When this threshold is turned off (set to 0), it defaults to the recommended value of one third of the entire buffer pool (NBUF). A non-zero value of NREMOTE overrides this default.

Note that the sum of NREMOTE and NLOCAL must not be greater than NBUF. If this condition is detected, a console warning message is printed and the default value (one third of NBUF) is used for both NREMOTE and NLOCAL.

RCACHETIME (caching time off)

This parameter can be used in two ways: 1) to turn off caching for your entire machine; 2) to define the number of seconds that network caching is turned off when a file is modified.

To turn off caching for your entire machine, the parameter must be set to -1.

The second use of RCACHETIME requires some explanation. When a write to a server file occurs, the server machine sends invalidation messages to all client machines that have the file open. The client machines remove data affected by the write from their caches. Caching of that file's data is not resumed until the writing processes close the file or until the seconds in this parameter have elapsed.

The assumption is that write traffic is "bursty" and that the first write may be closely followed by other writes. Turning off caching avoids the overhead of sending invalidation messages for subsequent writes.

RFS_VHIGH Highest RFS version number with which your machine will communicate.

RFS_VLOW Lowest RFS version number with which your machine will communicate.

In addition to the above, the NHBUF parameter has implications for RFS. The value of NHBUF is used to specify how many "hash buckets" to allocate for remote data in the buffer pool, as well as for local data. The hash buckets are used to search for a buffer given a remote server machine ID and file ID, rather than a linear search through the entire list of buffers. (See Sections "Buffer Cache" and "General Kernel Parameters" for further discussions of NHBUF.)

Figure 5-5 lists the key Remote File Sharing parameters and recommended values for different uses of Remote File Sharing. "Client Only" means that your machine will only be using remote resources, not sharing any from your own machine. "Server Only" means you will only offer your resources to other machines without mounting any remote resources. "Client+Server" means you will both offer local resources and use remote resources.

Parameter	Client Only	Server Only	Client+Server	Default Value	Size per Entry in Bytes
NSRMOUNT	0	50	50	24	
MAXGDP	10	24	24	24	104
NADVERTISE	0	25	25	25	32
NRCVD	40	300	150	150	48
NRDUSER	0	450	225	225	24
NSNDD	150	30	150	150	44
MINSERVE	0	3	3	3	9K
MAXSERVE	0	6	6	6	-
RFHEAP	2048	3072	3072	3072	1
NREMOTE	0	0	0	0	-
NLOCAL	0	0	0	0	-
RCACHETIME	10	10	10	10	-

Figure 5-5: RFS Tunable Parameter Settings

STARLAN Parameters

There are several parameters you can tune to best suit the way you use the STARLAN media driver and the Universal Receiver Protocol (URP) driver. All parameters have set default values that should work well for an average system.

Note that these parameters have no affect on your system unless the STARLAN add-on package is installed.

N_SRMPORTS Determines the number of supported Starlan administrative ports.

N_DRIVERS Maximum number of protocol drivers that can be multiplexed by the STARLAN URP protocol driver. The default is 20.

N_CIRCUITS Maximum number of virtual circuits that can be processed by the STARLAN URP protocol driver. The default is 20.

S52K (2K File System) Parameters

Note that these parameters have no affect on your system unless the S52K add-on package is installed.

S52KNBUF — Specifies how many 2K system buffers to allocate. This parameter performs the same function for 2K file systems that NBUF performs for 1K file systems. The entries are normally in the range of 100 to 400. Each buffer contains 2100 bytes. 2K hash buffers (S52KNHBUF) should be increased, along with (S52KNBUF), for optimal performance. If you configure 2K buffers in your system, you should reduce the number of 1K buffers (NBUF) to keep available memory at an acceptable level.

S52KNHBUF — Specifies how many "hash buckets" to allocate for 2K buffers. These are used to search for a buffer given a device number and block number rather than a linear search through the entire list of buffers. **This value must be a power of 2**. Each entry contains 12 bytes. The S52KNHBUF value must be chosen so that the value S52KNBUF divided by S52KNHBUF is approximately equal to 4.

Modifying an Existing Kernel Parameter

The **stune** manual page is used to modify a system tunable parameter from its default value in the *mtune* file. Not every system tunable parameter is contained in the *stune* file. Only those that are to be set to a value other than a system default need be entered there. Although the base UNIX system defines only a few values in *stune,* other add–on packages may have additional added entries into *stune*. Therefore, if the driver package you are building requires modifying a parameter value, the **idtune** command should be used. This command will take individual system parameters, search the *stune* file, and modify an existing value if already there, or add the parameter to *stune* if not defined. The value selected must always be within the minimum and maximum values in the *mtune* file.

A script called **idtune** is provided in */etc/conf/bin* to simplify modifying or adding an *stune* entry. This script is particularly useful when preparing software add-on packages that need to modify a system parameter. Since others may have already changed a system parameter you must be careful that when installing your package that a previous modification is not overwritten. This command will take individual system parameters, search the *stune* file, and modify an existing value if already there, or add the parameter to *stune* if not defined. The value selected must always be within the minimum and maximum values in the *mtune* file. For further information, see the **idtune**(1M) manual page.

Although it is not recommended that a parameter be set outside the *mtune* limits, if it is determined that a parameter must be set higher that permitted in the *mtune* file, you can edit the limits directly. Extreme care must be taken when modifying *mtune* that other values are not modified or deleted.

You should also be aware that the UNIX system kernel forces some parameters to be within preset limits. For example, the parameter NOFILES (number of open files per user process) is forced to fall within the 20/100 limit regardless of how you adjust the *mtune* and *stune* values. You should never modify a *mtune* value unless you have a full understanding of how the parameter is used in the UNIX system.

Reconfiguring the Kernel to Enable New Parameters

After the *stune* and *mtune* files are modified, the system must be reconfigured using the **/etc/conf/bin/idbuild** command. If a build is required, the system must then be shut down and rebooted. If you are modifying the parameter as part of adding your DSP, and your install script already involves **idbuild**, then, no additional build is required. The **idbuild** command builds a new UNIX system kernel and sets a lock file which is detected on the next shutdown. The new UNIX system kernel will be linked to /unix and executed on the next reboot automatically. The specific steps to modify a parameter are as follows:

1. Modify the */etc/conf/cf.d/stune* file.
2. Execute the **/etc/conf/bin/idbuild** command.
3. Change directories to / and execute **/etc/shutdown**.
4. Reboot the system.

What to Do if the System Does Not Boot.

There is a remote possibility that your new kernel will not boot properly. This can happen if a system parameter, or some combination of parameters you have modified has built a UNIX system kernel too large to boot, or has built a kernel that will not initialize properly. In such an event, the following steps can be used to recover your system.

1. Insert floppy disk #1 of your Base System Software.
2. Reset and Reboot the system from the floppy disk.
3. When the prompt "Strike RETURN to install the UNIX system on your hard disk" appears, hit the "del" key to break out of the installation program.
4. Check and mount the hard disk, then copy a good /unix image with the following commands:

   ```
   fsck -y /dev/dsk/0s1
   mount /dev/dsk/0s1 /mnt
   cp /unix /mnt/unix
   umount /dev/dsk/0s1
   ```

5. Remove the floppy disk and reset (use the RESET button or power down/up) the machine to reboot the recovered kernel.
6. Bring up the system as usual, modify */etc/conf/cf.d/stune* parameters back to the original values and execute the idbuild followed by a system reboot sequence as described in Section "Reconfiguring the Kernel to Enable New Parameters."

All About File Systems

All About File Systems

Chapter 6: All About File Systems

What Is a File System

A file system consists of directories, files, and special files. These components allow you to structure and maintain a file system to suit your needs. Each time you log in to the UNIX system, you'll be placed in a specific place in the file system structure. From this point, you can move through the levels of the file system to work in any directory or file you own or to access those belonging to others that you have permission to use.

You can find the disk space available for each file system on your computer with the **df** command. To use the **df** command, type from the UNIX system prompt:

```
# df Enter
```

The file system mount point, special device, available space, and available i-nodes will be displayed for each mounted file system. You can also use the **df** command with a file system argument and display the used and available disk space for that file system. See the **df**(1M) manual page in the *User's/System Administrator's Reference Manual* for additional information.

File System Structure

Every time a file is modified, the UNIX system does a series of file system updates. These updates, when written to the disk, produce a consistent file system. Let's take a look at the components of a file system.

Super-Block
The super-block defines the internal structure and size of a file system. There is one super-block for each file system.

Information Nodes (i-nodes)
An i-node is the internal definition of a file or directory. An i-node contains information about the type of file, the number of directory entries linked to the file, list of blocks claimed by the file, and the size of the file.

Data Blocks
A data block can contain either directory entries or file data. Each directory entry consists of a filename and an i-node number. Each data block contains 1024 bytes.

Indirect Blocks
Indirect blocks are needed to reference the data blocks of large files (over 10 blocks long). There are three types of indirect blocks: single-indirect, double-indirect, and triple-indirect.

First Free-List Block
The free-list blocks are lists of all the blocks not allocated to the super-block, i-nodes, or existing files. The super-block points to the first free-list block.

File System Reliability

The UNIX system is always checking to see if your file systems are in working order. The next few pages will tell you a little about file system reliability and how the UNIX system runs checks on file systems.

File System Integrity

Your computer has several reliability features built in. The following is a brief summary of these features:

- When a file is written to the hard disk, its i-node and blocks are written in an order that ensures maximum reliability. This is known as ordered writes.
- System buffers are periodically written to the hard disk to keep the file contents up to date. This is known as automatic update.
- If the file system becomes corrupted, you will be required to run the **fsck** program to clean up the file system before mounting it. This ensures the reliability of all computer mounted file systems.

File System Checking and Repair

When the UNIX system is booted, your computer runs a consistency check on the status of the *root* file system. If a potential problem exists, the **fsck** program will run automatically to repair the file system. To ensure that file system inconsistencies are repairable, the "**shutdown**" command should always be used to bring the system down.

The **fsck** program is a file system check-and-repair program that makes several consistency checks on a specified file system. Because **fsck** runs automatically on the *root* file system, when the system is booted, you should not have to run **fsck** for the *root* file system.

The **fsck** program can also be run manually to check floppy disks that have UNIX system file systems on them; or if you suspect something is wrong with your file system, you may want to check it. This should only be attempted by expert users. (See Appendix E for **fsck** error messages.)

File System Corruption

A file system can become corrupt in a variety of ways. Three of the most common ways of corrupting a file system are as follows:

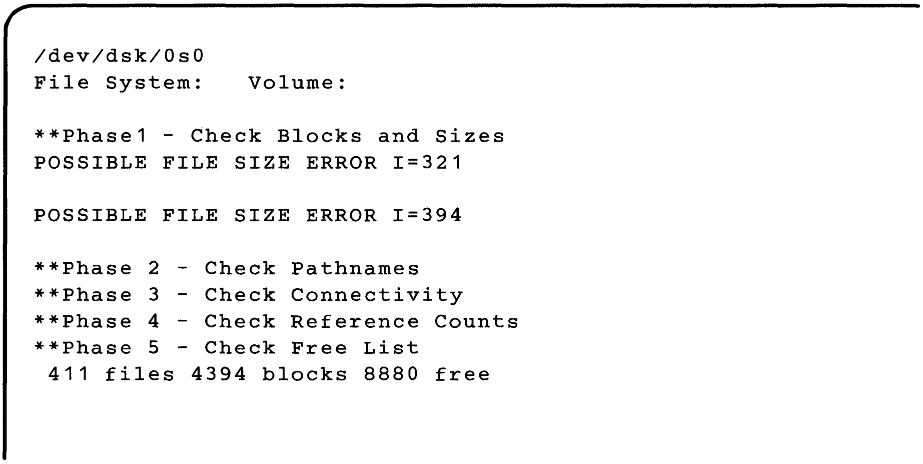
- Improper system shutdown and startup
- Removing media before unmounting its file system
- Hardware failure.

Using fsck

To run the **fsck** program manually, the file system must be mounted with the exception of the *root* file system. The legal **fsck** options are *-b, -f, -y, -n, -s, -S, -t, -q,* and *-D*. The *-y* option is recommended for **fsck**. This option answers yes to all questions prompted by **fsck** and requires no intervention by you. Another recommended option is *-s* which forces rebuilding of the free list in optimal order. The free list becomes disorganized with use. Rebuilding the free list improves performance on subsequently created files. Use the following command line for **fsck**:

```
# /etc/fsck -s -y special Enter
```

You'll get a display on your screen similar to the following:



```
/dev/dsk/0s0
File System:   Volume:

**Phase1 - Check Blocks and Sizes
POSSIBLE FILE SIZE ERROR I=321

POSSIBLE FILE SIZE ERROR I=394

**Phase 2 - Check Pathnames
**Phase 3 - Check Connectivity
**Phase 4 - Check Reference Counts
**Phase 5 - Check Free List
 411 files 4394 blocks 8880 free
```

See the **fsck**(1M) manual pages in the *User's/System Administrator's Reference Manual* for additional information.

fsck Phases

After the initial setup, **fsck** performs successive phases of tests over each file system, performing cleanup, checking blocks and sizes, pathnames, connectivity, reference counts, and the free-block list (possibly rebuilding it).

When an inconsistency is detected, **fsck** reports the error condition to the user. If a response is required, **fsck** will print a prompt message and wait for a response. The following paragraphs explain the meaning of an error condition, the possible responses, and the related error conditions.

The error conditions are organized by the "phase" of the **fsck** program in which they can occur.

For a list and explanation of the error messages you could encounter when using **fsck**, refer to Appendix E.

Phase 1: Check Blocks and Sizes

This phase concerns itself with the i-node list. Activities include checking i-node types, setting up the zero-link-count table, examining i-node block numbers for bad or duplicate blocks, checking i-node size, and checking i-node format.

Phase 1B: Rescan for More DUPS

When a duplicate block is found in the file system, the file system is rescanned to find the i-node that previously claimed that block.

Phase 2: Check Pathnames

This phase concerns itself with removing directory entries pointing to error-conditioned i-nodes from Phase 1 and Phase 1B. Checks are run for root i-node mode and status, directory i-node pointers in range, and directory entries pointing to bad i-nodes.

Phase 3: Check Connectivity

This phase concerns itself with the directory connectivity seen in Phase 2. This part lists error conditions resulting from unreferenced directories and missing or full *lost+found* directories.

Phase 4: Check Reference Counts

This phase concerns itself with the link count information seen in Phase 2 and Phase 3. This part lists error conditions resulting from unreferenced files; missing, or full *lost+found* directories; incorrect link count for files, directories, and special files; unreferenced files and directories; bad and duplicate blocks in files and directories; and incorrect total free i-node counts.

Phase 5: Check Free List

This phase concerns itself with the free-block list. This part lists error conditions resulting from bad blocks in the free-block list, bad free-block count, duplicate blocks in the free-block list, unused blocks from the file system not in the free-block list, and the total free-block count incorrect.

Phase 6: Salvage Free List

This phase concerns itself with the free-block list reconstruction. This part lists error conditions resulting from the blocks-to-skip and blocks-per-cylinder values.

Cleanup

Once a file system has been checked, a few cleanup functions are performed. This lists advisory messages (seen on the next page) about the file system and modifies status of the file system.

***** *FILE SYSTEM STATE SET TO OKAY* *****

A flag in the superblock will be set to indicate that the file system is not corrupted and can be mounted.

X files Y blocks Z free

This is an advisory message indicating that the file system checked contained X files using *Y* blocks leaving Z blocks free in the file system.

```
***** FSCK and the ROOT FILE SYSTEM *****
```

Root is the only file system that can (and must) be checked while mounted. Automated mechanisms are provided for checking the *root* file system. These mechanisms handle a dirty *root* when booting and periodic checks during shutdown. You can also force a check on shutdown. These mechanisms hide the messages from **fsck**. If they were not hidden, you would see the following message:

```
***** ROOT FILE SYSTEM WAS MODIFIED *****
```

This is an advisory message indicating that the *root* file system was modified by **fsck**. If a system reboot is necessary, **fsck** with the *-b* option forces an automatic reboot and prints the message:

**** *SYSTEM WILL REBOOT AUTOMATICALLY* ****

If you decide not to use the automated mechanisms; the *-b* option is not used, and a system reboot is necessary; strike RESET.

The automated procedures establish the proper environment (no processes fiddling with files) for checking *root*.

NOTE

Always use the automated procedures for *root*. Never **fsck** other file systems while they are mounted. If you attempt to do an **fsck** on a mounted file system other than the *root* file system, the following message is displayed:

```
/dev/dsk/ ?? is a mounted file system, ignored.
```

?? is the special device name.

Recommendations for File System Reliability

There are a few things you can do to try to keep your file systems reliable. The following are some recommendations:

- Never remove a floppy disk while the disk drive is on (red light on disk drive is on).
- Never remove a mounted UNIX system floppy disk without unmounting it.
- Always use the **shutdown** procedure before turning off your computer. The shutdown procedure will unmount all file systems.

Recommendations for File System Performance

The UNIX system reads and executes files faster if they are sequential. Initially, the free list of the *root* file system is ordered so new files are sequential. But, file creation and/or deletion activity can disorganize the free list. Automated mechanisms are provided that periodically rebuild the free list of the *root* file system. If you have other active file systems on your computer, periodic use of **fsck** *-s* on them when they are unmounted will improve disk performance.

See the **mkfs**(1M) manual pages in the *User's/System Administrator's Reference Manual* for additional information.

Bad Block Handling

The requirements of bad block handling fall into six catagories.

- Dynamic handling of bad blocks
- Maintenance of a bad block mapping table
- Detection of bad blocks
- Mapping of bad blocks
- Reporting of bad blocks
- Initialization of the hard disk for bad block handling.

Dynamic Handling of Bad Blocks

The basic requirement for the bad block handling feature is that it must be done dynamically, without user intervention. Dynamic handling provides immediate attention to the problem and thus minimizes data loss. It also avoids errors that may be introduced by the user.

Our current implementation reports problems to the console as they are found, without retaining the messages in a log. It is the responsibility of the user (a super user) to carry out the required action: mark the bad block as bad with the *mkpart* command and restore the system from the previous backup, if necessary.

Maintenance of a Bad Block Mapping Table

The bad block mapping table is created and stored on the hard disk when the disk is first formatted. It consists of a bad block list of alternate blocks commonly called the "alternate sector list or surrogate images." These two lists are in a one-to-one correspondence.

The bad block list is used to record the address (on disk) of the blocks found to be bad. The alternate sector list is used to record the address, on disk, of all the reserved sectors to be used as alternates for bad blocks.

Deletion of Bad Blocks

The bad block handling feature should be able to detect two different types of problems.

- Marginal blocks
- Unreadable blocks.

A marginable block is a block which is readable, but with some difficulty. That is, the hard disk controller's Error Correction Code (ECC) algorithm has to be used to successfully read the block.

Once it has been determined that the block in question is marginal, the system will:

- Report the problem to the user
- Copy the data of the marginal block into an available alternate sector
- Mark the marginal block as bad
- Inform the user that the block has been mapped.

Detection of Unreadable Blocks

An unreadable block is a harder problem to solve. There are two possible solutions. One method deals with the possible reconstruction of data to minimize data loss. The other simply accepts that the data is lost.

Reconstruction of data requires that a thorough and an extensive analysis of the block in question is done before any kind or form of data repair can be attempted.

While this method offers higher data conservation, its design and implementation will require a considerable amount of time and effort. Implementation of this method will not occur at this time, however, it should be considered as a future extension to the bad block handling feature.

Simply accepting a data loss when accessing an unreadable block is considered a less conservative method; however, the design and implementation of this method is simpler.

It should also be noted that by having, in place, an implementation which detects and takes care of marginal blocks on the fly, the incidence of potentially unreadable blocks is greatly reduced.

Dynamic Handling of Unreadable Blocks

Whenever a block is found to be unreadable, the system will:

- Inform the user that a bad block has been detected
- Assign an available alternate sector to the bad block (whenever appropriate)
- Warn the user about the data loss
- Determine if the bad block is part of a file system
- If the bad block is part of a file system, mark the file system to enforce a file system check the next time the system is rebooted.

Mapping of Bad Blocks

Mapping of bad blocks will be done dynamically by the system without user intervention. As the system finds potential bad blocks, and/or actual bad blocks on the disk, it will inform the user of the occurrence, analyze the block, and take appropriate action.

Reporting of Bad Blocks

The system will always report the occurrence of potential bad blocks and/or actual bad blocks. The required error message to be displayed by the system for both types of occurrences are provided below.

Reporting of Marginal Blocks

Whenever a potential marginal block is detected by the system, the block is analyzed to determine the kind of action to be taken. The user is then provided with the information.

The following is a list of possible conditions and the respective messages that will be displayed by the system:

- Potential marginal block detected on the sacred area of the hard disk.

NOTE Soft read error corrected by the ECC algorithm: block x drive n.

WARNING **A potential bad block has been detected (block x on drive n) on a sacred area of the hard disk. If this block goes bad, the UNIX System will be lost. Please backup your system.**

- Potential marginal block detected in the UNIX system partition, outside the sacred area of the disk, and the system is out of unassigned alternate sectors.

NOTE Soft read error corrected by the ECC algorithm: block x on drive n.

WARNING **A potential bad block has been detected (block x on drive n). The system is out of spare blocks for surrogates. If the block goes bad, it can not be mapped.**

- Potential marginal block detected in the UNIX system partition, outside the sacred area.

NOTE Soft read error corrected by the ECC algorithm: block x on drive n.

- Verification of a potential marginal block is started.

WARNING **A potential bad block has been detected (block x on drive n). Starting verification to determine if an alternate block needs to be assigned to this block.**

- Verification is complete and the block in question is determined to be marginal.

NOTE Verification completed. An alternate block will be assigned to block x on drive n.

- Verification completed and the block in question is determined to be a good block.

NOTE Verification completed. Block x on drive n is a good block.

- A marginal block has been mapped.

NOTE

An alternate block has been assigned to block x on drive n.

Reporting Unreadable Blocks

The system will always provide the appropriate information whenever an unreadable block is detected. The list below include possible conditions together with the respective messages to be displayed by the system.

- Detection of an unreadable block on the sacred area of the disk.

WARNING

A bad block (block x on drive n) has been detected on a critical area of the disk. The system can not recover from this failure. Must reinstall the UNIX system and restore from previous backup.

- Detection of an unreadable block in the UNIX system partition, outside the sacred area of the disk, and the system is out of unassigned alternate sectors.

WARNING

Block x on drive n is unreadable. Data of this block has been lost.

WARNING

The system is out of spare blocks for alternates. Block x on drive n can not be mapped.

- Detection of an unreadable block on the UNIX system partition outside the sacred area of the disk

WARNING

Block x on drive n is unreadable. Data of this block has been lost.

- An unreadable block is being mapped.

WARNING

An alternate block has been assigned to block x on drive n.

Hard Disk Layout for the UNIX System on the 80386

A description of the current hard disk layout, problems with the layout, and the required changes are provided below.

Required Changes to the Hard Disk Layout

Data structures used only by the UNIX operating system (i.e., *pdinfo, vtoc* and bad block mapping table) should be stored on disk within the UNIX system partition as these pertain to this UNIX system partition and to no other part of the disk. The same applies to the alternate sectors reserved for bad blocks, because these are administered on a per partition basis.

A disk layout for this strategy is as follows:

- Reserve the first sector of cylinder 0 for the primary bootstrap and the *ipart table.*
- Reserve the first 29 sectors of the UNIX system partition for the first-stage and the second-stage bootstrap.
- Reserve the 30th sector of the UNIX system partition for the *pdinfo* and the *vtoc* table.
- Reserve the 31st to the 34th sectors of the UNIX system partition for the bad block mapping table.
- Reserve as many consecutive sectors as needed, beginning with the 35th sector of the UNIX system partition, for alternate sectors.

The implementation of this layout eliminates the restrictions described above since the UNIX system will no longer be required to store essential data on physical cylinder 0 or 1 of the hard disk. Consequently:

- Installation of the UNIX system will never cause the destruction of an MS-DOS partition, regardless of where the MS-DOS partition is located on the disk.
- An MS DOS partition can be started anywhere on the disk. Therefore, the UNIX System *fdisk* command does not have to be changed.
- If any of the sectors where *pdinfo, vtoc* and the bad block mapping table are stored go bad, the hard disk can still be used for the UNIX system, provided that the UNIX system partition starts somewhere else on the disk.

Hard Disk Recovery

Even with bad track handling, it is possible that damage that cannot be repaired automatically could occur to the hard disk (fixed disk). When a hard disk error occurs, you may see a message like the following:

```
A hard disk operation of type x has failed at sector y.
```

The values of x are as follows:

0x02 = Failure to read a sector from disk into memory

0x03 = Failure to write a sector from memory onto disk

0x05 = Failure to format specified track

0x06 = Failure to format specified track and set bad sector flag

0x07 = Failure to format drive starting from track x.

If this happens, try to repair the file system using one of the following subsections.

Recovery From Minor Hard Disk Damage

When the hard disk cannot be repaired automatically, and you still have access to the system, try the following:

1. Save the contents of the hard disk. (Refer to Chapter 4, System Administration in section "Backup to Removable Media.")
2. If you have the System Test Diagnostics floppy disk, use it to run a check for bad tracks on the hard disk.
3. Install the UNIX operating system.

Recovery From Major Hard Disk Damage

When the file system becomes corrupted to the point where the system is inoperable, try the following:

1. Insert the first floppy disk of the Foundation Set into the floppy disk drive and strike RESET.
2. When you see the message asking if you are ready to install the UNIX system, strike [Ctrl] and [Break] at the same time.
3. Run **fsck** from the **root** prompt by typing:

   ```
   # /etc/fsck /dev/rdsk/0s1 [Enter]
   ```

   The **fsck** command will either run with no errors or will request action from the user on repairing the file system. Most of the time answering yes to the questions ask by **fsck** will be sufficient, but be aware this could remove some files.
4. Remove diskette and reboot the system by striking RESET.

Recovery of the UNIX System

There may be a time when booting up the computer you will see the message `/unix is missing or corrupted`. If this should occur, you'll need to replace */unix* with the default */unix*. When */unix* is corrupted, the results are unpredictable. In either case, try the following:

1. Insert the first floppy disk of the Base Foundation Set into the floppy disk drive and strike RESET.
2. When you see the message asking if you are ready to install the UNIX system, break out by striking [Ctrl] [Break] at the same time.
3. Run **fsck** from the **root** prompt by typing:

   ```
   # /etc/fsck /dev/rdsk/0s1 [Enter]
   ```

4. Mount the device **0s1** by typing:

   ```
   # /etc/mount/ /dev/dsk/0s1 /mnt [Enter]
   ```

5. Copy the **/unix** directory by typing:

   ```
   # cp /unix /mnt/unix [Enter]
   ```

6. Unmount the device **0s1** by typing:

   ```
   # /etc/umount /dev/dsk/0s1 [Enter]
   ```

7. Reboot the system by striking RESET.

Warning: When you get the UNIX system prompt, make sure you are logged in as root. You should then reinstall all drivers previously installed. This can be done via the *idbuild* command. See the manual page *idbuild* in the *UNIX system V Release 3.1 User's/System Administrator's Reference Manual.*

Alternate Recovery of the UNIX System

If you have added device drivers or changed configuration, you may want to use this alternate recovery procedure. Yo can copy */unix* to */unix.orig* and reboot from */unix.orig*. The advantage of doing this is that you do not have to reinstall all drivers previously installed. The disadvantage is that */unix.orig* will occupy additional disk space.

If you see the message */unix* is missing or corrupted, replace /unix with the backup */unix.orig*. When */unix* is corrupted, try the following:

1. Insert the first floppy disk of the Base Foundation Set into the floppy disk drive and strike RESET.
2. When you see the message asking if you are ready to install the UNIX system, break out by striking Ctrl Break at the same time.
3. Run **fsck** from the **root** prompt by typing:

   ```
   # /etc/fsck /dev/rdsk/0s1 Enter
   ```

4. Mount the device **0s1** by typing:

   ```
   # /etc/mount/ /dev/dsk/0s1 /mnt Enter
   ```

5. Copy /unix.orig to /unix by typing:

   ```
   # cp /unix /mnt/unix Enter
   ```

6. Unmount the device **0s1** by typing:

   ```
   # /etc/umount /dev/dsk/0s1 Enter
   ```

7. Reboot the system by striking HARDWARE RESET. When the boot prompt is received, enter

   ```
   boot: /unix.orig Enter.
   ```

Creating Backup Copies and Recovering Lost Files

The value of backing up a file is sometimes not appreciated until it's too late and data is lost. Backing up a system takes time, but recovering data that was not backed up takes much longer. The purpose of system backup is to back up your software on floppy disks (or tape) so that you will have it in case data is lost.

Refer to Chapter 4, System Administration for procedures on disk backup and restore.

Creating and Using File Systems

Creating a File System and Making It Available

Once a disk is formatted the next step is to define the file system. The File System Operations function in Chapter 4, System Administration can be used to create and mount a 1K file system on diskette. However, the interface can not be used to create file systems with a logical block size of 512 bytes or 2048 bytes (2K). The **mkfs**(1M) command is used for this purpose.

NOTE

You must have the 2K File System Utilities package installed to make a 2K file system.

The File System Gap

The **mkfs** command is used to create all file systems. One of its optional arguments is the rotational gap. For this computer, the gap should always be 2 (which is the default value). This puts the blocks in ascending order. Thus, new files are more likely to be in sequence, which are read faster. Because of this ordering, another optional argument to **mkfs** (cylinder size) is unimportant since you get the same order in all cases.

Using mkfs

The **mkfs** command has two formats:

```
mkfs special blocks[:i-nodes] [gap blocks/cyl] [–b blocksize]

mkfs special prototype [gap blocks/cyl] [–b blocksize]
```

Notice that in neither format is the file system actually given a name; it is identified by the filename of the special device file on which it will reside. The special device file, traditionally located in the directory **/dev**, is tied to the identifying controller and unit numbers (major and minor, respectively) for the physical device.

In the first format, the only other information that must be furnished on the **mkfs** command line is the number of 512-byte blocks the file system is to occupy. The second format lets you include that information in a prototype file that can also define a directory and file structure for the new file system, and it even allows for reading in the contents of files from an existing file system.

Both formats let you specify information about the interrecord gap and the blocks per cylinder. If this information is not given on the command line, default values are used. The recommendations depend on the logical block size of the file system; (see the discussion of the **-b** option at the end of this section.) The recommended values are different from the defaults used by the command. In the first **mkfs** format, even though the number of blocks in the file is required, the number of i-nodes may be omitted. If the number of i-nodes is omitted, the command uses a default value of one i-node for every four logical storage blocks.

If you use the first format of **mkfs**, the file system is created with a single directory. If you use a prototype file, as noted above, it can include information that causes the command to build and initialize a directory and file structure for the file system. The format of a prototype file is described in the **mkfs**(1M) pages of the *User's/System Administrator's Reference Manual.*

The final option to **mkfs** lets you specify the logical block size to be used for the file system. By default, the file system has a logical block size of 1024 bytes. [With the **-b** option, you can specify a logical block size of 512 bytes, 1024 bytes, or 2048 bytes.]

Choosing Logical Block Size

Logical block size is the size of the chunks the UNIX system kernel uses to read or write files. The logical block size is usually different from the physical block size, which is the size of the smallest chunk that the disk controller can read or write, usually 512 bytes.

An administrator who uses the **mkfs**(1M) command to make a file system may specify the logical block size of the file system. By default, the logical block size is 1024 bytes (1K). The **root** and *usr* file systems are delivered as 1K file systems. Besides 1K file systems, the UNIX system also supports 512-byte file systems and 2048-byte (2K) file systems. To use a 2K file system, you must install the 2K file system package.

To choose a reasonable logical block size for your system, you must consider performance and space. For information on file system space requirements, use the file system block analyzer, **fsba**(1M). For most systems, a 1K file system is a good compromise between system performance and use of space in primary memory and on disk. For a system that uses lots of large executable files and data files, a 2K file system may be a better choice.

Creating a File System on a Floppy Disk

You can create your own file system on floppy disks by using the File Systems Operations feature (in the Administration menu of the main interface menu) or by using the **mkfs** and **labelit** commands. You will actually be specifying the file system that you want on the floppy disk and then mounting the file system as a directory under the UNIX system. See the **volcopy**(1M) manual page in the *User's/System Administrator's Reference Manual* for additional information on the **labelit**(1M).

Creating a file system on floppies can be very useful; i.e., you can have portable file systems, and there will be more room on the hard disk. The maximum size of a file system that can be created on a floppy disk is 702 blocks (512-byte blocks) for a 360 KB floppy disk and 2370 blocks (512-byte blocks) for a 1.2 MB floppy disk.

The following steps,are used to create and identify a file system on a floppy disk:

NOTE You must login as root to do the following procedure. This assures that you have the proper read/write permissions.

1. Login as root.
2. Insert a formatted floppy disk into the floppy disk drive. Refer to Chapter 4, System Administration to learn how to format a floppy disk. The following are the format command lines for 1.2-MB and 360-KB floppy disks:

   For a 1.2-MB floppy, use:

   ```
   # /etc/format /dev/rdsk/f0q15dt Enter
   ```

   For a 360-KB floppy, use:

   ```
   # /etc/format /dev/rdsk/f0d9dt Enter
   ```

3. If you have a 360-KB floppy disk drive, proceed with the next step. If you have a 1.2-MB floppy disk drive, go to Step 5.

4. If you have a 360-KB floppy disk, make a file system of 702 blocks and 160 i-nodes using the following command. The rotational gap is 2 and the blocks per cylinder is 18.

   ```
   # /etc/mkfs /dev/dsk/f0d9d 702:160 2 18 [Enter]
   ```

   The 360-KB floppy disk has eighteen 512-byte blocks per cylinder.

5. If you have a 1.2-MB floppy disk, make a file system of 2370 blocks and 592 i-nodes using the following command. The rotational gap is 2 and the blocks per cylinder is 30.

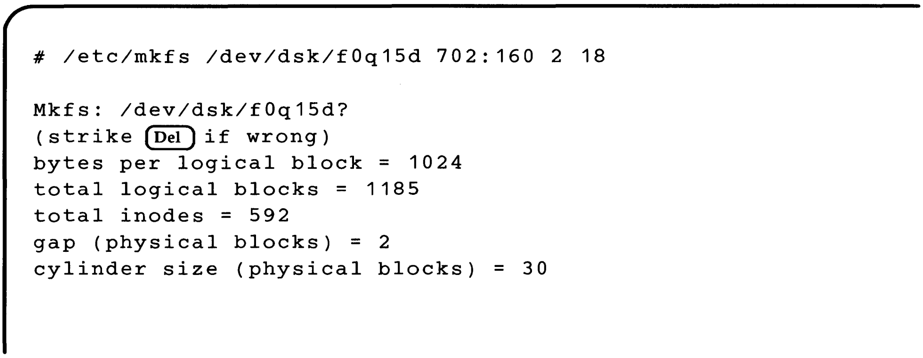
   ```
   # /etc/mkfs /dev/dsk/f0q15d 2370:592 2 30 [Enter]
   ```

   The 1.2-MB floppy disk has thirty 512-byte blocks per cylinder.

Assume that, for the rest of the example, you'll be using a 1.2MB floppy disk. Regardless of what size floppy you have, your screen will look similar to the one below.

NOTE

If the command output in the following screen is not what you wanted, type Ctrl and Break at the same time to cancel the command.

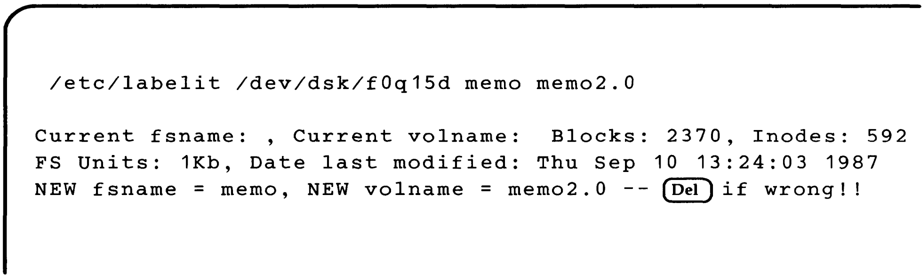


```
# /etc/mkfs /dev/dsk/f0q15d 702:160 2 18

Mkfs: /dev/dsk/f0q15d?
(strike Del if wrong)
bytes per logical block = 1024
total logical blocks = 1185
total inodes = 592
gap (physical blocks) = 2
cylinder size (physical blocks) = 30
```

6. Label the floppy disk file system using the **labelit** command. For this example, assume the file system will be called *memo*. The volume name will be **memo2.0**. Type:

   # /etc/labelit /dev/dsk/f0q15d memo memo2.0 Enter

The screen will look like the following:



```
 /etc/labelit /dev/dsk/f0q15d memo memo2.0

Current fsname: , Current volname:  Blocks: 2370, Inodes: 592
FS Units: 1Kb, Date last modified: Thu Sep 10 13:24:03 1987
NEW fsname = memo, NEW volname = memo2.0 -- Del if wrong!!
```

NOTE

On the computer, DEL refers to the key sequence Ctrl Break to cancel the process.

7. File systems are usually mounted in *root* (/) as directories. Make a directory in the *root* (/) directory with the same name as the file system you're mounting.

   # mkdir /memo Enter

WARNING

You must be root to **mount** and /or **umount** the UNIX system.

8. Mount the file system as follows:

   # /etc/mount /dev/dsk/f0q15d /memo Enter

   # /etc/mount Enter

   The following will be displayed:

```
# /etc/mount /dev/dsk/f0q15d /memo
# /etc/mount
/ on /dev/dsk/0s0 read/write on Wed Jun 12 13:30:10 1987
/memo on /dev/dsk/f0q15d read/write on Wed Jun 12 13:34:10 1987
```

The file system */memo* is now associated with a directory in the *root* file system. As long as the *memo* file system is mounted on */memo,* you can create and modify files on it as if it were an extension to the hard disk.

A directory *lost+found* should be created on the file system for use by **fsck**.

Mounting a file system at a directory that does not match the file system name produces a warning message defining what has been mounted. For example, to mount */dev/dsk/f0q15d* (file system name is *memo*) as directory */mnt,* enter the following command line:

```
# /etc/mount /dev/dsk/ f0q15d /mnt Enter
```

and the following warning message will be displayed:

```
/etc/mount: warning: <memo> mounted as </mnt>
```

Summary: Creating and Converting File Systems

Here is a summary of the steps in creating a new file system or converting an old one to a new logical block size:

1. If the new file system is to be created on a disk partition where an old file system resides, backup the old system. For information, see "Backup to Removable Media" and "Restore from Removable Media" in Chapter 4, System Administration.
2. If the new file system is to be created from an old file system and the new file system is to have a larger logical block size, then, because of fragmentation, the new file system will use more disk space than the old. Use the **fsba**(1M) command to find out the space requirements of the old file system with the new block size.
3. Use the information you get from the **fsba** command to make sure that the disk partition to be used for the new file system is large enough. Use the **mkpart** (1M) command to find the size of your current disk partitions.
4. Use the **mkfs**(1M) command with the **-b** option to make the new file system with the appropriate logical block size. The **mkfs** (1M) command is described in the *User's/System Administrator's Reference Manual.*

5. If necessary, adjust the kernel tunable parameters **NBUF** and **KHF2KBUF** for the logical block size in your system. These parameters control the size of the kernel buffer cache. See the "Tunable Parameters" section of Chapter 5. If you increase the number of 2K buffers, you may want to decrease the number of 1K buffers, so that the memory available for user processes stays reasonably high. If you change kernel tunables, you must rebuild \unix and reboot the machine.

NOTE The parameter **KHF2KBUF** is available along with the 2K FS package installed.

- Populate the new file system—for example, do a restore from a file system backup, or do a cpio(1M) from a mounted file system. "Restore from Removable Media" in Chapter 4, System Administration shows you how to do a file system restore, Refer to the *User's/System Administrator's Reference Manual* for details on cpio.

Unmounting a File System

When you have finished using the file system, you can unmount it. This is done with the **umount** command. All files in the file system to be unmounted must be closed, and you must **cd** to a directory not in this file system. For example, if your current directory (**pwd**) is in the file system you want to unmount, you must **cd** out of the file system before executing the **umount** command. Otherwise, you will get the following message:

```
/etc/umount:device busy
```

To unmount a file system from a 1.2MB floppy disk, type in the following:

```
# /etc/umount /dev/dsk/f0q15d Enter
```

If the file system is unmounted cleanly, there will be no need to run **fsck** next time it's mounted. If it does not unmount cleanly, the next attempt to mount it will produce the following error message:

```
mount: possibly damaged or old file system
on /dev/dsk/f0q15d
mount: check file system or mount read only
```

If this should happen, you can do one of two things:

- Run **fsck** on the file system and mount it again as follows:

  ```
  # /etc/fsck /dev/dsk/f0q15d [Enter]
  ```

- Mount the file system with read permission only as follows:

  ```
  # /etc/mount /dev/dsk/f0q15d /memo -r [Enter]
  ```

Root File System Free Space

A predetermined and finite amount of disk space is allocated for the *root* file system. The unoccupied disk space within this area, called free space, allows for additional and temporary files and often serves as a scratch pad for certain system programs. System administration and other types of programs require *root* file system free space to run. It is recommended that you try to avoid using all the space in the *root* file system. If you should run out of space in *root*, the message `no space on Fixed Disk Device 0x1` will be displayed.

If you see this message, you should manually remove the files you do not need from the **root** file system. Since the system creates the file **/etc/mnttab**(4) during start-up time, it is recommended that you save at least 10 free blocks in the *root* file system before shutting down the machine. The command **df**(1M) can be used to find out how many free blocks are in your file systems. Refer to the *Programmer's Reference Manual* for information on the **mnttab**(4) manual page. Refer to *User's/System Administrator's Reference Manual* for information on the **df**(1M) manual page.

Line Printer Spooler Administration

Line Printer Spooler Administration

Chapter 7: Line Printer Spooler Administration

Introduction to Line Printer Spooler Administration

Line Printer (LP) Spooling is a way of allowing one or two printers to be shared among users. Line Printer Spooling is the name given to the technique of temporarily storing files to be printed in a queue until a printer becomes available. In using a spooling environment, you can customize the system to the users. The flow of printing through the system is regulated by the LP Spooling Utilities.

These utilities allow you to:

- Queue and cancel print requests
- Start and stop the line printer from processing requests
- Change the configuration of printers
- Find the status of the LP system.

For information on setting up your computer with a printer, refer to Chapter 4, System Administration.

Terms You Need to Know

Here are a few terms that need to be defined before a brief summary of the LP Spooling commands can be given.

class Class is the name given to a list of one or more printers. A printer does not have to be assigned to a class but can be assigned to more than one class.

destination The destination is the location to which an LP Spooling output request is sent to be printed or to await printing. An output request directed to a specific printer will only be printed by that printer. An output directed to a class of printers will be printed by the first available printer in that class.

device A device can have one of two meanings: a physical peripheral device that can attach to your computer or a special UNIX system file called a *device file*. The UNIX system uses these device files to access peripherals such as printers and terminals.

printer A device that prints files.

User Commands and Descriptions

Figure 7-1 provides an overview of LP user commands:

USER COMMAND	DESCRIPTION
cancel	Cancels output requests.
disable	Suspends printing of the jobs that are in queue.
enable	Allows a printer to print the jobs that are in queue.
lp	Routes a print job to a destination. The destination can be a printer or a class of printers.
lpstat	Provides the status of a print job or other activities going on in the LP Spooling system.
prtinfo	Simplifies printer job control.

Figure 7-1: LP User Commands

cancel—Stop a Print Request

The **cancel** command cancels any print job from the print queue. For example, if a printer should become jammed, you could cancel the job instead of trying to unjam or fix the printer first. Then inform the person who sent the request that the job was canceled. When you cancel the request, mail is sent telling the person who sent the job that the request was canceled.

The following is the command format for **cancel**:

cancel [printer-name or request id number]

Canceling the printer name will cancel the job currently printing. Also, canceling the request identification number will cancel the print job with that ID number.

disable—Stop Queued Requests From Printing

The **disable** command prevents the specified printer from printing the jobs that are in queue. You may disable a printer for problems such as jammed paper, printer malfunction, or running out of paper, for example.

When a printer is printing a job at the time it is disabled, the job will be reprinted in its entirety when the printer is enabled again.

Print jobs can be sent to a disabled printer. The requests are put into the queue until the printer is enabled.

The following is the command format for **disable**:

disable [**-c**] [**-r**[*reason*]] printers

The **-c** option causes the current print request to be canceled in addition to disabling the printer.

The **-r** option allows you to let other users know why the printer was disabled. *Reason* is a brief explanation of why the printer was disabled. If the reason consists of several words separated by spaces, enclose the reason in double quotes (" ").

enable—Enable Printer Request

The **enable** command allows you to restart a disabled printer. The jobs in the queue will then start to print again. A job, stopped in the middle of printing due to the **disable** command, will start again from the beginning after the printer is started with the **enable** command.

The following is the command format for **enable**:

enable printers

lp—Send a Print Request

The **lp** command allows you to route a print request to a destination. The destination may be a printer or a class of printers. If no destination is given, the request will be routed to a default destination.

Each time an **lp** request is made, a "request id" is assigned to the print job. The request ID consists of two parts: the destination where the request was sent and a sequence number that is unique to the LP system. The request ID is important because you must use it when canceling requests.

The following is the command format for **lp**:

lp [*options*] file(s)

The options are:

-c The **c** option will immediately make a copy of the file(s) to be printed. This option is useful when more than one person is working on a file and you want a printout of the file before other changes can be made.

-d*dest* The **d** option specifies the printer where you want the file(s) to be printed.

-m The **m** option tells the system to send you mail indicating your print job is finished.

-n*number* The **n** option specifies the number of copies you want printed. If no number is given, one copy will be printed.

-o*option* The **o** and *option* refers to the optional printing modes some printers have. For example, some printers may have compressed print or expanded print.

-s The **s** option suppresses messages such as "request id is ..."

-t*title* The **t** option will banner a title on your printout to distinguish it from other jobs. By default, printing of title banners is disabled (see "Printer Interface Programs" in this chapter).

-w The **w** option will allow the system to send you a message when your print job is finished. If you log off before your job is printed, mail will be sent to you.

lpstat—Request LP Status

The **lpstat** command gives you a report on all ongoing processes with the LP Spooling system. It gives you a report on such things as jobs in queue, which job is printing, and which printers are busy or idle.

The following is the command format for **lpstat**:

lpstat [*options*]

The options are:

-a[*list*] The **a** option reports whether or not printers are accepting requests. *List* is a list of printer names and class names.

-c[*list*] The **c** option reports all class names and their members. *List* is a list of class members.

-d The **d** option reports the default destination printer.

-o[*list*] The **o** option reports the status of requests. *List* is a list of printer names, class names, or request id numbers.

-p[*list*] The **p** option reports the status of printers. *List* is a list of printer names.

-r The **r** option is used to determine if the LP scheduler is on or off.

-s The **s** option reports a status summary. This summary includes a list of class names and their members and a list of printers and their associated devices.

-t The **t** option reports all status information and all the information given with the **s** option and also reports the acceptance and idle/busy status of all printers.

-u[*list*] The **u** option reports the status of requests for users. *List* is a list of login names.

-v[*list*] The **v** option identifies the associated device for each LP printer. *List* is a list of printer names.

prtinfo—Printer Information

The **prtinfo** command simplifies printer job control and printer setup. This command allows you to:

- View or change printer queue
- View printer status
- View spooler configuration.

The following is the command format for **prtinfo**:

```
prtinfo
```

Administration Commands and Descriptions

To administer the LP Spooling system, you must be logged in as **root**. All administrative commands can be accessed only from **root**. Figure 7-2 provides an overview of LP administration commands:

ADMIN COMMAND	DESCRIPTION
accept	Permits print-job requests to be queued for a specific destination.
reject	Prevents print jobs from being queued at a specific destination.
lpadmin	Sets up or modifies the LP configuration.
lpmove	Moves output requests from one destination to another.
lpsched	Starts the LP scheduler.
lpshut	Stops the LP scheduler.

Figure 7-2: LP Administration Commands

accept—Allows Print Requests

The **accept** command allows print jobs to be placed in a queue at the named destination(s). The destination is a printer or class of printers.

The following is the command format for **accept**:

/usr/lib/accept *destination(s)*

reject—Prevent LP Requests

The **reject** command is used to stop the LP from routing requests to a destination. For example, if a printer is not working or too many requests are building up at a destination, you may want to prevent new jobs from being queued at that destination. You can use the **reject** command for this type of situation.

If requests are in the queue at the time the **reject** command is invoked, those requests will be printed as long as the printer is enabled. Use the **accept** command to allow requests to be received again.

The following is the command format for **reject**:

/usr/lib/reject [-r[*reason***]]***destination(s)*

The **-r** option allows you to let users know why requests are being rejected. *Reason* is a brief explanation of why requests are being rejected. If the reason consists of several words separated by spaces, enclose the reason in double quotes (" ").

The *destinations* are the printers that are no longer accepting requests.

lpadmin—Configure Printers

The **lpadmin** command is used to reconfigure the LP system when necessary. With a few exceptions, the **lpadmin** command will not alter the LP configuration when the LP scheduler is running.

The **lpadmin** command requires an option. One of the following three options must be used on the command line with **lpadmin**:

/usr/lib/lpadmin -d*[dest]*

/usr/lib/lpadmin -x*dest*

/usr/lib/lpadmin -p*printer*

The **d***[dest]* option is used to define the system default destination. The destination must already exist. This option can be executed when the LP scheduler is running.

The **x***dest* option is used to remove a destination. This option cannot be executed when the scheduler is running.

No other options are allowed with the **d** and **x** options. The following are the options used with the **p***printer* option.

-c*class* The **c** option assigns the printer specified in the **p** option to a specified class.

-e*printer* The **e** option allows you to use an existing interface program for a new printer you're adding to the LP system. When you select this option, the interface program for the printer specified in this option is then copied to the new printer.

-h The **h** option indicates that the new printer you are adding is hardwired to the computer.

-i*interface* The **i** option is used if you are creating a new interface program for the printer specified in the **p** option. *Interface* is the pathname of the new program.

-l The **l** option is used when adding a new printer to indicate that the device associated with the printer is a login terminal.

-m*model* The **m** option is used to select the model interface program you want to use with the printer you're adding to the LP system. Many different model interface programs are supplied with the LP Spooling Utilities. Model interface programs are used to support some of the common printers used with the computer.

-r*class* The **r** option is used to remove a printer from a class.

-v*device* The **v** option must be used when you add a new printer to the LP system. It associates the printer with the UNIX system file specified by *device*. The complete pathname must be given for the file.

lpmove—Move a Request to Another Printer

The **lpmove** command is used to move output requests from one destination to another. You might, for example, use this command to move all queued requests from a printer being repaired to a working printer. Remember, all job requests routed to a destination without a printer are automatically rejected.

You can also move specific requests from one destination to another. The **lpmove** command will not move requests while the LP scheduler is running.

The following is the command format for **lpmove**:

```
/usr/lib/lpmove requests dest
```

Requests are the request identification numbers (request IDs) of jobs waiting to be printed. *Dest* is the destination (printer or class of printers) to which print requests are being moved.

While the **prtinfo** command and office administration facilities require some printer administration, moving request from one queue to another can only be accomplished with **lpmove**.

lpsched—Start the LP Scheduler

The **lpsched** command starts the LP scheduler. The LP scheduler takes the top job request off the queue and distributes it to the appropriate interface program to be printed on the printer. The LP scheduler also keeps track of the job's progress. When the job is finished, the scheduler takes the next job in queue and repeats the process. As long as the LP scheduler is running, jobs requested by the LP will be printed.

The LP scheduler is started automatically each time the system is turned on. This is done by the shell script **lpstartsched** in the **/etc/rc.d** directory. The file is created the first time the system is configured to support a printer.

Every time the scheduler is started, **lpsched** creates a file called *SCHEDLOCK* in the */usr/spool/lp* directory. As long as the *SCHEDLOCK* file is present, the system will not allow another scheduler to run. When the scheduler is stopped, the *SCHEDLOCK* file is removed. If the system should go down abnormally, there is a possibility that *SCHEDLOCK* will not be removed. To be sure that *SCHEDLOCK* is removed, the **lp** file contains a command line to first remove *SCHEDLOCK* before it attempts to start the scheduler.

The following is the command format for **lpsched**:

```
/usr/lib/lpsched
```

lpshut—Stop the LP Scheduler

The **lpshut** command is used to stop the LP scheduler and terminate all printing activity. Many of the **lpadmin** command options cannot be executed unless the scheduler is stopped. All requests in the process of printing will be reprinted in their entirety when the scheduler is restarted.

The following is the command format for **lpshut**:

```
/usr/lib/lpshut
```

Adding an LP Printer

Normally, the computer will have a printer on the parallel port, such as the AT&T 473 dot-matrix printer or the AT&T 457 letter-quality printer. This printer may be identified through the AT&T Administration Interface or the **lpadmin** command.

If you do not want the printer modes for your printer to be reset between the printing of separate print jobs, identify your printer to the system as type "other" through "Setting Up Your Printer" in Chapter 4, System Administration.

Cleaning Up the LP Subsystem

If you experience problems with the LP Subsystem, try to correct the problem first by restarting the printer using the procedure in the section "Printer Restart" in Chapter 4, System Administration. If this does not work, try the following procedure:

1. Log in to the UNIX system as **root** and type:

   ```
   lpstat -t Enter
   ```

   for a list of enabled printers, their status, and their queue entries.

2. Shut down the LP scheduler by typing:

   ```
   /usr/lib/lpshut Enter
   ```

   This cancels the most recent process.

   To find out if there is more than one LP scheduler process running, type **ps -ef** to find the Process IDentification (PID) number. Cancel any additional LP scheduler processes by typing:

   ```
   kill -9 PID# Enter
   ```

3. Deactivate the printer by typing:

   ```
   /usr/lib/reject [-ir[reason]] destination(s) Enter
   ```

   where *destination(s)* is the name of the printer(s).

4. Change directories to */usr/spool/lp* and type the following to clean up the administrative LP files. Press Enter after each line.

   ```
   rm -rf class/*
   rm -rf request/*
   rm -rf member/*
   rm -rf interface/*
   rm -f default
   rm -f pstatus
   rm -f qstatus
   touch pstatus qstatus
   chown lp *status
   ```

   If */etc/.rs232* contains a line that looks like the following:

   ```
   TYPE=PRINTER
   ```

Remove it by typing:

```
rm -f /etc/.rs232 [Enter].
```

5. If everything is in order, type **lpstat -t** [Enter] and the following message will appear:

```
Scheduler not running.  No system default destination.
```

You can now set up your printers again through the AT&T Administration Interface.

Printer Interface Programs

Every line printer must have an interface program. Each print request made with the **lp** command is routed through the appropriate printer interface program before it is printed on the line printer. Some interface programs, referred to as "model" interface programs, are furnished with the Foundation Set.

The interface program will reset the printer to a "normal state" in case the previous printout used special options. The interface program will also route the printout through filters that can modify the data system to suit the printer's needs.

The computer interface programs can operate with or without separator pages (**Banner** pages). The interfaces check for an environment variable **$BANNER** and, if one is present, will print all the identifier information between the LP files. This should be set up as desired in the file */etc/rc.d/lpstartsched.* To enable separator pages, add as the first line in this file:

```
BANNER=/usr/bin/banner; export BANNER
```

Then perform the normal system shutdown and reboot the system.

Writing Interface Programs

Interface programs can be shell procedures, C programs, or other executable programs. The LP model interface programs are all written as shell procedures and can be found in the */usr/spool/lp/model* directory. When a print job is requested, the **lpsched** command routes an output request to printer X. The interface program for printer X is invoked in the directory */usr/spool/lp* as follows:

```
interface/X id user title copies options file ...
```

In the above example, the following arguments are defined:

X	The name of the printer
id	The request id returned by **lp**
user	The logname of the user who made the request
title	The optional title specified by the user
copies	The number of copies requested by the user
options	A blank-separated list of class- or printer-dependent options specified by the user
file	The full pathname of the file to be printed.

When an interface program is invoked, its standard input comes from **/dev/null** and both the standard output and standard error output are directed to the printing device.

Interface programs base their output on the command line arguments. For serial (RS-232) interface printers, you'll want to ensure that the interface program has the correct **stty** modes (terminal characteristics such as baud rate and output options). This is important since you can't count on stty modes having a consistent default state.

You can set the terminal characteristics by adding **stty** command lines in the form of the following:

```
stty mode options <&1
```

The above command line takes the standard input for the **stty** command from the device. An example of an **stty** command line that sets the baud rate at 1200 and sets some of the option modes is shown below.

```
stty -parenb 1200 cs8 cread clocal ixon 0<&1
```

Where:

-parenb disables parity generation and detection.

1200 is the baud rate.

cs8 is the character size.

cread enables the receiver.

clocal assumes a line without modem control.

ixon enables START/STOP output control.

Depending on your printer, the **stty** command line may look different and have other options. Also, different printers have different numbers of columns; be sure the header and trailer for your interface program correspond to your printer. See the **stty**(1) manual pages in the *User's/System Administrator's Reference Manual* for additional information.

For parallel printers, no line discipline need be specified. The interface and filter programs provide access to the printer in the appropriate manner and use control codes to govern printer operations.

Some application programs have their own printer control built in. For output from such programs, the "other" interface program that uses no filter is suitable.

When printing is complete, the interface program exits with a code that indicates the status of the print job. Exit codes are interpreted by **lpsched** as shown in Figure 7-3.

CODE	MEANING TO LPSCHED
0	The print job has completed successfully.
1 to 127	A problem was encountered in this print request (e.g., too many nonprintable characters). This problem will not affect future print jobs. The **lpsched** command notifies you by mail that there was an error in printing the request.
greater than 127	These codes are reserved for internal use by **lpsched**. Interface programs must not exit with codes in this range.

Figure 7-3: Exit Codes as Interpreted by lpsched

When problems occur that are likely to affect future print jobs (e.g., a device filter is missing), it is a good idea to have your interface program disable printers so print requests are not lost. When a busy printer is disabled, the interface program will be terminated with signal 15.

If you have a printer that is not supported by one of the model interface programs, you'll have to write your own program. The shell script for a "simple" printer interface program is shown in the following example. Use this example and modify it to create the interface program you need.

This example is the actual interface for printer type "other," which covers both parallel and serial and raw and filtered modes.

Simple Line Printer Interface Program

```
   # computer lp interface for a simple serial
# line printer
#
# Except in 'raw' mode, all output to the printer is
#       filtered to replace tabs with spaces, using
#       tab stops at every eighth character.
#
# Additionally, the following options have been added:
#       [-]raw[+]  Use RAW mode; default filter is
#                  'cat.' The '+' suffix suppresses
#                  job separation (TOF) both before
#                  and after.
#
#       [-]filtered
#       [-]cooked   Use filtered mode regardless of
#                   whether logical printer was
#                   configured as RAW.
#
# FILES:        /usr/lib/customfilter   (parallel)
#               /usr/lib/customfilterS  (serial)
#
#       If either of these files exist and are
# executable, output will be filtered through it
# instead of through 'pr.'  This allows special
# handling of nonstandard printers through
# user installable custom filters.
#

x="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
printer=`basename $0`
copies=$4

# determine whether printer is RAW or filtered
case "$printer" in
        *R )
                print_mode=raw
                ;;
        * )
                print_mode=filtered
esac
```

```
# process args passed from 'lp' (e.g., -oraw).
for i in $5
do
    case "$i" in
        -raw | raw )    # do not invoke filter
            print_mode=raw
            ;;
        -raw+ | raw+ )  # raw plus no job
                        # separation (print stream)
            print_mode=raw+
            ;;
        -filtered | filtered | -cooked | cooked )
            print_mode=filtered
            ;;
    esac
done

case "$printer" in
    *S | *SR )      # serial printer
stty 1200 -opost ixon -ixany istrip ignpar cread  0<&1
case "$print_mode" in
      raw* )
        ;;
      * )
        if [ -x /usr/lib/customfilterS ] ; then
            myownfilter=/usr/lib/customfilterS
        fi
        ;;
       esac
       ;;
  * )  # parallel printer
       case "$print_mode" in
      raw* )
          ;;
      * )
          if [ -x /usr/lib/customfilter ] ; then
              myownfilter=/usr/lib/customfilter
          fi
       esac
       ;;
esac

if [ X${BANNER:+yes} = Xyes -a "$print_mode" != "raw+" ]
then
    echo "\0014\c"
    echo "$x\n$x\n$x\n$x\n"
    banner "$2"
```

```
        echo "\n"
        user=`grep "^$2:" /etc/passwd | line | cut -d: -f5`
        if [ -n "$user" ]
        then
                echo "User: $user\n"
        else
                echo "\n"
        fi
        echo "Request id: $1    Printer: $printer\n"
        date
        echo "\n"
        if [ -n "$3" ]
        then
                banner $3
        fi
        echo "Print Options:  \n"
        echo "          PrintMode=$print_mode\n"
        echo "          Copies="$copies"\n"
fi
if [ "$print_mode" != "raw+" ] ; then
        echo "\0014\c"
fi

shift; shift; shift; shift; shift
files="$*"
i=1
while [ $i -le $copies ]
do
        for file in $files
        do
                case "$print_mode" in
                 raw* )
                    cat "$file" 2>&1
                    ;;
                 * )
                    if [ -n "$myownfilter" ] ; then
                             cat "$file" | $myownfilter
                    else
                             pr -te8 "$file" 2>&1
                                fi
                esac
                if [ "$print_mode" != "raw+" ] ; then
                        echo "\014\c"
                fi
        done
        i=`expr $i + 1`
done
exit 0
#-----------------------------------------------------------
```

Basic Networking Administration

Basic Networking Administration

Chapter 8: Basic Networking Administration

Introduction to Basic Networking Administration

This chapter describes the administration of the Basic Networking Utilities (BNU). It allows you to communicate with other UNIX system computers using either dial-up or hard-wired communication lines. The BNU comes with your computer UNIX system Foundation Software Set and is specifically part of the Base System Package. The BNU files are located on a separate floppy along with all modem scripts and a file containing a list of supported modems such as: AT&T 2212C, AT&T 2224B, AT&T 4000 Model 1A01, AT&T 4000 Model 1A02, AT&T 4024, and AT&T 4112.

The installation script for the Base System is normally used to select the type(s) of modems that you will be using. However, if you have not made a selection, refer to "Serial Port Setup" in Chapter 4, System Administration. The basic instructions for setting up your computer to communicate with other computers is provided there. This will include connecting your computer to a modem so you can send electronic mail. Once a particular modem has been selected, refer to Appendix D for additional details.

Also, you can refer to the *UNIX System V User Guide,* Chapter 8, Communication Tutorial, for information on using Basic Networking. If Basic Networking is new to you, it might be a good idea to read the "Communication Tutorial" before proceeding with this chapter. This document is not included in your documentation set. Consult your Documentation Roadmap for ordering information.

Basic Networking is complex; the documentation included in this chapter will cover only the information most important for you.

Terms You Need to Know

The following list contains some terms used in Basic Networking that you may not be familiar with, and a brief description of each item.

local machine Refers to the machine on the "near" end of a communication link, normally your computer.

remote machine Refers to a machine on the "far" end of a communication link, normally a machine that your computer talks to.

active machine A machine that has Basic Networking and the hardware required to establish communication links (i.e., Auto Dial Modem).

passive machine A machine that has Basic Networking, but does not have the hardware required to establish communication links.

network A group of machines set up to exchange information and resources.

node A terminating point (machine) on a network.

UUCP This term is used to indicate a group of programs and files that allow systems to send or copy information from one system to another. UUCP means "UNIX system-to-UNIX system copy." In general, it refers to Basic Networking with the exception of the **cu** and **ct** programs. If "uucp" (lowercase) is used in text with bold type (**uucp**), it refers specifically to the **uucp** program or login ID.

Overview of Basic Networking

Basic Networking allows machines using the UNIX operating system to communicate with one another. In general, Basic Networking allows you to do the following:

- Transfer files and send electronic mail to other UNIX system machines as background processes
- Interactively communicate with UNIX system machines, and in some cases, non-UNIX system machines
- Execute commands (restrictive) on a remote machine without logging in
- Call a remote terminal and allow the user of that terminal to log in on your system.

The latter part of this chapter discusses the various ways information is transferred from one machine to another. It also discusses how commands are executed remotely and how your computer can call a remote terminal. But first, you should become familiar with the hardware and software associated with Basic Networking.

What Kind of Hardware Is Needed

Before your computer can communicate with a remote machine, a communication link must be established to the remote machine. There are two types of hardware used to establish a communication link to another machine.

The first is a direct link from a serial port on the computer to a serial port on the other machine. This type of connection is useful when two machines communicate with each other on a regular basis. Even though the RS-232 standard recommends that direct links be limited to 50 feet or less, two machines may be separated by several hundred feet provided that noise on the direct link does not become a problem. If noise becomes a problem or greater distance is needed between the two machines, the transfer rate may need to be decreased or limited distance modems placed at each end of the connection.

The second type of communication link uses the telephone network. In this type of link, the machine that establishes the connection (local machine) must have an Automatic Call Unit (ACU). The ACU dials the specified telephone number upon request from Basic Networking. The called (remote) machine must have a telephone modem capable of answering incoming calls so other machines can contact it through the telephone network. The computer supports a number of automatic dial modems as ACUs. See Appendix D for details.

The Basic Networking Software

Basic Networking is composed of software programs, daemons (background routines), and a supporting data base. The supporting data base contains support files that store information such as telephone numbers, location of the devices (hardware) used to establish links and security restrictions. The software programs and a skeleton data base are supplied in Basic Networking.

The Directories and Their Purpose

There are several directories that contain the programs and support files of Basic Networking. Some of these directories are unique to Basic Networking, while others are common to the UNIX operating system and the computer. The directories used by Basic Networking follow:

/usr/bin — This directory is used by the UNIX operating system and by Basic Networking to store executable programs.

/usr/lib/uucp — This directory is the "HOME" directory for the **uucp** administrative login. It contains the files of the supporting data base and some executable programs.

/usr/spool/locks

This directory contains the lock (LCK) files for the Basic Networking hardware devices. Lock files prevent duplicate conversations and multiple attempts to use the same device.

/usr/spool/uucp

This directory is the "spool directory" for "work" that is to be processed by Basic Networking. It contains a tree-like structure of subdirectories associated with remote machines that your computer wishes to communicate with or has communicated with recently. These subdirectories are also used for administrative purposes such as storing log and status information.

/usr/spool/uucppublic

This directory is the "public" directory for UUCP transfers. The public directory is used to store files that have been sent to your computer. Some remote machines may be restricted to placing files in this directory, while others may have permission to place files elsewhere.

The Software Programs and Their Purpose

There are several types of programs associated with Basic Networking. Some of these programs are used by regular users to transfer data and obtain status information, while others are used for administration purposes or are executed internally. The following paragraphs contain a brief description of the programs and their purpose.

User Programs

cu: Connects your computer to a remote machine and allows you to be logged in on both machines at the same time. This allows you to transfer files or execute commands on either machine without dropping the link.

ct: Connects your computer to a remote terminal and allows you to log in to that terminal. The user of the remote terminal may call into the computer and request that the computer call the remote terminal back. In this case, the computer drops the initial link so that the modem will be available when it is called back.

uucp: Performs all of the preliminary work to allow you to send files to remote machines. It creates "work" files that contain the instructions for transferring the queued file(s). Depending on the options specified, it may make a copy of the file to be transferred in the spool directory. These files are called "data" files. Once the "work" and "data" files have been created, **uucp** calls the **uucico** daemon that attempts to contact the remote machine to deliver the files.

uuto: This program works very similarly to the **uucp** program. In fact, it calls the **uucp** program to create "work" and "data" files. The main difference between **uuto** and **uucp** is the way the transferred files are placed on the remote machine. With **uucp**, you can specify a pathname on the remote machine where you want the files to be placed. With **uuto**, all transferred files are placed in the *uucppublic* directory under */usr/spool/uucppublic/receive*. See the **uuto**(1) manual pages in the *User's/System Administration Reference Manual* for additional information.

uupick: When files are transferred to a machine using **uuto**, **uupick** can be used to retrieve the files placed under */usr/spool/uucppublic/receive*.

uux: This program creates "work" files, "data" files, and "execute" files for executing commands on a remote machine. The "work" file contains the same information as files created by **uucp** and **uuto**. The "execute" files contain the command string to be executed on the remote machine and a list of the "data" files. The "data" files are those files required for the command execution.

uustat: This program displays status information for requested transfers (**uucp**, **uuto**, or **uux**). It also provides you with a means of controlling queued transfers.

Administrative Programs

uulog: This program displays the contents of a specified machine's log file. Individual log files are created for each remote machine that your computer communicates with using the **uucp**, **uuto**, and **uux** programs.

uucleanup: This program has several functions associated with the cleanup of the spool directory. It is normally executed out of a shell script called **uudemon.cleanu** that is started by **cron**. See **cron**(1M) manual page for additional information.

Uutry: This program is a shell script used to test call processing capabilities with a moderate amount of debugging. It invokes the **uucico** daemon to establish the communication link between your computer and the specified machine.

uucheck: This program checks for the presence of Basic Networking directories, programs, and support files. It is also capable of checking certain parts of the *Permissions* file.

Internal Programs

uugetty: This program is very similar to the **getty** program except it permits a line (port) to be used in both directions. The **uugetty** program allows users to log in on your computer; if the line is not in use, it will allow **uucico**, **cu**, or **ct** to use it for dialing out. If one of these programs attempts to dial out when the line is busy, **uugetty** will deny the requester permission and echo a message indicating that the device is unavailable. The **uugetty** is executed as a function of the **init** program.

The UUCP Daemons and Their Purpose

There are three daemons that are part of Basic Networking. These daemons are routines that run as background processes to handle file transfers and command executions.

uucico: This daemon is referred to as the transport program for UUCP requests. It selects the device used for the link, establishes the link to the remote machine, performs the required login sequence, and performs permission checks. It also transfers "data" and "execute" files, logs results, and notifies specified users of transfer completions via **mail**. When the local **uucico** daemon calls a remote machine, it "talks" to the **uucico** daemon on the remote machine during the session. The **uucico** daemon is executed by several methods. It is started by the **uucp**, **uuto**, and **uux** programs to contact the remote machine after all the required "data," "work," and/or "execute" files have been created. It is also started by the **uusched** and **Uutry** programs.

uuxqt: This daemon is the execution program for remote execution requests. It searches the spool directory for "execute" files (X.) that have been sent from a remote machine. When an X. file is found, **uuxqt** opens it to get the list of data files required for the execution. It then checks to see if the required data files are available and accessible. If the files are present and can be accessed, **uuxqt** will check the *Permissions* file to verify that it has permission to execute the requested command. The **uuxqt** daemon is executed out of the **uudemon.hour** shell script that is started by **cron**. **uusched**: This daemon schedules the queued work in the spool directory. Before starting the **uucico** daemon, **uusched** randomizes the order in which remote machines will be called. The **uusched** is executed out of a shell script called **uudemon.hour** that is started by **cron**.

The Supporting Data Base Files and Their Purpose

As mentioned earlier, several of the Basic Networking programs require information contained in support files. These support files are located in the */usr/lib/uucp* directory. The **cu**, **ct**, **uucico**, and **uuxqt** programs require supporting information from the following files:

Devices — This file contains information concerning the location and line speed of the automatic call unit, direct links, and possibly network devices.

Dialers This file contains character strings required to negotiate with network devices (automatic calling devices) in the establishment of connections to remote computers (non-801-type dialers).

Systems This file contains information needed by the **uucico** daemon (and possibly the **cu** program) to establish a link to a remote machine. It contains information such as the name of the remote machine, the name of the connecting device associated with the remote machine, when the machine can be reached, telephone number, login ID, password, etc.

Dialcodes This file contains dial-code abbreviations that may be used in the phone number field of *Systems* file entries.

Permissions This file defines the level of access granted to machines when they attempt to transfer files or remotely execute commands on your computer.

There are several other files that may be considered part of the supporting data base, but these files are not directly related to the process of establishing a link and transferring files. For this reason, discussion of these files is reserved for the Administration section in this chapter.

How Basic Networking Operates

The operation of Basic Networking is briefly described here. There are five programs that allow your computer to communicate with remote machines. The following paragraphs briefly describe what happens when you execute these programs.

ct—Connect a Terminal

The **ct** program instructs your computer to initiate a call to a remote terminal and issue a **getty** to that remote terminal. The **ct** command line must contain the telephone number of the remote terminal. Of course, the remote terminal must be attached to a modem that will automatically answer the call.

When the **ct** command line is issued, the **ct** program will search for an automatic dialer in the *Devices* file with a transfer rate that matches what was specified in the command line. If no transfer rate was specified, it will default to 1200 bps. When **ct** finds the dialer to be used, it attempts to dial the telephone number specified in the command line. If no dialer is available, **ct** asks if it should wait for an available dialer and, if so, how many minutes it should wait. An option is available to override this dialogue. When the modem at the remote terminal answers the call from your computer, it is issued a **getty** (login) process. At this point, the user at the remote terminal may attempt to log in.

The user at a remote terminal can call your computer, log in, and request that the computer call the remote terminal back using the **ct** command. If this scenario is used, the remote user will issue a **ct** command and the link from the remote terminal is dropped. After **ct** finds an available dialer in the *Devices* file, it will call the remote terminal back.

cu—Call a UNIX System

The **cu** command enables you to call another machine and log in as a remote user. The telephone number or node name of the remote machine is required in the command line. If the telephone number is specified, it is passed on to the automatic dial modem. If a system name is specified, the telephone number is obtained from the associated *Systems* file entry. If an automatic dial modem is not used to establish the connection, the line (port) associated with the direct link to the remote machine can be specified in the command line.

If an automatic dial modem is used, the **cu** program will search for an automatic dialer in the *Devices* file with a transfer rate that matches what was specified in the command line. If no speed is specified, the first dialer listed (if available) will be used regardless of its transfer rate. After the link has been established and you have successfully completed the login process, you will be logged in on both computers. This will allow you to execute commands on either computer and/or transfer ASCII coded files from one computer to another. After you have terminated the connection, you will still be logged in on your computer (calling computer). This command can only be executed by an active computer.

uucp—UNIX System-to-UNIX System Copy

The **uucp** command will allow you to transfer file(s) to a remote computer without knowing any details of the connection. All that you are required to know is the name of the remote computer and possibly the login ID of the remote user to whom the file(s) is being sent. The details of the connection are kept in the *Systems* file.

When you enter a **uucp** command, the **uucp** program creates a "work" file and possibly a "data" file for the requested transfer. The "work" file contains information required for transferring the file(s). The "data" file is simply a copy of the specified source file. After these files have been created in the spool directory, the **uucico** daemon will start.

The **uucico** daemon attempts to establish a connection to the remote machine that is to receive the file(s). It first gathers the information required for establishing a link to the remote machine from the *Systems* file. This is how **uucico** knows what type of device to use in establishing the link. Then, **uucico** searches the *Devices* file looking for the devices that match the requirements listed in the *Systems* file. After **uucico** has found an available device, it will attempt to establish the link and log in on the remote machine.

When **uucico** logs in on the remote machine, it starts the **uucico** daemon on the remote machine. The two **uucico** daemons then negotiate the line protocol to be used in the file transfer(s). The local **uucico** daemon then transfers the file(s) to the remote machine, and the remote **uucico** places the file in the specified pathname(s) on the remote machine. After your computer completes the transfer(s), the remote machine may send files that are queued for your computer. The remote machine can be denied permission to transfer these files with an entry in the *Permissions* file. If this is done, the remote machine must establish a link to your computer to perform the transfers. If the remote machine or the device selected to make the connection to the remote machine is unavailable, the request will remain queued in the spool directory. Each hour, **cron** starts **uudemon.hour** which in turn starts the **uusched** daemon. When the **uusched** daemon starts, it searches the spool directory for the remaining "work" files, generates the random order in which these requests are to be processed, and then starts the transfer process (**uucico**) described in the previous paragraphs.

The transfer process described generally applies to an active machine. An active machine (one with calling hardware and Basic Networking software) can be set up to "poll" a passive machine. A passive machine can queue file transfers (because it has Basic Networking software), but it cannot call the remote machine because it does not have the required hardware. The *Poll* file (*/usr/lib/uucp/Poll*) contains a list of machines that are to be polled in this manner. For additional information, refer to the discussion on the *Poll* file and **uudemon.poll** in the Administration section of this chapter.

uuto—Public UNIX System-to-UNIX System Copy

The **uuto** program uses the **uucp** program to build "work" files and "data" files in the spool directory for requested transfers. The difference is that the **uuto** command will not allow you to specify a pathname as a destination for the file. The **uuto** command automatically puts the file in a directory under */usr/spool/uucppublic/receive.* Once the transfer is complete, mail is sent to the appropriate user indicating that a file has arrived and was placed in the public area. That user can then use the **uupick** command to retrieve that file. The **uupick** command will search the public area for files destined to the user and allow the user to interactively delete, print, or move the file to a named directory.

uux—UNIX System-to-UNIX System Execution

The **uux** command allows commands to be executed on a remote machine. It gathers files from various computers, executes the specified command on these files, and sends the standard output to a file on the specified computer. This can be useful when some of the required resources (commands and/or files) are not present on your computer. Remote mail is implemented using the **uux** program, but its execution is embedded in the standard **mail** command. For security, many machines will limit the list of commands that can be executed via **uux** to the default (receipt of mail).

When the **uux** command is issued, the **uux** program creates an "execute" (X.) file that contains the names of the files required for execution, your login name, the destination of the standard output, and the command to be executed. **Uux** also creates "work" (C.) files that are used to gather the files required for execution. These files are then sent to the remote machine, along with the "execute" file, by the **uucico** daemon and placed in the remote spool directory.

Periodically, the **uuxqt** daemon on the remote machine is started to search for X. files in the spool directory. Upon finding an X. file, the **uuxqt** daemon checks to see if all the required data files are available and accessible. It then checks the *Permissions* file to verify that the command(s) listed can be performed. After execution, **uuxqt** sends the standard output to a file on the specified computer.

Administration

The files and tasks associated with the operation of Basic Networking is discussed here. The amount of effort required to administer Basic Networking depends on the amount of "traffic" that enters or leaves your computer. For an average computer, little—if any—intervention with the automatic cleanup functions is required. A computer with a large amount of traffic may require more attention as problems arise.

By now, you've probably realized that the "UUCP facilities" make up the bulk of Basic Networking. The UUCP facilities could generally be defined as all of the programs and support files in Basic Networking with the exception of the **ct** and **cu** programs.

Administrative Files

TM—temporary data file

This data file is created under the spool directory (i.e., */usr/spool/uucp/XXXX)* when receiving a file from another machine. The directory *"XXXX"* has the same name as the remote machine that is sending the file. The temporary data filename has the format:

TM.*pid.ddd*

Where:

pid	is a process ID
ddd	is a sequential 3-digit number starting at zero.

After the entire file is received, the *TM.* file is moved to the pathname specified in the command line. If the file was sent via the **uuto** program, the file will be automatically moved to the public area. If processing is abnormally terminated, the *TM.* file may remain in the "XXXX" directory. This file should be periodically removed.

LCK—lock file

The lock file is created in the */usr/spool/locks* directory for each device in use. A lock file prevents duplicate conversations and multiple attempts to use the same calling device. The filename has the format:

LCK..*str*

where *str* is either a device or computer name. The file may be left in the spool directory if runs abort (usually on computer crashes). The lock file will be ignored (reused) after the parent process is no longer active.

Work (C.) file

The work file is created in a spool directory when work (transfers or remote command executions) has been queued for a remote computer. The name has the format:

C.*sysnxxxx*

where *sys* is the name of the remote computer, *n* is the ASCII character representing the grade (priority) of the work, and *xxxx* is the 4–job sequence number assigned by UUCP. A work file contains the following information:

- Full pathname of the file to be sent or requested
- Full pathname of the destination or ˜user/filename

NOTE The ˜ is shorthand for */usr/spool/uucppublic* and must be included if full pathname is not used.

- User login name
- List of options
- Name of associated data file in the spool directory (If the **-c** or **-p** option was specified, a dummy name [*D*.0] will be used.)
- Mode bits of the source file
- Remote user's login name to be notified upon completion of the transfer.

Data (D.) file

The data file is created when it is specified in the command line to copy the source file to the spool directory. The filename has the following format:

D.*sysnxxxx*

where *sys* is the name of the remote computer, *n* is the character representing the grade (priority) of the work, and *xxxx* is the 4-character job sequence number assigned by **uucp**. The 4-character job sequence number may be followed by a subjob number that is used when there are several *D.* files created for a work (C.) file.

Execute (X.) file

The execute file is created in the spool directory prior to remote command executions. The filename has the following format:

X.*sysnxxxx*

where *sys* is the name of the remote computer, *n* is the character representing the grade (priority) of the work, and *xxxx* is the 4-character sequence number assigned by UUCP.

The execute file contains the following information:

- Requester's login and computer name
- Name of file(s) required for execution
- Input to be used as the standard input to the command string
- Computer and filename to receive standard output from the command execution
- Command string
- Option lines for return status requests.

Machine Log File

The log file is created for each remote machine with which your computer communicates. Each machine may have four log files, one for **uucico, uuxqt, uux,** and/or **uucp** requests depending on the type of communication that has taken place. The log files are kept in the directory */usr/spool/uucp/.Log.* Each day,
these log files are combined and stored in the directory */usr/spool/uucp/.Old* when **uudemon.cleanu** is executed. The combined files are kept 3 days before they're removed. If space is a problem, the administrator may consider reducing the number of days the files are kept by modifying the **uudemon.cleanu** shell file.

Supporting Data Base

The data base that supports Basic Networking is composed of several support files. These support files contain information required by the **uucico** and **uuxqt** daemons during file transfers or remote command executions. All of the support files are located in the */usr/lib/uucp* directory.

Devices File

The **Devices** file (*/usr/lib/uucp/Devices*) contains the information for all of the devices that may be used to establish a link to a remote machine. It contains information for both automatic call units, direct links, and network connections. Although provisions are made for several types of devices, only Modems and Direct Links are supported by AT&T.

This file works very closely with the *Dialers, Systems,* and *Dialcodes* files. It may be beneficial to become familiar with these files before attempting to gain an understanding of the *Devices* file.

Each entry in the *Devices* file has the following format:

Type Line Line2 Class Dialer-Token-Pairs (DTP)

where each field (separated by a space) is defined in the following paragraphs.

Type: This field may contain one of four keywords:

Direct This keyword indicates a Direct Link to another computer (for **cu** connections only).

ACU This keyword indicates that the link to a remote computer is made through an automatic call unit (Automatic Dial Modem). This modem may be connected either directly to the computer or indirectly through a Local Area Network (LAN) switch.

Network This keyword indicates that the link is established through a LAN switch where **Network** is replaced with either **micom** or **develcon**. These two LAN switches are the only ones that contain caller scripts in the *Dialers* file. Other switches may be used if caller scripts are constructed and placed in the *Dialers* file.

Modem Control This keyword causes the device to be opened with O_NDELAY set (so the open does not hang waiting for carrier). After the open, O_NDELAY is cleared.

System-Name This keyword indicates a direct link to a particular machine where **System-Name** is replaced by the name of the particular computer. This naming scheme is used to convey the fact that the line associated with this *Devices* entry is for a particular machine.

The keyword used in the Type field is matched against the third field of *Systems* file entries as follows:

Devices: ACU tty1,M - 1200 penril

Systems: eagle Any ACU 1200 3-2-5-1 ogin: nuucp ssword: Oakgrass

Line: This field contains the device name of the line (port) associated with the *Devices* entry. For instance, if the Automatic Dial Modem for a particular entry was attached to the /dev/tty1 line, the device name would be tty1. The ,M indicates that modem control is being used. **Line2**: If the ACU keyword was used in the Type field and the ACU is an 801-type dialer, this field would contain the device name of the 801 dialer. It should be noted that 801-type ACUs do not contain a modem. Therefore, a separate modem is required and would be connected to a different line (defined in the Line field). This means that one line would be allocated to the modem and another to the dialer. Since the computer will not normally use this type of configuration, this field is ignored, but must contain a pseudo entry as a placeholder (use a "-" as a placeholder).

Class: If an ACU keyword is used, this may be just the speed of the device. It may contain a letter and speed (e.g., C1200, D1200, etc.) to differentiate between classes of dialers (centrex or DIMENSION PBX). This is necessary because many larger offices may have more than one type of telephone network. One network may be dedicated to serving only internal office communications while the other handles the external communications. Therefore, it is necessary to distinguish which line(s) should be used for internal communications and which should be used for external communications. The same distinction must be made in the *Systems* file because a match is made against the fourth field of *Systems* file entries as follows:

Devices: ACU tty1,M - **D1200** penril

Systems: eagle Any ACU **D1200** 3-2-5-1 ogin: nuucp ssword: Oakgrass

Some devices can be used at any speed, so the keyword "Any" may be used in the Class field. If "Any" is used, the line will match any speed requested in a **Systems** entry. If this field is "Any" and the *Systems* Class field is "Any," the speed will default to 1200 bps.

Dialer-Token-Pairs: This field contains pairs of dialers and tokens. The "dialer" portion may be an automatic dial modem, or "direct" for Direct Link devices. The "token" portion may be supplied immediately following the "dialer;" or if not present, it can be taken from the *Systems* file. This field has the format:

```
    dialer-token dialer-token
```

where the last pair may or may not be present, depending on the associated device (dialer). In most cases, the last pair will contain only a "dialer" and the "token" is retrieved from the Phone field of the *Systems* entry. The DTP field may be structured four different ways, depending on the device associated with the entry:

1. If a direct link is established to a particular computer, the DTP field of the associated entry will contain the keyword "direct." This is true for both types of direct link entries, Direct and System-Name (refer to discussion on the Type field).

2. If an automatic dialing modem is connected directly to an computer port, the DTP field of the associated *Devices* entry will only have one pair. This pair would normally be the name of the modem. This name is used to match the particular *Devices* entry with an entry in the *Dialers* file. Therefore, this "dialer" must match the first field of a *Dialers* file entry as follows:

   ```
   Devices: ACU tty1,M - 1200 ventel

   Dialers: ventel =&-% " " \M\r\p\r\c $ <K\T%%\r>\c ONLINE!\m
   ```

   Notice that only the "dialer" (**ventel**) is present in the DTP field of the *Devices* entry. This means that the "token" to be passed on to the dialer (in this case the telephone number) is taken from the Phone field of a *Systems* file entry.

3. If an automatic dialing modem is connected to a local area network (LAN), the computer must first access the switch and the switch will make the connection to the automatic dialing modem. This type of entry would have two pairs. The "dialer" portion of each pair (fifth and seventh fields of entry) is used to match entries in the *Dialers* file as follows:

   ```
   Devices: ACU tty1 - 1200 develcon vent ventel

   Dialers: ventel =&-% " " \M\r\p\r\c $ <K\T%%\r>\c ONLINE!\m
   Dialers: develcon " " " " \pr\ps\c est:\077 \E\D\e \007
   ```

   In the first pair, develcon is the "dialer" and vent is the "token" that is passed to the Develcon switch to tell it which device (ventel modem) to connect to the computer. This token would be unique for each LAN switch since each switch may be set up differently. Once the ventel modem has been connected, the second pair is accessed where ventel is the "dialer" and the "token" is retrieved from the *Systems* file.

4. If a machine that you want to communicate with is on the same local network switch as your computer, your computer must first access the switch and then the switch can make the connection to the other machine. In this type of entry, there is only one pair. The "dialer" portion is used to match a *Dialers* entry as follows:

   Devices: develcon tty1 - 1200 **develcon** \D

   Dialers: **develcon** " " " " \pr\ps\c est:\007 \E\D\e \007

   As shown, the "token" is left blank. This indicates that it's retrieved from the *Systems* file. The *Systems* file entry for this particular machine will contain the token in the Phone field that is normally reserved for the telephone number of the machine (refer to "Systems File" Phone field). This type of DTP contains an escape character (\D) which ensures that the contents of the Phone field will not be interpreted as a valid entry in the *Dialcodes* file.

There are two escape characters that may appear at the end of a DTP field:

\T Indicates that the Phone (token) field should be translated using the *Dialcodes* file. This escape character is normally placed in the *Dialers* file for each caller script associated with an automatic dial modem (penril, ventel, etc.). Therefore, the translation will not take place until the caller script is accessed.

\D Indicates that the Phone (token) field should not be translated using the *Dialcodes* file. If no escape character is specified at the end of a **Devices** entry, the \D is assumed (default). A \D is also used in the *Dialers* file with entries associated with network switches (develcon and micom).

Dialers File

The *Dialers* file (*/usr/lib/uucp/Dialers*) is used to specify the initial handshaking that must take place on a line before it can be made available for transferring data. This initial handshaking is usually a sequence of ASCII strings that are transmitted and expected and is often used to dial a telephone number using an ASCII dialer (such as the AT&T 2212C Modem). As shown in the above examples, the fifth field in a *Devices* file entry is used as an index into the *Dialers* file. Here an attempt is made to match the *Devices* field with the first field of each *Dialers* entry. In addition, each odd numbered *Devices* field starting with the seventh position is used as an index into the *Dialers* file. Changes must be made using one of the editors (**ed** or **vi**).

If the match succeeds, the *Dialers* entry is interpreted to perform the dialer negotiations. The first field matches the fifth and additional odd numbered fields in the *Devices* file. The second field is used as a translate string (the first of each pair of characters is mapped to the second character in the pair). This is usually used to translate "=" and "-" into whatever the dialer requires for "wait for dial tone" and "pause." The remaining fields are "expect-send" strings. The following *Dialers* file entries are typical examples.

```
att4000  =,-,    "" \M\dat\r\c OK\r \EATDT\T\r\c CONNECT \m\c
penril   =W-P    "" \d > s\p9\c )-W\p\r\ds\p9\c-) y\c : \E\TP > 9\c OK
ventel   =&-%    "" \M\r\p\r\c $ <K\T%%\r>\c ONLINE!\m
hayes    =,-,    "" \M\dAT\r\c OK\r \EATDT\T\r\c CONNECT\m\c
rixon    =&-%    "" \d\r\r\c $ s9\c )-W\r\ds9\c-) s\c : \T\r\c $ 9\c LINE
vadic    =K-K    "" \005\p *-\005\p-*\005\p-* D\p BER? \E\T\e \r\c LINE
develcon""       "" \pr\ps\c est:\007 \E\D\e \007
micom    ""      ""\s\c NAME? \D\r\c GO
direct
```

The meaning of some of the escape characters (those beginning with "\") used in the *Dialers* file are shown in the following list:

\p	Pause (approximately ¼ to ½ second)
\d	Delay (approximately 2 seconds)
\D	Phone number or token without **Dialcodes** translation
\M	Sets no modem control
\T	Phone number or token with **Dialcodes** translation

\K	Insert a BREAK
\E	Enable echo checking (for slow devices)
\e	Disable echo checking
\r	Carriage return
\c	No new-line
\m	Restores modem control
\n	Send new-line
\nnn	Send octal number.

Additional escape characters that may be used are listed in the section discussing the *Systems* file. The penril entry in the *Dialers* file is executed as follows. First, the telephone number argument is translated, replacing any "=" with a (pause). The handshake given by the remainder of the line works as follows:

" "	Wait for nothing.
\d	Delay for 2 seconds.
>	Wait for a ">".
s\p9\c	Send an "s," pause for ½ second, send a "9," send no terminating new-line.
)-W\p\r\ds\p9\c-)	Wait for a ")". If it is not received, process the string between the "-" characters as follows. Send a "W," pause, send a carriage return, delay, send an "s," pause, send a "9," without a new-line, and then wait for the ")".
y\c	Send a "y, without a new-line."
:	Wait for a ":".
\M	Sets no modem control (CLOCAL).
\m	Restores modem conrol. Typically, CLOCAL is set for the duration of the dialer chat, then cleared (so **uucico**, **cu**, or **ct** will detect dropped lines) once connected to the remote system.

\E\TP	Enable echo checking. (From this point on, whenever a character is transmitted, it will wait for the character to be received before doing anything else.) Then, send the telephone number followed by a pause character (P). The \T means take the telephone number passed as an argument and apply the *Dialcodes* translation and the modem function translation specified by field number 2 of this entry.
>	Wait for a ">".
9\c	Send a "9" without a new-line.
OK	Waiting for the string "OK."

Systems File

The *Systems* file (*/usr/lib/uucp/Systems*) contains the information needed by the **uucico** daemon to establish a communication link to a remote machine. Each entry in the file represents a machine that can be called by the computer. Furthermore, only those machines listed in the *Systems* file will be permitted to communicate with your computer via Basic Networking (UUCP), unless the execute permissions for **remote.unknown** are changed to permit communications with other machines (refer to "**remote.unknown**"). More than one entry may be present for a particular machine. The additional entries represent alternate communication paths that will be tried in sequential order.

Each entry in the *Systems* file has the following format:

System-Name Time Type Class Phone Login

where each field is defined in the following paragraphs.

System-name: This field contains the node name of the remote machine.

Time: This field is a string that indicates the day of week and time of day when the remote machine can be called. The day portion may be a list containing some of the following:

```
Su Mo Tu We Th Fr Sa
```

Wk: For any weekday

Any: For any day

Never: For a passive arrangement with the remote machine. In this case, the computer will never initiate a call to the remote machine. The call must be initiated by the remote machine. The computer is in a passive mode in respect to the remote machine. (See discussion of *Permissions* file.)

The time should be a range of times such as 0800-1230. If no time portion is specified, any time of day is assumed to be allowed for the call. Note that a time range that spans 0000 is permitted. For example, 0800-0600 means all times are allowed other than times between 6 a.m. and 8 a.m. An optional subfield is available to specify the minimum time (in minutes) before a retry, following a failed attempt. The subfield separator is a semicolon (;). For example, **Any ;9** is interpreted as call any time; but wait at least 9 minutes before retrying if a failure occurs.

Type: This field contains the device type that should be used to establish the communication link to the remote machine. The *Devices* file is searched for the device type listed and the device found is used to establish the connection (if available). The following keywords may appear in this field:

ACU This keyword indicates that the link to a remote computer is made through an automatic call unit (Automatic Dial Modem). This modem may be connected either directly to the computer or indirectly through a Local Area Network (LAN) switch.

Network This keyword indicates that the link is established through a LAN switch where **Network** is replaced with either **micom** or **develcon**. These two LAN switches are the only ones that contain caller scripts in the *Dialers* file. Other switches may be used if caller scripts are constructed and placed in the *Dialers* file.

System-Name This keyword indicates a direct link to a particular machine where *System-Name* is replaced by the name of the particular computer (should be same as field one).

The keyword used in this field is matched against the first field of *Devices* file entries as follows:

Systems: eagle Any **ACU** D1200 3-2-5-1 ogin: nuucp ssword: Oakgrass

Devices: **ACU** tty1 - D1200 penril

Class: This field is used to indicate the transfer speed of the device used in establishing the communication link. It may contain a letter and speed (e.g., C1200, D1200, etc.) to differentiate between classes of dialers (refer to the discussion on the "Devices File," Class field). Some devices can be used at any speed, so the keyword "Any" may be used. This field must match the Class field in the associated *Devices* entry as follows:

Systems: eagle Any ACU **D1200** 3-2-5-1 ogin: nuucp ssword: Oakgrass

Devices: ACU tty1 - **D1200** penril

Phone: This field is used to provide the telephone number (token) of the remote machine for automatic dialers (LAN switches). The telephone number is made up of an optional alphabetic abbreviation and a numeric part. The abbreviation must be one that is listed in the *Dialcodes* file. In this string, an equal sign (=) tells the ACU to wait for a secondary dial tone before dialing the remaining digits. A dash in the string (-) instructs the ACU to pause 4 seconds before dialing the next digit.

If your computer is connected to a LAN switch, you may access other machines that are connected to that switch. The **Systems** entries for these machines will not have a telephone number in the Phone field. Instead, this field will contain the "token" that must be passed on to the switch so it will know which machine the computer wishes to communicate with. The associated **Devices** entry should have a \D at the end of the entry to ensure that this field is not translated using the *Dialcodes* file. For direct connections, the telephone field is ignored. A "-" should be used as a place holder.

Login: This field contains the login information given as a series of fields and subfields of the format:

[expect send] ...

where *expect* is the string that is received and *send* is the string that is sent when the *expect* string is received. The expect field may be made up of subfields of the form:

expect[-send-expect]...

where the *send* is sent if the prior *expect* is not successfully read and the *expect* following the *send* is the next expected string. For example, with "login--login," UUCP will expect "login." If UUCP gets "login," it will go on to the next field. If it does not get login, it will send nothing followed by a new-line, then look for login again. If no characters are initially expected from the remote machine, the characters " " (null string) should be used in the first expect field. Note that all send fields will be sent followed by a new-line unless the send string is terminated with a **c**.

There are several escape characters that cause specific actions when they're a part of a string sent during the login sequence. The following escape characters are useful in UUCP communications:

\N Send a null character.

\b Send a backspace character.

\c If at the end of a string, suppress the new-line that is normally sent. Ignored otherwise.

\d Delay 2 seconds before sending or reading more characters.

\p Pause for approximately ¼ to ½ second.

\n Send a new-line character.

\r Send a carriage return.

\s Send a space character.

\t Send a tab character.

**** Send a \ character.

EOT Send EOT character (actually EOT new line is sent twice).

BREAK Send a break character.

\ddd Collapse the octal digits (ddd) into a single character and send that character.

Dialcodes File

The *Dialcodes* file (*/usr/lib/uucp/Dialcodes*) contains the dial-code abbreviations used in the Phone field of the **Systems** file. Each entry has the format:

```
abb dial-seq
```

where **abb** is the abbreviation used in the *Systems* file (Phone field), and **dial-seq** is the dial sequence that is passed to the dialer when that particular **Systems** entry is accessed.

The entry:

```
jt 9=847-
```

would be set up to work with a Phone field in the *Systems* file such as jt7867. When the entry containing jt7867 is encountered, the sequence 9=847-7867 would be sent to the dialer.

Permissions File

The *Permissions* file (*/usr/lib/uucp/Permissions*) is used to specify the permissions that remote machines have with respect to login, file access, and command execution. Options are provided for restricting the ability to request files and the ability to receive files queued by the local site. In addition, an option is available to specify the commands that a remote site can execute on the local machine. Changes must be made using one of the editors (**vi** or **ed**).

How Entries Are Structured

Each entry is a logical line with physical lines terminated with a \ to indicate continuation. Entries are made up of "white space" delimited options. Each option is a name/value pair. These are constructed by an option name followed by an "=" and the value. Note that no white space is allowed within an option assignment.

Comment lines begin with a "#," and they occupy the entire line up to a new-line character. Blank lines are ignored (even within multiline entries). There are two types of **Permissions** entries:

LOGNAME	Specifies permissions that take effect when a remote machine logs in on (calls) your computer.
MACHINE	Specifies permissions that take effect when your computer logs in on (calls) a remote machine.

LOGNAME entries will contain a LOGNAME option, and MACHINE entries will contain a MACHINE option.

Considerations

The following items should be considered when using the *Permissions* file to restrict the level of access granted to remote machines:

1. All login IDs used by remote machines to log in for UUCP-type communications must appear in one and only one LOGNAME entry.
2. Any site that is called whose name does not appear in a MACHINE entry will have the following default permissions/restrictions:
   - Local send and receive requests will be executed.
   - The remote machine can send files to your computer */usr/spool/uucppublic* directory.
   - The commands sent by remote machine for execution on your computer must be one of the default commands; usually **rmail**.

Options

This section provides the details of each option, specifying how they're used and their default values.

Request. When a remote machine calls your computer and requests to receive a file, this request can be granted or denied. The REQUEST option specifies whether or not the remote machine can request to set up file transfers from your computer. The string:

```
REQUEST=yes
```

specifies that the remote machine can request to transfer files from your computer. The string:

```
        REQUEST=no
```

specifies that the remote machine cannot request to receive files from your computer. The "no" string is the default value. It will be used if the REQUEST option is not specified. The REQUEST option can appear in either a LOGNAME (remote calls you) entry or a MACHINE (you call remote) entry.

Sendfiles. When a remote machine calls your computer and completes its work, it may attempt to take work that your computer has queued for it. The SENDFILES option specifies whether or not your computer can send the work queued for the remote machine. The string:

```
        SENDFILES=yes
```

specifies that the computer may send the work that is queued for the remote machine as long as it logged in as one of the names in the LOGNAME option. This string is mandatory if the computer is in a "passive mode" with respect to the remote machine. The string:

```
        SENDFILES=call
```

specifies that files queued in your computer will only be sent when the computer calls the remote machine. The call value is the default for the SEND-FILE option. This option is only significant in LOGNAME entries since MACHINE entries apply when calls are made out to remote machines. If the option is used with a MACHINE entry, it will be ignored.

Read and Write. These options specify the various parts of the file system that **uucico** can read from or write to. The READ and WRITE options can be used with either MACHINE or LOGNAME entries.

The default for both the READ and WRITE options is the *uucppublic* directory as shown in the following strings:

```
READ=/usr/spool/uucppublic WRITE=/usr/spool/uucppublic
```

The strings:

```
        READ=/ WRITE=/
```

specify permission to access any file that can be accessed by a local user with "other" permissions.

The value of these entries is a colon-separated list of pathnames. The READ option is for requesting files, and the WRITE option is for depositing files. One of the values must be the prefix of any full pathname of a file coming in or going out. To grant permission to deposit files in /usr/news as well as the public directory, the following values should be used with the WRITE option:

```
WRITE=/usr/spool/uucppublic:/usr/news
```

It should be pointed out that if the READ and WRITE options are used, all pathnames must be specified because the pathnames are not added to the default list. For instance, if the */usr/news* pathname was the only one specified in a WRITE option, permission to deposit files in the public directory would be denied.

Noread and Nowrite. The NOREAD and NOWRITE options specify exceptions to the READ and WRITE options or defaults. The strings:

```
READ=/ NOREAD=/etc WRITE=/usr/spool/uucppublic
```

would permit reading any file except those in the */etc* directory (and its subdirectories - remember, these are prefixes) and writing only to the default */usr/spool/uucppublic* directory. NOWRITE works in the same manner as the NOREAD option. The NOREAD and NOWRITE can be used in both LOGNAME and MACHINE entries.

Callback. The CALLBACK option is used in LOGNAME entries to specify that no transaction will take place until the calling system is called back. The string:

```
CALLBACK=yes
```

specifies that your computer must call the remote machine back before any file transfers will take place.

The default for the CALLBACK option is:

```
CALLBACK=no
```

The CALLBACK option is very rarely used. Note that if two sites have this option set to "yes" for each other, a conversation will never get started.

Commands.

WARNING

The COMMANDS option can be hazardous to the security of your system. Use it with extreme care.

The **uux** program will generate remote execution requests and queue them to be transferred to the remote machine. Files and a command are sent to the target machine for remote execution. The COMMANDS option can be used in MACHINE entries to specify the commands that a remote machine can execute on your computer. The string:

```
COMMANDS=rmail
```

indicates the default commands that a remote machine can execute on your computer. If a command string is used in a MACHINE entry, the default commands will be overridden. For instance, the entry:

```
MACHINE=owl:raven:hawk:dove \
COMMANDS=rmail:rnews:lp
```

overrides the COMMAND default such that the command list for machines owl, raven, hawk, and dove now consists of **rmail**, **rnews**, and **lp**. In addition to the names as specified above, there can be full pathnames of commands. For example:

```
COMMANDS=rmail:/usr/lbin/rnews:/usr/local/lp
```

specifies that command **rmail** uses the default path. The default paths for the computer are */bin*, */usr/bin*, and */usr/lbin*. When the remote machine specifies **rnews** or **/usr/lbin/rnews** for the command to be executed, **/usr/lbin/rnews** will be executed regardless of the default path. Likewise, **/usr/local/lp** is the **lp** command that will be executed.

Including the "ALL" value in the list means that any command from the remote machine(s) specified in the entry will be executed. If you use this value, you give the remote machine full access to your computer.

The string:

```
COMMANDS=/usr/lbin/rnews:ALL:/usr/local/lp
```

illustrates two points. The ALL value can appear anywhere in the string. And, the pathnames specified for rnews and lp will be used (instead of the default) if the requested command does not contain the full pathnames for **rnews** or **lp**.

The VALIDATE option should be used with the COMMANDS option whenever potentially dangerous commands like **cat** and **uucp** are specified with the COMMANDS option. Any command that reads or writes files is potentially dangerous to local security when executed by the UUCP remote execution daemon (**uuxqt**).

Validate. The VALIDATE option is used with the COMMANDS option when specifying potentially dangerous commands. It is used to provide a certain degree of verification of the caller's identity. The use of the VALIDATE option requires that privileged machines have a unique login/password for UUCP transactions. An important aspect of this validation is that the login/password associated with this entry be protected. If an outsider gets that information, that particular VALIDATE option can no longer be considered secure.

A great deal of consideration should be given to providing a remote machine with a privileged login and password for UUCP transactions. Giving a remote machine a special login and password with file access and remote execution capability is like giving anyone on that machine a normal login and password on your computer. Therefore, if you cannot trust someone on the remote machine, do not provide that machine with a privileged login and password.

The LOGNAME entry:

```
LOGNAME=uucpfriend VALIDATE=eagle:owl:hawk
```

specifies that if one of the remote machines that claims to be eagle, owl, or hawk logs in on your computer, it must have used the login **uucpfriend**. As can be seen, if an outsider gets the **uucpfriend** login/password, masquerading is trivial. But what does this have to do with the COMMANDS option that only appears in MACHINE entries? It links the MACHINE entry (and COMMANDS option) with a LOGNAME entry associated with a privileged login. This link is needed because the execution daemon is not running while the remote machine is logged in. In fact, it is an asynchronous process with no knowledge of what machine sent the execution request. Therefore, the real question is how does your computer know where the execution files came from?

Each remote machine has its own "spool" directory on your computer. These spool directories have write permission given only to the UUCP programs. The execution files from the remote machine are put in its spool directory after being transferred to your computer. When the **uuxqt** daemon runs, it can use the spool directory name to find the MACHINE entry in the *Permissions* file and get the COMMANDS list; or if the machine name does not appear in the *Permissions* file, the default list will be used.

The following example shows the relationship between the MACHINE and LOGNAME entries:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
COMMANDS=ALL \
READ=/  WRITE=/

LOGNAME=uucpz VALIDATE=eagle:owl:hawk \
REQUEST=yes SENDFILES=yes \
READ=/  WRITE=/
```

These entries provide unlimited read, write, and command execution for the remote machines eagle, owl, and hawk. The ALL value in the COMMANDS option means that any command can be executed by either of these machines. Using the ALL value gives the remote machine unlimited access to your computer. In fact, files that are only readable or writable by user "uucp" (like *Systems* or *Devices*) can be accessed using commands like **ed**. This means a user on one of the privileged machines can write in the *Systems* file as well as read it!

In the first entry, you must make the assumption that when you want to call one of the machines listed, you're really calling either eagle, owl, or hawk. Therefore, any files put into one of the eagle, owl, or hawk spool directories are put there by one of those machines. If a remote machine logs in and says that it is one of these three machines, its execution files will also be put in the privileged spool directory. You, therefore, have to validate that the machine has the privileged login "**uucpz**."

MACHINE Entry for "Other" Systems

You may want to specify different option values for the machines your computer calls that are not mentioned in specific MACHINE entries. This may occur when there are many machines calling in, and the command set changes from time to time. The name "OTHER" for the machine name is used for this entry as follows:

```
MACHINE=OTHER \
COMMANDS=rmail:rnews:/usr/lbin/Photo:/usr/lbin/xp
```

All other options available for the MACHINE entry may also be set for the machines that are not mentioned in other MACHINE entries.

Combining MACHINE and LOGNAME Entries

It is possible to combine MACHINE and LOGNAME entries into a single entry where the common options are the same. For example, the two entries:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
  READ=/  WRITE=/

LOGNAME=uucpz REQUEST=yes SENDFILES=yes \
  READ=/  WRITE=/
```

share the same REQUEST, READ, AND WRITE options. These two entries can be merged into one entry as follows:

```
MACHINE=eagle:owl:hawk REQUEST=yes \
LOGNAME=uucpz SENDFILES=yes \
  READ=/  WRITE=/
```

Sample Permissions Files

Example 1. This first example represents the most restrictive access to your computer.

```
LOGNAME=nuucp
```

It states that login "**nuucp**" has all the default permissions/restrictions:

- The remote machine can only send files to *uucppublic.*
- The remote machine cannot request to receive files (REQUEST option).
- No files that are queued for the remote machine will be transferred during the current session (SENDFILES option).
- The only commands that can be executed are the defaults.

This entry alone is sufficient to start communications with remote machines, permitting files to be transferred only to the */usr/spool/uucppublic* directory.

Example 2. The next example is for remote machines that log in, but have fewer restrictions. The login and password corresponding to this entry should not be distributed to the general public; it is usually reserved for closely coupled systems where the **Systems** file information can be tightly controlled.

```
LOGNAME=uucpz REQUEST=yes SENDFILES=yes \
  READ=/  WRITE=/
```

This entry places the following permissions/restrictions on a machine that logs in as "**uucpz**:"

- Files can be requested from your computer (REQUEST option).
- Files can be transferred to any directory or any file that is writable by user "other." That is a file/directory that is writable by a local user with neither owner nor group permissions (WRITE option).
- Any files readable by user "other" can be requested (READ option).
- Any requests queued for the remote machine will be executed during the current session. These are files destined for the machine that has called in (SENDFILES option).
- The commands sent for execution on the local machine must be in the default set.

Example 3. The two previous examples showed entries that referred to remote machines when they log in to your computer. This example is an entry used when calling remote machines.

```
MACHINE=eagle:owl:hawk:raven \
  REQUEST=yes READ=/  WRITE=/
```

When calling any of the systems given in the MACHINE list, the following permissions prevail:

- The remote machine can both request and send files (REQUEST option).
- The source or destination of the files on the local machine can be anywhere in the file system (with read/write option).
- The only commands that will be executed for the remote machine are those in the default set.

Any site that is called that does not have its name in a MACHINE entry will have the default permissions as stated in Example 1, with the exception that files queued for that machine will be sent. (The SENDFILES option is only interpreted in the LOGNAME entry.)

Poll File

The *Poll* file (*/usr/lib/uucp/Poll*) contains information for polling specified Machines. Each entry in the *Poll* file contains the name of the remote machine to call, followed by a TAB character, and finally the hours the machine should be called. The entry:

```
eagle	0 4 8 12 16 20
```

will provide polling of machine eagle every 4 hours.

NOTE It should be understood that **uudemon.poll** does not actually perform the poll, it merely sets up a polling work (C.) file in the spool directory that will be seen by the scheduler, started by **uudemon.hour**. Refer to the discussion on **uudemon.poll**.

Maxuuxqts File

The *Maxuuxqts* (*/usr/lib/uucp/Maxuuxqts*) file contains an ASCII number to limit the number of simultaneous **uuxqt** programs running. This file is delivered with a default entry of 2. This may be changed to meet local needs. If there is a lot of traffic from **mail**, it may be advisable to increase the number of **uuxqt** programs that will run to reduce the time it takes for the mail to leave your system. However, keep in mind that the load on the system increases with the number of **uuxqt** programs running.

Maxuuscheds File

The *Maxuuscheds* (*/usr/lib/uucp/Maxuuscheds*) file contains an ASCII number to limit the number of simultaneous **uusched** programs running. Each **uusched** running will have one **uucico** associated with it; limiting the number will directly affect the load on the system. The limit should be less than the number of outgoing lines used by UUCP (a smaller number is often desirable). This file is delivered with a default entry of 2. Again, this may be changed to meet the needs of the local system. However, keep in mind that the load on the system increases with the number of **uusched** programs running.

remote.unknown

The **remote.unknown** program (**/usr/lib/uucp/remote.unknown**) is a shell file that is executed when a remote site that is not in the **Systems** file calls in to start a conversation. The shell script will append the name and time information to the file **/usr/spool/uucp/.Admin/Foreign**. Since it is a shell, it can be easily modified. For example, it can be set up to send mail to the administrator. The contents of this file, as delivered, is as follows:

```
FOREIGN=/usr/spool/uucp/.Admin/Foreign
echo "'date': call from system $1" >>$FOREIGN
```

If you want to permit machines that are not listed in your *Systems* file to communicate via Basic Networking, remove the execute permissions from the *remote.unknown* file. For example,

```
chmod 444 /usr/lib/uucp/remote.unknown
```

When **remote.unknown** is executable, your computer will hang up if a machine that is not in your *Systems* file calls in (to UUCP) on your system.

Administrative Tasks

There is a minimum amount of maintenance that must be applied to your computer to keep the files updated, to ensure that the network is running properly, and to track down line problems. When more than one remote machine is involved, the job becomes more difficult because there are more files to update and because users are much less patient when failures occur between machines that are under local control. The **uustat** program provides you with information about the latest attempts to contact various machines and the age and number of jobs in the queue for remote machines. The following sections describe the routine administrative tasks that must be performed by someone acting as the UUCP administrator or are automatically performed by the UUCP daemons (demons).

The biggest problem in a dialup network like UUCP is dealing with the backlog of jobs that cannot be transmitted to other machines. The following cleanup activities should be routinely performed.

Cleanup of Undeliverable Jobs

The **uustat** program should be invoked regularly to provide information about the status of connections to various machines and the size and age of the queued requests. The **uudemon.admin** shell should be started by **cron** at least once per day. This will send the administrator the current status. Of particular interest is the age (in days) of the oldest request in each queue, the number of times a failure has occurred when attempting to reach that machine, and the reason for failure. In addition, the age of the oldest execution request (X. file) is also given.

The **uudemon.cleanu** shell file is set up to remove any jobs that have been queued for several days and cannot be sent. Leftover data (*D.*) and work (C.) files are removed after 7 days, and execute (X.) files are removed after 2 days. It also provides feedback to the user indicating when jobs are not being accomplished and when these jobs are being deleted.

Cleanup of the Public Area

In order to keep the local file system from overflowing when files are sent to the public area, the **uudemon.cleanu** procedure is set up with a **find** command to remove any files that are older than 7 days and directories that are empty. This interval may need to be shortened by changing the **uudemon.cleanu** shell file if there is not sufficient space to devote to the public area.

Since the spool directory is very dynamic, it may grow large before transfers take place. Therefore, it is a good idea to reorganize its structure. The best way to do this on your computer is to use the **crontab** command to clean out the spool directory at a specified time.

First, specify the file you want to have the cleanup code in as follows:

```
crontab clean.wk [Enter]
```

The *clean.wk* file will contain the code for all files cleaned at a specified time (every Monday, for example), based on the time specified in the *crontab* file. You may already have entries in *clean.wk* which means you will also have the cleanup time specified. See **crontab**(1) in the *User's/System Administration Reference Manual* for additional information. If you wish to specify a new cleanup time, first, make a new file with the **crontab** command as above. Edit the *crontab* file to specify the time of cleanup. For example,

```
0 0 1 15 * 1
```

in the *crontab* file would indicate cleanup on the first and fifteenth of each month, as well as on every Monday. In the file you specified with the **crontab** command, enter the following code (the # sign lines are comment lines):

```
#       Clean up /usr/spool/uucp
#       Most cleanup is now done by uudemon.cleanu
#       so just copy out and back.
#
echo "UUCP SPOOL DIRECTORIES CLEANUP STARTED"
#
cd /usr/spool/uucp
mkdir ../nuucp
chown uucp ../nuucp
chgrp uucp ../nuucp
find . -print|cpio -pdml ../nuucp
cd ..
mv uucp ouucp
mv nuucp uucp
rm -rf ouucp
rm -f /usr/spool/locks/LCK*
#
#     Note:
#     Change the tty?? device to the
#     device you are using for UUCP.
#     For example change tty?? to tty01.
#
chown uucp /dev/tty??
chgrp uucp /dev/tty??
chmod 0644 /dev/tty??
chmod 0222 /dev/tty??
echo "UUCP SPOOL DIRECTORIES CLEANUP FINISHED"
```

Compaction of Log Files

This version of Basic Networking has individual log files for each machine and each program. For example, machine eagle has a log file for **uucico** requests and a log file for **uuxqt** execution requests. The **uulog** program gives the user access to the information in these files by machine name. These files are combined and stored in directory */usr/lib/uucp/.Old* whenever

uudemon.cleanu is executed. This shell script saves files that are 2 days old. The 2 days can be easily changed by changing the appropriate line in the **uudemon.cleanu** shell. If space is a problem, the administrator might consider reducing the number of days the files are kept.

Cleanup of sulog and cron log

The */usr/adm/sulog* and */usr/lib/cron/log* files are both indirectly related to UUCP transactions. The *sulog* file contains a history of the **su** command usage. Since each **uudemon** entry in the */usr/spool/cron/crontab/root* file uses the **su** command, the *sulog* could become rather large over a period of time. The *sulog* should be purged periodically to keep the file at a reasonable size.

Similarly, a history of all processes spawned by **/etc/cron** are recorded in */usr/lib/cron/log*. The cron *log* file will also become large over a period of time and should be purged periodically to limit its size.

UUCP and Cron

The **cron** daemon is a tool that proves to be very useful in the administration of UNIX systems. When the computer is in run state 2 (multiuser), **cron** scans the */usr/spool/cron/crontab/root* file every minute for entries that contain "work" scheduled to be executed at that time. It is recommended that the UUCP administrator make use of **cron** to aid in the administration of Basic Networking.

As delivered, Basic Networking contains four entries in the **root** *crontab* file. Each one of these entries executes shell scripts that are used for various administrative purposes. These shell scripts can be easily modified to meet the needs of your system.

uudemon.admin

The **uudemon.admin** shell script mails status information to the UUCP administrative login (**uucp**) using **uustat** commands with the -p and -q options. Refer to the **uustat** manual page for interpretation of these options.

The **uudemon.admin** shell script should be executed daily by an entry in the root crontab file. The default **root** crontab entry for **uudemon.admin** is as follows:

```
48 11,14, ** 1-5 /bin/su uucp -c "/usr/lib/uucp/uudemon.admin >
/dev/null 2>&1"
```

uudemon.cleanu

The **uudemon.cleanu** shell script cleans up the Basic Networking log files and directories. Archived log files are updated so that no log information over 3 days old is kept. Log files for individual machines are taken from the */usr/spool/uucp/.Log* directory, merged, and placed in the */usr/spool/uucp/.Old* directory along with the older log information. Files and directories that are no longer needed in the spool directories are removed. After cleanup is performed, the UUCP administrative login (**uucp**) is mailed a summary of the status information gathered during the current day.

The **uudemon.cleanu** shell script should be executed by an entry in the **root** *crontab* file. It can be run daily, weekly, or whenever, depending on the amount of UUCP traffic that enters and leaves your computer. The default root crontab entry for **uudemon.cleanu** is as follows:

```
45 23 * * * ulimit 5000; /bin/su uucp -c
"/usr/lib/uucp/uudemon.cleanu > /dev/null 2>&1"
```

If log files get very large, the ulimit may need to be increased.

uudemon.hour

The **uudemon.hour** shell script is used to call UUCP programs on an hourly basis. The **uusched** program is called to search the spool directory for work files (C.) that have not been processed and schedule these files for transfer to a remote machine. The **uuxqt** daemon is called to search the spool directory for execute files (X/C .) that have been transferred to your computer and were not processed at the time they were transferred.

The **uudemon.hour** shell script should be executed by an entry in the root crontab file. If the amount of traffic leaving and entering your computer is large, it may be started once or twice an hour. If it is small, it may be started once every 4 hours or so. The default root crontab entry for **uudemon.hour** is as follows:

```
26,56 * * * * /bin/su uucp -c
"/usr/lib/uucp/uudemon.hour > /dev/null"
```

uudemon.poll

The **uudemon.poll** shell script is used to poll the remote machines listed in the *Poll* file (*/usr/lib/uucp/Poll*). It creates work files (C.) for machines according to the entries listed in the *Poll* file. It should be set up to run once an hour just prior to **uudemon.hour** so that the work files will be present when **uudemon.hour** is called.

The **uudemon.poll** script should be executed by an entry in the root *crontab* file. The exact times it runs is dependent on the scheduling of **uudemon.hour**. The default root *crontab* entry for **uudemon.poll** is as follows:

```
40 * * * * /bin/su uucp -c "/usr/lib/uucp/uudemon.poll >
/dev/null"
```

Notice how **uudemon.poll** is scheduled to run 11 minutes before **uudemon.hour** runs.

Inittab Entries

The */etc/inittab* file contains information for the processes to be spawned on the computer devices, including the ports. Ports that are used by Basic Networking are normally bidirectional ports. Bidirectional ports can be used to receive incoming calls, as well as place outgoing calls. The **uugetty** program is used in place of **getty** for those bidirectional ports associated with Basic Networking. After **uucp** has been set up on a line, for example tty00, the next step is to enable a **uugetty** login on that line. This can be done by editing */etc/inittab* to add the new tty, and then telling **init** to re-read inittab. Do the following:

1. Edit */etc/inittab*, and add the line:

   ```
   :23:respawn:/usr/lib/uucp/uugetty -r tty00 1200
   ```

2. Tell **init** to re-read the */etc/inittab* file:

   ```
   init q
   ```

UUCP Logins and Passwords

There are two login IDs associated with Basic Networking; one is the UUCP administrative login **uucp**, and the other is an access login (**nuucp**) used by remote computers to access your computer. These logins should not be changed from their default settings of **uucp** and **nuucp**.

The **uucp** administrative login is the owner of all the UUCP object and spooled data files. The following is a sample entry in the */etc/passwd* file for the administrative login:

```
        uucp:zAvLCKp:5:1:UUCP.Admin:/usr/lib/uucp:
```

The **nuucp** access login allows remote machines to log in on your computer. The following is a sample entry in the */etc/passwd* file for the access login:

```
nuucp:zaaAA:6:1:UUCP.Admin:/usr/spool/uucppublic:
/usr/lib/uucp/uucico
```

Notice that the standard shell is not given to the **nuucp** login. The shell that **nuucp** receives is the **uucico** daemon that controls the conversation when a remote machine logs in to your machine.

The assigning of passwords for the **uucp** and **nuucp** logins is left up to the administrator. The passwords should be at least six to eight characters. Only the first eight characters of the passwords are significant. If the password for the access login is changed for security reasons, make certain that the remote machines that are a part of your network are properly notified of the change.

Remote File Sharing Administration

Remote File Sharing Administration

Chapter 9: Remote File Sharing Administration

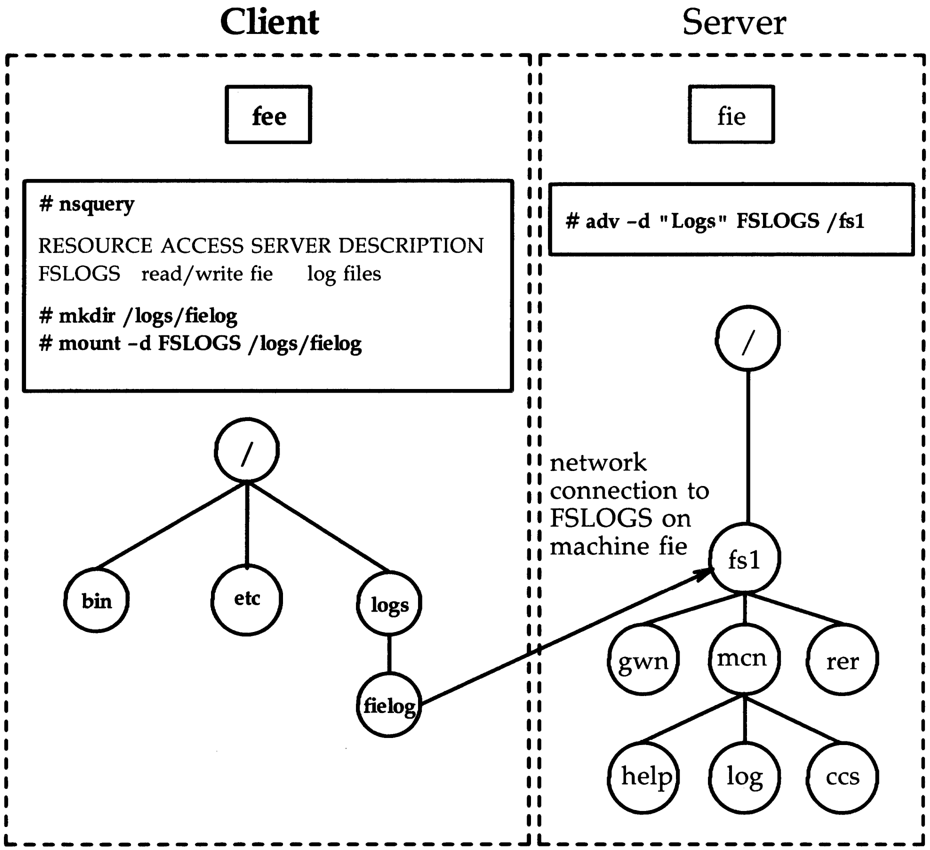
Overview of Remote File Sharing

Remote File Sharing (RFS) allows computers running UNIX System V to selectively share resources (directories containing files, subdirectories, devices, and/or named pipes) across a network. As an administrator of an computer on an RFS network, you can choose directories on your system you want to share and add them to a list of available resources on the network. From this list, you can choose resources on remote computers that you would like to use on your computer.

Resource Sharing

Sharing a resource on a Remote File Sharing system begins with a pathname to a UNIX system directory. If there is a directory you want to share, assign it a resource identifier and "advertise" it to other machines, using the **adv**(1M) command. The resource identifier is how other machines reference that directory. Computers that pass the security checks you have set up can then mount your resource as they would mount a file system locally. The **mount** command with the **-d** option is used for mounting remote resources.

Figure 9-1 shows how two computers can share resources. In this example, the administrator of a computer named **fie** on a Remote File Sharing system wants to share all files and directories under */fs1* on its file system tree. The administrator advertises */fs1* as a resource called FSLOGS.



```
# nsquery
RESOURCE ACCESS SERVER DESCRIPTION
FSLOGS   read/write fie    log files
# mkdir /logs/fielog
# mount -d FSLOGS /logs/fielog
```

```
# adv -d "Logs" FSLOGS /fs1
```

Figure 9-1: Example—Sharing Resources

Another machine in **fie**'s domain is called **fee**. The administrator from **fee** uses the **nsquery** command to see that FSLOGS is available on **fie**. **fee**'s administrator then creates a directory called */logs/fielog* on **fee** (**mkdir** command) and **mounts** FSLOGS on */logs/fielog*.

Files or subdirectories from */fs1* are now accessible to users on **fee**. Users can **cd** to the remote directory, list the contents, and run a remote program locally. If the resource contained the */dev* directory, users could direct output to a remote device as though the device were on the local machine.

Domains

Each machine on a Remote File Sharing network must be assigned to a domain. The main reasons for domains are to simplify name service and provide a focal point for security of a group of machines.

Domain names act like telephone area codes. You can address all computers and resources in your domain directly. For outside domains, simply attach the domain name to the node name or resource identifier. This becomes increasingly valuable as RFS networks expand.

Name Service

Each domain must be assigned a primary and zero or more secondary domain name servers. These machines can share resources, like any other computer in the domain, but they have some special responsibilities.

Primary

The main duty of the primary domain name server is to keep track of all computers and resources within the domain it serves. It ensures that all resource identifiers and machine node names are unique within the domain.

A required task of the primary is to add each computer to the Domain Member List and assign its RFS password.

A list of advertised resources are automatically stored on the primary, so any computer can see a complete list of available resources for the domain. Also, when a computer advertises a resource, it registers its network address with the primary. Therefore, when a computer tries to mount another machine's resources, the primary can tell the computer where the resource can be found on the network.

An optional function of the primary is to gather lists of each computer's users (*/etc/passwd*) and groups (*/etc/group*). Each computer in the domain can then use these lists to specifically define the permissions each machine's users will have to its resources.

A primary can also gather names and network addresses of other domains' name servers. Once the primary knows another domain name server's address, machines in its domain have the potential to access resources from any machine in the other domain.

Secondaries

If the primary fails, domain name service functions are automatically assumed by one of the secondary domain name servers. The secondary is intended to take over temporarily, until the primary comes back up.

While the secondary will have information needed to run the domain name server, domain information should not be modified on the secondary. As soon as the primary comes back up, the secondary should be instructed to pass name server responsibility back to the primary (**rfadmin -p** command). Then the primary's administrator can change the domain member list, edit the *rfmaster* file, or gather optional user and group information again on the primary.

Transport Provider

The transport provider provides the pathway used by Remote File Sharing to communicate with other machines. The term transport provider is used to refer to the physical network that connects the machines and the software needed to send messages across the network.

Remote File Sharing can communicate using any transport provider that is compatible with the AT&T Transport Interface Specification. The STARLAN network is one transport provider that can be used with RFS.

Although the transport provider is not considered part of the RFS package, RFS will not work if the transport provider is not functioning properly. Also, some information needed to configure RFS varies from one transport provider to another. For example, network addresses of the primary and secondaries and the network specification to identify the transport provider to RFS are dependent on the particular transport provider used.

The following sections describe transport provider information that relates to RFS administration: Network Listener, Network Specification, and Network Addresses.

Network Listener

The network listener is part of the Networking Support Utilities package. Essentially, the listener's function is to wait for requests from the network. A call coming in from the network will request a particular service code. The service code will tell the listener to direct the call to a particular process.

Service code **105** is used to request RFS services. If all software installation was done as noted in the *Remote File Sharing Release Notes*, the RFS service code should be automatically configured for the listener of every transport provider you installed. Otherwise, you will need to use the **nlsadmin**(1M) command to manually configure the listener. (See the "Setting Up RFS" section of this chapter.)

Network Specification

Since you could have several transport providers on one computer, you must tell RFS which transport provider will handle RFS on your machine. The network specification is the name you will use when you initially configure RFS to indicate its transport provider. [The STARLAN network, for example, uses **starlan** as its network specification. This tells RFS that */dev/net/nav/md000* is the device representing the transport provider to use.]

Network Addresses

When Remote File Sharing is started on a machine, the machine tries to contact its domain's primary name server. In order to do that, the machine must know the primary's network address.

The form of the network address varies according to the transport provider used. The STARLAN network convention for network addressing is to use a machine's node name and append the string **.serve** to create the network address. For example, the network address for a machine whose node name is **charlie** would be **charlie.serve** on a STARLAN network.

Security

Remote File Sharing provides several mechanisms for ensuring the security of your resources. Some of these mechanisms, however, require diligence to set up and maintain. This is especially true if the machines, resources, and users are constantly changing on the network.

As a system administrator, you can maintain strict control of your resources. No files, directories, or devices in an unshared file system can be accessed by other computers. Standard UNIX system file security measures can be used in combination with special RFS facilities to protect your resources.

Direct access to your computer is controlled because local users still have to log in as they always have. As for remote accessibility, you can set up security to allow only certain remote computers to access your resources.

The major mechanisms in RFS for protecting your resources are described in the following sections: "Verify Computers," "Restrict Resources," and "Map IDs."

Verify Computers

When a remote computer tries to mount a resource from your computer, and no other resources are mounted, it tries to set up a connection (virtual circuit) across the network to your machine. Once this virtual circuit is set up, the remote machine can mount any resource you have made available to it. This virtual circuit is closed when the last resource is unmounted.

Before this virtual circuit is created, you can verify that the computer is the one it claims to be by checking its RFS password. The following text describes what happens when verification is and is not used.

- No verify: any computer can connect

  If the computer is listed in the *domain/passwd* file, your machine will check its password. Otherwise, your computer will accept it as the machine it claims to be.

- Verify: some computers can connect

  If you use the RFS verification feature, you can make sure that only specific machines can use any of your resources. Those machines must be listed in the proper *domain/passwd* file and must match the password you have for them. (*domain* is the domain name of the requesting machine.) You can tailor this file if you only want a subset of machines to be allowed to connect. (A description of how to use this feature is contained in the "Setting Up RFS" section of this chapter.)

Restrict Resources

Once a remote computer has established a connection to your computer, the resources it can mount from your machine depend on how you advertised each resource. These are your choices:

- Any machine can mount.

  You may have advertised the resource so that any machine that can connect to your machine can mount it.

- Some machines can mount.

  You restricted access to the resource to certain machines. The remote computer trying to mount it must be one of those machines.

You also may have advertised the resource as read-only. In that case, the remote computer can only mount the resource read-only instead of read/write (default).

Map IDs

Remote users' permissions can be defined to provide another layer of security for a mounted resource. Remote users and groups can be mapped into your computer's user and group list to set permissions they will have to your resources.

You can set these mapping rules on a global or per-machine basis. The global rules set user and group permissions for all remote machines that do not have explicit mapping rules.

Here are the ways you can map remote machines' users into your machine. These rules apply to both global and per-machine mapping.

- No mapping

  If you don't set any special mapping for any remote computer, all users will be mapped into your machine as a "special guest" user ID/group ID. This is the easiest approach because you don't need to keep any records for the remote machine, create rules files, or run the **idload** command.

- Default mapping

  You can set default mapping so that all remote users are mapped into one of these permissions:

  - The local user ID number that matches each remote user's ID (**default transparent**)
  - A single local ID number
  - A single local ID name
  - The local user name that matches each remote user's name (**map all**).

  Group permissions can be mapped in the same way. Users and groups are mapped independently. If there are exceptions to the default mapping, you can **exclude** certain users and groups so they only have special guest permissions (for example, **exclude 0**).

- Specific mapping

  You can map any user or group from any remote machine into a specific user or group on your machine. You can do this by user name or numeric ID.

Using these mapping techniques and standard methods for setting file permissions, you can keep strict controls over your resources, even after they are remotely mounted. (See the "Mapping Remote Users" section of this chapter for more details.)

RFS Features

Some Remote File Sharing features that reflect improvements over other distributed file systems are described in the following paragraphs.

Compatibility Once you mount a remote resource on your system, it will look to your users as though it is part of the local system. You will be able to use most standard UNIX system features on the resource. Standard commands and system calls, as well as features like File and Record Locking, work the same on remote resources as they do locally. Applications should be able to work on remote resources without modification.

Flexibility

Since you can mount a remote resource on any directory on your system, you have a lot of freedom to set up your computer's view of the world. You do not have to open up all your files to every machine on the network. Likewise, you do not have to make all files on the network available to your computer's users.

Performance (Client Caching)

The client caching feature of RFS provides substantial performance improvements over non-caching systems by reducing the number of times data must be read across the network. Client refers to the computer that is using a remote resource, while caching refers to the client's ability to store data in local buffer pools.

The first time a client process reads a block of data from a remote resource, it is placed in local buffer pools. Subsequent client processes reading a server file can avoid network access by finding the data already present in local buffers. This generally causes a large reduction in network messages, resulting in improved performance.

In order for client caching to work simply and reliably, the following features were built into it.

- Cache consistency. Checking mechanisms are used to ensure that the cache buffers accurately reflect the contents of the remote file the user is accessing.
- Transparency. The only difference users should see between caching and non-caching systems is improved response time. RFS-based applications do not have to be changed to run on a Remote File Sharing system that caches remote data.
- Administration. By default, client caching is on. However, options are available to turn off caching for an entire system or for a particular resource. (You would probably only do this if you have an

application that does its own network buffering.) There are also some tunable parameters available to fine tune your system according to the way you use RFS. (See the ''Monitoring'' and ''Parameter Tuning'' sections of this chapter for more information.)

Setting Up RFS

In most cases, you will not need the set of tasks described in this section because the basic RFS configuration and reconfiguration can be handled using the commands described in Procedure 9.1. These tasks are for those who want to go deeper into the workings of RFS or are having problems with particular components.

These tasks are run from the shell. They should be run initially in the order described.

Once these tasks are completed, go to the "Starting/Stopping RFS" section for information on starting RFS.

Prerequisites

Before you begin setting up RFS, the following must be installed and running: UNIX System V Release 3.1 (or later) software, Remote File Sharing Utilities, Networking Support Utilities, and transport provider software. (See the *Remote File Sharing Release Notes* and the transport provider manuals that accompany the product for installation instructions.)

You must also log in as **root**.

Set Node Name

CAUTION Changing the node name of your computer requires careful coordination with all machines that communicate with yours using Remote File Sharing or other communications packages that rely on node name.

Check to see if your computer's node name is set to the name you want (**uname -n**). If it's not, set it by typing:

```
uname -S nodename
```

You will be asked to type in your computer's node name. A node name that is valid for Remote File Sharing can consist of up to eight characters of letters (uppercase and lowercase), digits, hyphens (-), and underscores (_). Some networks, such as the STARLAN network, require that every node name in the network be different. Remote File Sharing, however, only requires that every node name in a domain be different.

Set Up Network Listener

If you have installed the Networking Support Utilities, the AT&T implementation of the STARLAN network, and Remote File Sharing in the order described in the Chapter 2, Software Installation, you can skip this task. The listener will already be installed and set up to run automatically and Remote File Sharing will be listed as an available service.

If you are using another transport provider, or suspect that your STARLAN network listener is set up improperly, this task will show how to manually set up the listener. In the following example the STARLAN network is used. To set up the listener for other networks compatible with the AT&T Transport Interface, you should replace **starlan** with the name of the network (network specification) you are installing. [For more details, see the **nlsadmin**(1M) manual page.]

To determine if the listener is properly installed and set up for use by RFS, type the following:

```
nlsadmin -v starlan
```

If service code 105 is listed, then the listener is configured to be used for Remote File Sharing.

Run the following commands if the listener is not properly set up. If you run any of these commands and they have already been run, you will receive a message telling you so. This will not harm your listener configuration. Type:

```
nlsadmin -i starlan
```

to initialize the files needed for the listener process for the network specified, in this case **starlan**.

Next, type:

```
nlsadmin -a 105 -c /usr/net/servers/rfs/rfsetup -y "rfsetup" starlan
```

to add the Remote File Sharing service (**rfsetup**) to the list of services available to the **starlan** listener.

Use the following command line to report the status of the **starlan** listener process installed on this machine (ACTIVE or INACTIVE):

```
nlsadmin -x
```

Next, type:

```
nlsadmin -l "nodename.serve" -t "nodename" starlan
```

to register the network addresses of your machine. The listener will listen for requests for these addresses on the network. Only the **-l** address is required by Remote File Sharing. The **-t** address is used only for terminal services and may not be needed on all networks.

To start the listener, type:

```
nlsadmin -S starlan
```

Normally, it will be started automatically when your machine enters multiuser mode (**init 2**).

Set the Domain Name

Set the domain name by typing:

```
dname -D domain
```

where *domain* is replaced by the domain of which your machine will be a member. The domain name must:

- Contain no more than 14 characters
- Consist of any combination of letters (uppercase or lowercase), digits, hyphens, and underscores
- Be different from the name of any other domain used on the network if there is more than one domain on your network.

You can check the current domain name by typing:

```
dname
```

Set the Transport Provider

To identify the network, you must tell Remote File Sharing the network (transport provider) it should use. (In our example, this is **starlan** for the STARLAN network.)

```
dname -N starlan
```

This command indicates the device, relative to the */dev* directory, that is used for the transport provider.

Create rfmaster File

The *rfmaster* file should only be created manually on the primary. If your machine is not the primary, you should skip this task; the *rfmaster* file for your domain will automatically be placed on your machine the first time you start RFS (**rfstart -p** *primary_addr*).

If you are on the primary, you can create an *rfmaster* file in the */usr/nserve* directory using any standard file editor. The contents of this file will define:

- The primary name server for your domain
- Secondary name servers for your domain
- Network addresses for each of these machines.

(See the section on "Multiple Domain Name Service" in this chapter for a description of other information you may want to put into the *rfmaster* file.)

Here is an example of an *rfmaster* file for a domain called **peanuts**, whose primary and secondary name servers' node names are **charlie**, **linus**, and **lucy**. Adding each machine's domain name (**peanuts**) to its node name, separated by a period, forms its full Remote File Sharing machine name. Each line of the example translates as follows.

- For domain **peanuts** the primary is **peanuts.charlie**.
- For domain **peanuts** a secondary is **peanuts.linus**.
- For domain **peanuts** another secondary is **peanuts.lucy**.

- For computer **peanuts.charlie** the network address is **charlie.serve**.
- For computer **peanuts.linus** the network address is **linus.serve**.
- For computer **peanuts.lucy** the network address is **lucy.serve**.

(The addresses shown are an example of STARLAN Network addresses. These addresses should be in the form nodename.serve.)

```
peanuts         p           peanuts.charlie
peanuts         s           peanuts.linus
peanuts         s           peanuts.lucy
peanuts.charlie a           charlie.serve
peanuts.linus   a           linus.serve
peanuts.lucy    a           lucy.serve
```

Each line in the example is an entry. The second field is the *Type* field, which indicates whether the entry defines a primary name server (**p**), secondary name server (**s**), or the network address (**a**) for one of these name servers. Here is the information needed for the first field, *Name,* and the third field, *Rdata,* for each type of entry.

p Primary entry. *Name* is the domain name. *Rdata* is the full RFS machine name of the domain's primary name server (*domain.nodename*).

s Secondary entry. *Name* is the domain name. *Rdata* is the full RFS machine name of the domain's secondary name server (*domain.nodename*).

a Address entry. *Name* is the full RFS machine name (*domain.nodename*) of a name server computer. *Rdata* is the network address of the computer. The manuals that come with your network should describe how to find a computer's network address.

Here are some special considerations when creating the file.

- Fields in each entry must be separated by one blank or one tab.
- An entry can extend beyond one line if you enter a back slash, then a carriage return to continue to the second line.

- This file should be write-protected from all but *root,* but all read permissions should be enabled (644 permissions).
- If you start a line with the # character in column 1, the entire line will be treated as a comment.

Add/Delete Domain Members

If your computer is the current primary name server for the domain, you must add each computer to the domain member list. (If a secondary has temporarily taken over, the secondary must pass name server responsibility back to the primary using the **rfadmin -p** command.) To add members, use the following command:

```
# rfadmin -a domain.nodename
Enter password for nodename:
Re-enter password for nodename:
```

where *nodename* is replaced by the node name of the computer you want to add to your *domain.* (The two names must be connected by a period.)

You will be prompted for an initial password, which will be stored in the */usr/nserve/auth.info/domain/passwd* file for the *domain.* When the computer you added starts Remote File Sharing, the computer's administrator must enter this password. You can simply type a <**CR**> for a null password. Otherwise, the password must conform to the same criteria used with the **passwd**(1) command. Repeat this command for each computer you want to add to the domain.

NOTE

Adding a primary and secondary to the *rfmaster* file does not automatically add them to the domain. You must do this procedure for each of those machines.

You can also use the **rfadmin** command to delete members from the domain member list, as follows:

```
rfadmin -r domain.nodename
```

Remote Computer Verification

NOTE — This procedure assumes you are starting RFS from the shell using **rfstart** with the **v** option or **init 3**.

When you start Remote File Sharing, you can indicate that all remote machine passwords be verified when they try to use your computer's resources. **rfstart** is the command that is run automatically when you go into Remote File Sharing state (**init 3**).

If you use **rfstart** with the **-v** option, any machine that tries to mount your resources must match a name and password you have in the *passwd* file in the */usr/nserve/auth.info/domain* directory on your machine, where *domain* is replaced by the name of the remote computer's domain. If the remote computer is not listed in this *domain/passwd* file, if it is listed and the password doesn't match, or if no *domain/passwd* file exists, the remote mount will fail. (This file is automatically on the primary, but it must be added to other machines, as described in this procedure, to use verification.)

If you do not use the **-v** option, the following validation will occur. If a *domain/passwd* exists for the remote computer's domain on your computer and the remote computer is listed, but the password does not match, a mount request will fail. If the computer is not listed in the file or if the *domain/passwd* file does not exist, the computer will be allowed to mount your resources without validation. (Of course, a remote mount could still fail if the resource was advertised to a limited subset of machines or was advertised read-only and the machine tried to mount it read/write.)

The following steps describe how verification is set up.

Step 1: Obtain *domain/passwd* file(s). The */usr/nserve/auth.info/domain* directory on the primary will contain a file called *passwd.* (*domain* is replaced by the domain name.) This file will have the name and encrypted password for each machine in the domain.

You must make the *domain/passwd* file, plus the *domain/passwd* file for any outside domains containing machines you want to verify, accessible to your machine in one of the following ways:

Step 1A: Place a copy of this file(s) in the same directory on your machine. The *passwd* file for each domain must be in the appropriate *domain* subdirectory.

or

Step 1B: Have the primary for each domain advertise the */usr/nserve/auth.info/domain* directory; then have it automatically mounted in the same location on your computer. This way you can automatically pick up any changes in machines or passwords. (See the description of */etc/fstab* in the "Automatic Remote Mounts" section of this chapter for information on setting up automatic mounts.)

Step 2: **rfstart -v**. You must edit the */etc/rc3.d/S21rfs* file to automatically run **rfstart** with the **-v** option. You will add the **-v** to about line 61 of this file, after the **rfstart** command, as shown in the following example.

```
'rfstart')
  trap 'rm -f /usr/tmp/rfs$$;exit' 0 1 2 3 15
  stat=1
  retries=0
  while [ ${stat} -eq 1 ]
  do
    /usr/bin/rfstart -v </dev/console >/dev/console 2>/usr/tmp/rfs$$
    stat=$?
    case ${stat} in
```

Step 3: If you want to verify only a limited subset of these computers, you must use manually edited versions of the *domain/passwd* files, removing any computers you want to prevent using your resources. (You cannot edit this file if you are a primary or secondary name server or if you have mounted the file from the primary.)

Resource Sharing With Other Domains

For computers in your domain to share resources with computers in other domains on your network, you must do the following:

Step 1: Find out:

- The primary name server for each domain
- The secondary name server(s) for each domain
- The network address for each of the above name servers.

Step 2: You must see that the information in Step 1 is added to your domain's */usr/nserve/rfmaster* file on the primary. See the description of the *rfmaster*(4) file in the format of the *rfmaster* file. The following example shows the information added to contact a domain called **docs**.

```
docs           p          docs.big
docs           s          docs.little
docs.big       a          big.serve
docs.little    a          little.serve
```

Step 3: Stop Remote File Sharing on the primary (**rfstop** or **init** 2).

Step 4: Restart Remote File Sharing on the primary (**rfstart** or **init** 3). Make sure start-up has completed before going to the next step.

Step 5: If a secondary machine took over name service when the primary was stopped, pass name service responsibilities back to the primary by typing the following from the secondary:

rfadmin -p

Step 6: Mount resources from an outside domain. Once the name server machines have picked up the new domain names, you can mount a resource from a remote domain on your own machine. You would use the same method of mounting a resource from an outside domain as you would to mount a resource from your domain, with one exception. When you specify the resource to be mounted, you must prepend the domain name to the resource identifier. For example, the command:

mount -d docs.INFO /usr/info

could be used to mount a resource called INFO that is advertised in domain **docs** with read/write permissions.

Multiple Domain Name Service

Once you have defined a set of primary and secondary name servers to serve a domain, that set of machines may also be name servers for another domain on the same network. The following procedure describes how this can be configured:

Step 1: Edit *rfmaster* file. You must add the information on the new domain's name servers to the *rfmaster* file on the primary. The following is an example of two sets of name servers that serve domains called **docs** and **peanuts**.

```
docs          p          docs.big
docs          s          docs.little
docs.big      a          big.serve
docs.little   a          little.serve
peanuts       p          docs.big
peanuts       s          docs.little
```

Step 2: Stop and restart RFS. You must stop all machines served by the primary (**rfstop** or **init 2**). You must then restart the primary (**rfstart** or **init 3**). Then start each machine on the system, starting machines that previously had other machines as domain name servers with the **rfstart -p** *address*, where *address* is replaced by the network address of the new primary domain name server. This will ensure that the new information is picked up by each machine.

Complex User ID/Group ID Mapping

ID mapping lets you control the access remote users will have to files and directories that make up your shared resources. This feature lets you assign each remote user the permissions of one of your local users (listed in */etc/passwd*) or the permissions of a special "guest ID," with respect to your shared resources. The "guest ID" will never overlap with any of your local users. The same mechanism can be used to define group permissions (listed in */etc/group*).

Use this procedure as a tutorial for ID mapping and as a procedure for setting up mapping. If you have questions about particular mapping components, refer to the "Mapping Remote Users" section of this chapter.

When Not to Map

In most cases, ID mapping is not necessary. If you never set up mapping, all users will be mapped into a single special guest ID. This special guest ID is represented by an ID number that is one higher than the maximum allowed for your system. By default, the maximum number of users and groups on a system is 60000, so the special guest is ID number 60001.

No mapping, or the default mapping, provides the maximum security for your shared resources. When a remote user lists the permissions of your files (**ls -l**), all files will be owned by 60001 or 60002. The 60001 means the file was created by a remote user and, therefore, is owned by every remote user that can access your resource. 60002 means the file was created by one of your local users and, therefore, The remote users can only access the file if the "other" permissions are set (the last 3 bits of the **rwx** permissions).

When to Map

Using mapping increases the power and flexibility of RFS. The following are some reasons you may want to use mapping:

- Special permissions.

  You may want to map some or all remote users into particular local users' permissions. For example, if you are the administrator of several machines, you may want to map all **root** logins together across the machines. Therefore, you would be able to modify any remote resources mounted on any machine you are working from.

- Transparent mapping.

  If you set up a group of computers to have the same */etc/passwd* and */etc/group* files, mapping transparently can be a very powerful technique. When a user creates a file, the user will maintain sole ownership of the file, whether or not the file resides on a remote resource.

  With transparent mapping, you could share many resources that require a consistent view of user ownership. For example, you could share your */usr/mail* directory, mount it on */usr/mail* on other computers, and have one mail directory for the entire set of machines. The basic concept is that you can avoid duplication of many files and directories while maintaining consistent user permissions.

- Mapping by machine.

  You may want to map users from one machine differently than users from another machine. For example, you may want to map all users from one machine into user ID **600**, from another machine into **700**, and from a third into **800**. In that way you could monitor which remote machine's users were creating files within your resources.

Mapping Tools and Files

The result of this procedure is "mapping translation tables." These tables will be used by your system to process requests from remote users for access to your resources that are mounted on their computers.

The command used to create the translation tables is **idload**. When **idload** is run with no options, it does the following:

- Reads the rules files to determine how you want to set up the mapping
- Reads the *passwd* and *group* files on your computer, and copies of those files from other computers, if needed
- Creates translation tables.

There are two options to **idload** you also may want to use when setting up translation tables.

idload -n Before you run **idload** with no options, the **-n** option lets you do a trial run without actually changing the mapping tables. The result is a listing at your terminal of the tables you would create if you ran **idload** with no options.

idload -k After you run **idload** with no options, the **-k** option lets you read the mapping that is currently in effect on your computer.

Figure 9-2 illustrates the components described in the previous paragraphs.

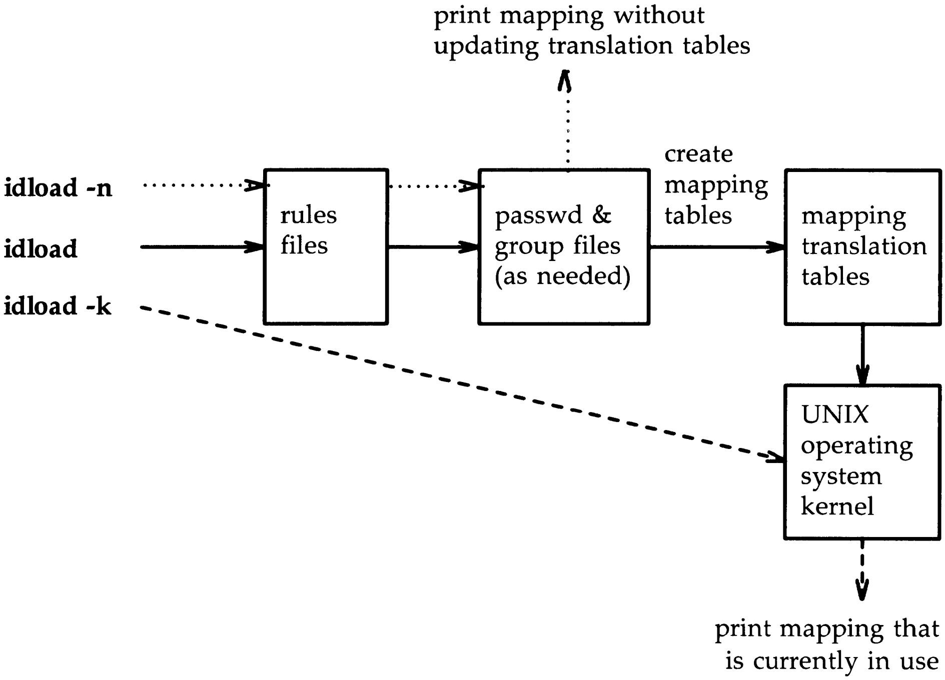

print mapping without updating translation tables
idload -n
idload
idload -k
rules files
passwd & group files (as needed)
create mapping tables
mapping translation tables
UNIX operating system kernel
print mapping that is currently in use

Figure 9-2: ID Mapping Components

Figure 9-3 illustrates the files that are involved in setting up ID mapping.

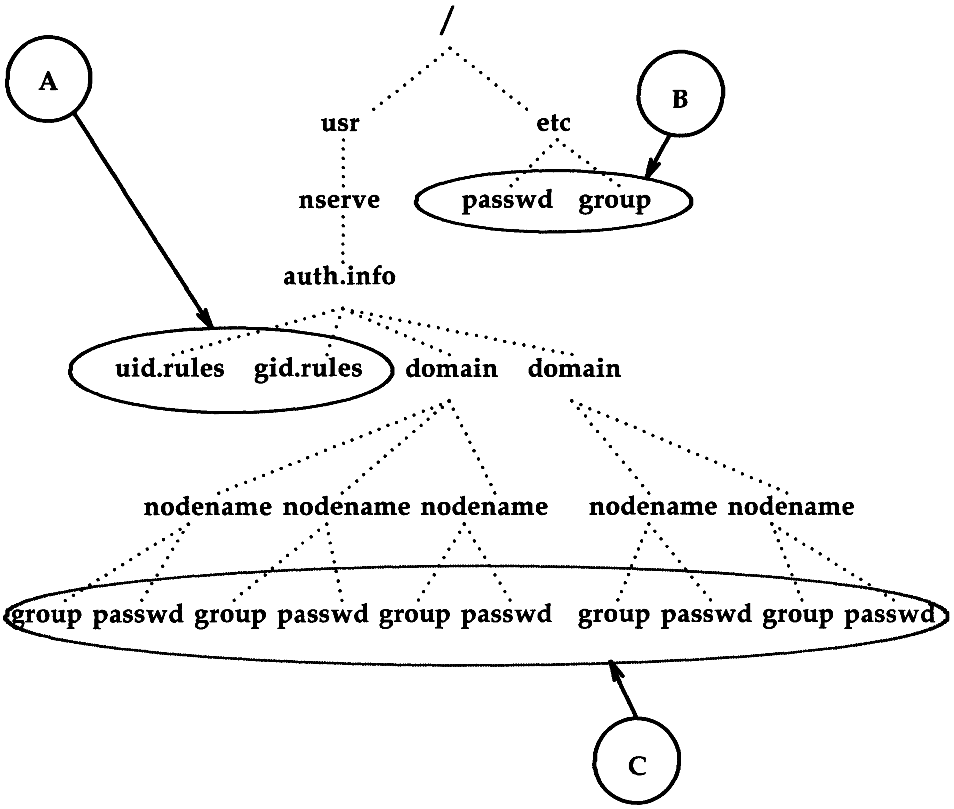

/
A
usr
etc
B
nserve
passwd group
auth.info
uid.rules gid.rules
domain domain
nodename nodename nodename nodename nodename
group passwd group passwd group passwd group passwd group passwd
C

Figure 9-3: ID Mapping Files

The files used for ID mapping are divided into the following three groups, as shown in Figure 9-3.

A. Rules Files
The *uid.rules* and *gid.rules* files are located in the */usr/nserve/auth.info* directory. The information you add to these files tells the **idload** command how to create the mapping tables.

B. Local *passwd* and *group* Files
The */etc/passwd* and */etc/group* files contain lists of the local users on your system. Though you don't modify these files to do ID mapping, you will be interested in the information that is in these files. The first field in each line of your *passwd* and *group* files contains local user and group names, respectively. The third field contains the related ID number. If you map by local name in the rules files, these files are read to translate the names into numbers.

C. Remote *passwd* and *group* Files
Because mapping translation tables are sets of numbers, if you want to map a remote user by name you must have a copy of the *passwd* and/or *group* files for the remote user's machine. These files should be placed in the */usr/nserve/auth.info/domain/nodename* directories, where *domain* and *nodename* are replaced by the remote computer's domain and node names, respectively.

Step 1: Create uid.rules File

The following steps describe how to create the rules used to map remote users.

Using any standard file editor (**ed** or **vi**, for example), create or edit the *uid.rules* file in the */usr/nserve/auth.info* directory. Steps 1A-1D will help you set up a **global** block of mapping information; Steps 1E-1H are for **host** blocks of mapping information. The **global** block defines the permissions that will apply to the users on all computers that do not have specific mapping. Note that all lines within a **global** block are optional.

Step 1A: Add the **global** line. (Add only this line if you want to define a block of global information.) The global block of information must begin with the following keyword on a line by itself:

global

Step 1B: Add a **default** line. (Add only this line if you want to define default information for a global block.) Following the **global** line, you can choose the default permissions that will apply to users from all machines that are not specifically mapped. If this line is not used, the system assumes **default 60001**. (In most cases, **default 60001** is fine.) The two types of default lines are illustrated below.

The line **default transparent** means that each user will have the permissions of the user with the same ID number on your system. (This strategy is most valuable when the */etc/passwd* files are identical on the two machines.) In the line **default** *local,* the word *local* can be replaced by a local ID number or ID name. This means that any users that are not specifically mapped will have the permissions of a particular user on your system. (Use only one **default** line in a **global** block.)

default transparent
or
default *local*

Step 1C: Add **exclude** line(s). (Add only this line(s) if you want to exclude certain users.) **exclude** lines let you exclude certain users from having the permissions defined in the default line. For example, if you used **default transparent**, you may want to use **exclude 0** to make sure that the **root** user doesn't have permission to modify the restricted files owned by **root** in your resources. The two types of exclude lines are illustrated below.

In **exclude** *remoteid, remoteid* is replaced by a remote user ID number. The remote user would then have the permissions of the guest user (UID 60001) to your resources.

The **exclude** *remoteid–remoteid* line lets you specify a range of remote IDs to exclude. For example, **exclude 0-100** could be used to exclude all administrative logins from your default mapping.

exclude *remoteid*
or
exclude *remoteid–remoteid*

Step 1D: Add **map** line(s). (Add only this line if you want to map specific users from global machines.) **map** lines let you take specific remote user IDs and map them into the permissions of one of your local users. The two types of map lines are illustrated below.

In **map** *remoteid:local, remoteid* is replaced by a remote user ID number and *local* is replaced by a local user's name or ID number. For example, the line **map 20:root** would map the remote user with ID number 20 into your machine's **root** permissions (UID **0**). The line **map** *remoteid* says give the remote user the permissions of the user with the same ID number on the local system. For example, **map 0** would give **root** from a remote machine the same permissions as **root** on your machine.

> **map** *remoteid:local*
> or
> **map** *remoteid*

Once **global** mapping is done, you may want to add **host** mapping information to the *uid.rules* file. A **host** block defines the permissions that will apply to the users on particular remote machines. You can have one **host** block for each remote machine you want to map specifically. Note that all lines within a **host** block are optional.

Step 1E: Add a **host** line. (Add only this line if you want to define a block of host information.) The host block of information must begin with the following keyword on a line by itself:

> **host** *domain.nodename*

where *domain* is replaced by the remote machine's domain name and *nodename* is replaced by the machine's node name.

Step 1F: Add a **default** line. (Add only this line if you want to define default information for a host block.) Following the **host** line, you can choose the default permissions that will apply to all users on the remote machine that are not specifically mapped or excluded. If this line is not used, the system assumes **default 60001**. (In most cases, **default 60001** is fine.) The two types of default lines are illustrated below.

The line **default transparent** means that each user will have the permissions of the user with the same ID number on your system. (This strategy is most valuable when the */etc/passwd* files are identical on the two machines.) In the line **default** *local,* the word *local* can be replaced by a local ID number or ID name. This means that any users that are not specifically mapped will have the permissions of a particular user on your system.

> **default transparent**
> or
> **default** *local*

Step 1G: Add **exclude** line(s). (Add only this line(s) if you want to exclude certain users from default permissions.) **exclude** lines let you exclude certain users from having the permissions defined in the default line. For example, if you used **default transparent**, you may want to use **exclude 0** to make sure that the **root** user doesn't have permission to modify the restricted files owned by **root** in your resources. The two types of default lines are illustrated below.

In **exclude** *remote, remote* is replaced by a remote user name or UID number. The *remote* user would then have the permissions of the guest user (UID 60001) to your resources.

The **exclude** *remoteid-remoteid* line lets you specify a range of remote IDs to exclude. For example, **exclude 0-100** could be used to exclude all administrative logins from your default mapping.

exclude *remote*
or
exclude *remoteid-remoteid*

Step 1H: Add **map** line(s). (Add only this line(s) if you want to map particular users.) **map** lines let you map specific remote users from specific remote machines into the permissions of one of your local users. The two types of map lines are illustrated below.

The **map all** line says to map all user names into the permissions of the users with the same names on your system. In **map** *remote:local, remote* is replaced by a remote user ID name or number and *local* is replaced by a local user's name or ID number. For example, the line **map 20:root** would map the remote user with ID number 20 into your machine's **root** permissions (UID **0**). The line **map** *remoteid* says give the remote user the permissions of the user with the same ID number on the local system. For example, **map 0** would give **root** from a remote machine the same permissions as **root** on your machine.

map all
or
map *remote:local*
or
map *remote*

Repeat steps 1E-1H for each specific computer whose users you want to map.

THE **uid.rules** FILE IS NOW COMPLETE!

Figure 9-4 is an example of what your rules file may look like.

```
global
default 1000
exclude 0

host peanuts.snoopy
default transparent
exclude 0

host peanuts.linus
default 60001
map 0:100
```

Figure 9-4: Example **uid.rules** File

Step 2: Create gid.rules File

The following steps describe how to create the rules used to map remote groups.

Create the *gid.rules* file. Using any standard file editor (**ed** or **vi** for example), edit the *gid.rules* file in the */usr/nserve/auth.info* directory. The *gid.rules* file follows the same format as the *uid.rules* file. Therefore, you can use Steps 1A through 1H to set up the *gid.rules* file, replacing any references to users with references to groups.

NOTE

If you create a *uid.rules* file you should also create a *gid.rules* file. Although **idload** will still work without the *gid.rules* file (**idload** will use defaults for mapping groups), a warning message will be produced.

Step 3: Add passwd and group Files

If, when you edited the *uid.rules* and *gid.rules* files, you referenced any remote users by name, you must have copies of the *passwd* file from the remote users' computers in the */usr/nserve/auth.info/domain/nodename* directories on your machine. The same is true of the *group* file for groups referenced by name. (Note that **map all** maps by name.)

The best way to obtain these files is as follows:

Step 3A: Obtain files. Have each machine whose users you want to map by name send you its */etc/passwd* and */etc/group* files using any standard file transfer method (such as **uucp**). (The information in the password field can be removed from each entry, if you prefer. The password is made up of the characters between the first and second colon in each entry.)

Step 3B: Create directories. You must create a separate directory on your machine for each computer whose users and groups you map by name. Each directory must be created using the path */usr/nserve/auth.info/domain/nodename,* where *domain* is replaced by the remote machine's domain name and *nodename* is replaced by the remote machine's node name. For example, you create the following directory for a machine called **linus** in domain **peanuts**:

/usr/nserve/auth.info/peanuts/linus

Step 3C: Place files. Place the remote machines' *passwd* and *group* files in the directory you created in the previous step.

Step 4: Run idload

Step 4A: Run **idload -n**. This command will print a listing of the mapping rules you set up, without creating translation tables. Figure 9-5 is the output from **idload -n** using the *uid.rules* file shown after Step 1H and a *gid.rules* file with simply **default 60001** in the global block.

```
TYPE  MACHINE         REM_ID     REM_NAME   LOC_ID         LOC_NAME

USR   GLOBAL          DEFAULT    n/a        1000           n/a
USR   GLOBAL          0          n/a        60001          guest_id
USR   peanuts.snoopy  DEFAULT    n/a        transparent    n/a
USR   peanuts.snoopy  0          n/a        60001          guest_id
USR   peanuts.linus   DEFAULT    n/a        60001          n/a
USR   peanuts.linus   0          n/a        100            n/a

GRP   GLOBAL          DEFAULT    n/a        60001          n/a
```

Figure 9-5: Example Output From **idload -n**

Step 4B: Run **idload**. If the output from **idload -n** was acceptable, type the **idload** command with no options to create the translation tables. The **global** rules and **host** rules for any computer that currently has your resources mounted will immediately take effect. Rules for any other computer that you mapped will take effect as soon as that computer mounts one of your resources.

Step 4C: Run **idload -k**. This will print the mapping that is currently in use on your computer. (Remember that rules for any other computer that you mapped will not be in effect until that computer mounts one of your resources.)

ID MAPPING IS NOW COMPLETE!

Once mapping is set up, it can be changed whenever you like. You can edit rules files and run **idload** again at any time. It doesn't matter if resources are mounted or even if RFS is running.

Starting/Stopping RFS

Before a non-primary machine can start Remote File Sharing, RFS must be configured on the machine and the primary must be up and running RFS.

Is RFS Running?

If you are not sure if RFS is running, type **rfadmin -q**. This will tell you simply that RFS is or is not running.

Another way is to check that processes related to RFS are active. To do this, type **ps -e**. These processes should be active:

listen
rfdaemon
nserve
rfudaemon
recovery
server (*optional*)

There may be multiple processes of some of these names running.

Initial RFS Start

The first time you start RFS on a non-primary machine, you should use the following command.

```
rfstart -p "nodename.serve"
rfstart: Please enter machine password:
```

where "nodename.serve" is the address for the primary name server for this domain.

RFS Password

You will be prompted for a password the first time you start RFS. The password must match the password entered when your machine was added to the domain member list in the primary name server (the **rfadmin -a** command). If password verification succeeds, your computer will save this password automatically so you do not have to enter it again.

Likewise, your machine will save the network address of the primary name server. Therefore, the next time you start up Remote File Sharing, you will be able to do it via **init 3**.

RFS Password Mismatches

Any time you start RFS (**rfstart**) and your password does not match the one on the current domain name server, you will receive a warning, but **rfstart** will NOT fail.

Though Remote File Sharing will be active, you may have a problem if the *domain/passwd* file from the primary domain name server is shared with other machines to use for verification. In that case, your remote **mount** requests will fail if the passwords don't match. For this reason, it is recommended that RFS passwords always be kept up-to-date on each computer and the primary name server. If passwords aren't important to you, you can simply enter a carriage return for the passwords on each computer and the primary.

If you do get warnings that your password is out of sync with the current domain name server and you want to fix it, you should handle it differently if the primary is the current domain name server than if the secondary has temporarily taken over.

First, find out which machine is the current name server, and whether it is the primary or the secondary, by doing the following:

```
# rfadmin
the acting name server for domain domain is domain.nodename
# cat /usr/nserve/rfmaster
domain    P   domain.nodename
domain    S   domain.nodename
domain.nodename  A  network_address
domain.nodename  A  network_address
```

Then, depending on which machine is the current name server, do one of the following:

- Secondary is the current name server

  If the primary went down and a secondary took over as domain name server, the secondary may not have a *domain/passwd* file or may have one that is out-of-date. In this case, do not try to correct your password until the primary takes over as domain name server again.

- Primary is the current name server

  Try to correct your password by reentering it with the **rfpasswd** command. If that does not work, follow the sequence shown below, replacing *domain.nodename* with your computer's RFS machine name.

  From the primary name server:

```
# rfadmin -r domain.nodename
# rfadmin -a domain.nodename
Enter password for nodename: type password
```

From your computer:

```
# rfstop
# rm /usr/nserve/loc.passwd
# rfstart
rfstart: Please enter machine password: type password
```

You should then make sure that any computer that verifies your computer's password copies the new *domain/passwd* file from the primary.

Changing RFS Password

If you want to change your RFS password later, you must use the **rfpasswd** command. This will change your RFS password, both on your computer and on the primary domain name server. Processing of the new password follows the same criteria as **passwd**(1) in the *User's/System Administrator's Reference Manual.*

Since changing passwords requires communication with the primary domain name server, Remote File Sharing must be running on both your computer and the primary domain name server. You cannot change your RFS password if the primary is down and a secondary is the current domain name server.

CAUTION When you change your password, computers that are authenticating your computer may not automatically receive the change. If you are unable to mount a resource from a remote machine after you change your password, check that the remote machine has copied the latest version of your domain's passwd file from your primary domain name server.

Automatic RFS Startup (init 3)

There are several steps involved in starting up Remote File Sharing and sharing resources. To simplify this procedure, a new Remote File Sharing run level has been defined: run level 3.

When you enter run level 3 using the **init 3** command, Remote File Sharing is automatically started via **/etc/rc3** from shell scripts in your computer's */etc/rc3.d* directory. These scripts start Remote File Sharing, advertise local resources, and mount remote resources. When you leave run level 3 (using **shutdown** or **init 2**, for example), RFS processes will be stopped.

You can add your own shell scripts to those that start run level 3. You can also tailor the run level 3 shell scripts to suit the way you use RFS.

This section will describe those shell scripts used in run level 3 and suggest how to modify or add to them.

NOTE Before you can enter Remote File Sharing mode, you must have already installed and configured Remote File Sharing. See Procedure 9.1 for information on setting up Remote File Sharing.

Entering Run Level 3

You can go into **init** level 3 in one of three ways:

1. From single-user mode (run level **s**)

   Remote File Sharing mode is also a multiuser mode. Therefore, when you type **init 3** from single-user mode, all multiuser processes (**gettys**,

cron, etc.) wi]l be started, followed by Remote File Sharing mode processes.

2. From multiuser mode (run level 2)

   When **init 3** is run from run level 2, init checks that all multiuser processes are running, then starts the Remote File Sharing mode processes. (**init 3** will not spawn another process for a level 2 script that is already running.)

3. At boot time

   By default, your system will enter run level 2 at boot time. You can change that to have run level 3 start automatically at boot time by changing the value for **initdefault** in the */etc/inittab* file so it reads as follows:

   ```
   is:3:initdefault:
   ```

init 3 Processing

When **init 3** is run, all entries in the *inittab* file that indicate level 3 are started, including */etc/rc3*. */etc/rc3* executes all shell scripts in */etc/rc3.d* that begin with *S*.

RFS places only one file in */etc/rc3.d*: *S21rfs*. This file is linked to the *rfs* file in */etc/init.d*. Also, the *rfs* file is linked to *K50rfs* in */etc/rc2.d* and *K65rfs* in */etc/rc0.d*.

/etc/rc3 executes *S21rfs* with the **start** option upon entering run level 3. *S21rfs* then does the following:

- Validates that the domain name has been defined for your machine.
- Validates that the *rfmaster* file has been created. (This may have been created automatically the first time you ran **rfstart -p** if your machine is not the primary. The latest copy is then sent to your machine from the primary domain name server.)
- Executes the **rfstart** command continuously, with 60-second sleep intervals, until it succeeds or returns a fatal error.
- Executes **/etc/init.d/adv** to advertise all system resources you set up in your */etc/rstab* file. (The */etc/rstab* file contains an entire **adv** command line for each advertised resource.)

- Executes **/etc/rmountall** to mount all remote resources you listed in your */etc/fstab* file. Any remote mount that does not succeed will be tried continuously until it does via **/etc/rmount**(1M). (See "Automatic Remote Mounts" in this chapter for the format of */etc/fstab*.)

When you leave run level 3 via **init 1** or **2**, **/etc/init.d/rfs** is executed with the **stop** option. This will execute **rfstop**.

NOTE

If for some reason RFS fails to terminate the **rfudaemon**, Remote File Sharing may continue to run in the lower run state. You can always bring down Remote File Sharing by running **unadv** and **fumount** for each advertised resource, **umount** for each mounted remote resource, and **rfstop**.

Changing init 3 Processing

Going into **init** state **3** makes some assumptions about how you use your Remote File Sharing system. Here is a description of how to change some of the processing that takes place.

- Retry **rfstart**

  By default, **init 3** will keep trying to start Remote File Sharing (**rfstart**) until it succeeds. If you want it to try a limited number of times, you must edit the */etc/rc3.d/S21rfs* file. Find the line RETRIES=0 and change the number **0** (try forever) to the number of times you want it to retry.

- Retry mounts

  When you enter **init 3**, the system tries separately, every 60 seconds, to mount each resource listed in */etc/fstab* until it succeeds or you leave state 3. To change this behavior you can edit */etc/rmount*. Find the line RETRIES=0 and change **0** (try forever) to the number of times you want to attempt to mount each resource. Find the line TIME=60 and change 60 to the number of seconds you want it to wait between retries.

Adding RFS Mode Scripts

All files in */etc/rc3.d* and other */etc/rc?.***d** directories are shell scripts, so you can read them to see what they do. You can modify the existing files, though it is preferable to add your own since the delivered scripts may change in future releases. To create your own scripts you should follow these rules:

- Place the file in */etc/init.d*.
- Link the file to files in appropriate run level directories using the naming convention described below.
- Have the file accept the **start** and/or **stop** options.

You should name the files using the following conventions:

S*00name*
or
K*00name*

The file names can be split into three parts:

S or **K** The first letter of each file defines whether the process should be started (**S**) or killed (**K**) upon entering the new run level.

00 The next two characters represent a number from 00 to 99. These numbers indicate the order in which the files will be started (S00, S01, S02, etc.) or stopped (K00, K01, K02, etc).

name The rest of the file name is the */etc/init.d* file name to which this file is linked.

For example, the *init.d* file *rfs* is linked to the */etc/rc3.d* file *S21rfs* and *rc2.d* file *K50rfs*. When you enter **init 2**, this file is executed with the **start** option: **sh S68starlan start**. When you enter **init 0**, this file is executed with the **stop** option: **sh K67starlan stop**.

Stopping RFS

If you started RFS using **init 3**, you can stop it by going to a lower run state (**init 2**, **init S**, or **shutdown**). If you started RFS using **rfstart**, you can stop it by typing **rfstop**.

Before you can use **rfstop**, you must:

- Unadvertise all your resources (**unadv**).
- Unmount everything you have mounted from remote machines (**umount**).
- Make sure all your advertised resources are unmounted from remote machines (can be forced by using **fumount**).

These steps will happen automatically when you leave **init** state 3.

If you are the primary name server, you should not stop RFS unless a secondary is up and ready to take over. If the primary goes down and no secondary is available to take over, computers in the domain that are not the primary or a secondary will be able to start RFS. Computers that are already running RFS will continue to run RFS; however, they will not be able to mount or advertise new resources.

Sharing Resources

This section describes how to share your local resources with other computers (advertising) on a Remote File Sharing system and how to use the resources other machines have made available (mounting).

Local Resource Advertising

The **adv** command is used to advertise a local directory (one that physically resides on your machine) so it is accessible to other machines. When you advertise a directory, you must assign it a resource name. This resource must have a unique name within your domain.

When **adv** is performed with the options listed below, the resource is registered with your domain name server. Any computer that has access to your domain can find a listing of your resource from your domain's advertise table (**nsquery** command). A remote computer will not know the exact location of the resource on your machine. All the remote computer will know is its resource name, the short description you assign, that it resides on your computer, and the read/write permissions.

You can set up your system so resources are advertised automatically when you enter **init 3**. To do this, place the entire command line for each advertised resource in the */etc/rstab* file.

The syntax of the **adv** command to advertise a resource is:

adv [**-r**] [**-d** "*description*"] *resource pathname* [*clients...*]

The syntax of **adv** to modify an advertised resource entry is either:

adv -m *resource* **-d** "*description*" [*clients* ...]
or
adv -m *resource* [**-d** "*description*"] *clients* ...

The options are as follows:

-r The **-r** option, for read-only, is used to advertise the resource with read-only access. If it is not used, read/write access is assumed.

-d The **-d** option indicates that the next argument (*description*) is a description of the resource. The description can be from 0 to 32 characters and should be in quotes.

resource — This is the resource name you assign. The name is limited to 14 printable ASCII characters; slash (/), period (.), and spaces and tabs may not be used. (If you enter more than 14 characters, the name will be accepted and truncated.)

pathname — This is the full pathname to the directory you want to share. The directory must be on your local system, and it cannot be already advertised.

clients... — This is an optional list of one or more remote machines or domain names to which you want to restrict this resource. [A domain name must have a period (.) appended to it.] If clients are not included, the resource will be accessible to any Remote File Sharing computer that can connect with your computer. You can also define aliases so a single name can represent a group of computers and/or domains. (See "Aliases" on the following page.)

-m *resource* — This option is used to modify the *description* or *client* fields for an advertised resource listing. It cannot be used to change the read/write permissions of a resource.

Below are two examples of **adv** command lines:

```
adv -r -d "Department news" DNEWS /usr/news peanuts.
adv -d "My devices" MDEV /dev lucy linus doc.compl
```

The first example advertises your */usr/news* directory with read-only permissions under the resource name **DNEWS** to all computers in the **peanuts** domain. The second advertises your */dev* directory as **MDEV** to computers **lucy** and **linus** in your domain and **compl** in domain **doc**.

Automatic Advertising

You can set up all your **adv** commands to start automatically when the system enters the Remote File Sharing state (**init 3**). Do this by placing each full **adv** command line in the */etc/rstab* file. As soon as **init 3** successfully starts Remote File Sharing, all **adv** commands in */etc/rstab* will be run.

The following is an example of an */etc/rstab* file to automatically advertise the two resources shown previously:

```
# cat /etc/rstab
adv -r -d "Department news" DNEWS /usr/news peanuts.
adv -d "My devices" MDEV /dev lucy linus comp1.doc
#
```

Aliases

The **adv** command reads the */etc/host.alias* file to find the definitions of any aliases in the *clients* field. The format of the file is:

alias name client1 client2 client3 ...

where *name* is replaced by the character string you want to represent the list of clients. Each client can be a machine name, domain name, or an alias name previously defined in the file. The three fields must be separated by blanks or tabs. If you have too many *clients* to fit on one line, you can extend an entry beyond one line by entering a back slash and then a carriage return.

Resource Security

These are the levels of security that protect your resource once you have advertised it:

Verify computers Only those remote computers that pass your security checks can even connect to your machine. You may have indicated that only those machines you have a record of can connect (see **rfstart -v**).

Restrict Resource You may have advertised your resource so only selected remote computers can mount it (see the *client* option of the **adv** command).

Map IDs The permissions remote users will have to your resources are set on a computer-by-computer basis. In other words, the user and group mappings you set up for a remote computer will apply to any of your resources that computer mounts.

UNIX system security Normal UNIX system access security, governing read, write, and execute permissions, will apply to any advertised resource.

Local Advertise Table

All advertised resources for your computer are contained in your computer's local advertise table. Any user can use the **adv** command with no options to display the local advertise table. The output will be a listing like the one that follows:

```
# adv
CUSTOMER /usr/bin/cust read-only  "Atlanta customers"    lucy linus doc.tick
SCCS     /sccs         read/write "Project Y source"     unrestricted
CALENDAR /usr/bin/cal  read-only  "UNIX System calendar" peanuts. compgrp
```

The information will match what you entered using the **adv** command, with appropriate options, for each resource. Some of the information shown was implied when the resource was advertised. For example, access is read/write if **-r** is not specified and clients are not limited to certain machines (**unrestricted**) when no clients are identified. Clients listed in the last field can be:

- Computer names in your domain (**lucy**)
- Computer names in other domains (**doc.compl**)
- Domain names (**peanuts.**)
- Aliases listed in */etc/host.alias* (**compgrp**)
- **unrestricted** if the resource is not restricted to certain machines.

Domain Advertise Table

All advertised resources for your domain are in the domain advertise table on your domain name server. The **nsquery** command is available to all users to list any or all of the advertised resources in a domain. The syntax of the command is:

nsquery [**-h**] [*name*]

where the **-h** option can be used to suppress printing of the heading line and the *name* option can be replaced by one of the following:

nodename — To list the resources advertised by a particular computer in your domain.

domain. — To list all resources advertised by all machines in a domain. (A period at the end of a name causes it to be interpreted as a domain name.)

domain.nodename — To list all resources advertised by a particular computer in a domain outside your own domain.

If the *name* option is not used, **nsquery** will print a list of all advertised resources in your domain. Here is an example of output from an **nsquery** command:

```
# nsquery peanuts.lucy
RESOURCE      ACCESS        SERVER          DESCRIPTION

GRAPHICS      read/write    peanuts.lucy    Domain files
CALENDAR      read/write    peanuts.lucy    Monthly meetings
USERHELP      read-only     peanuts.lucy    System help information
```

For each available resource, **nsquery** will list the resource name (RESOURCE), the permissions (ACCESS), the computer that owns the resource (SERVER), and the description of the resource.

NOTE

The output from **nsquery** does not indicate whether you have permission to mount the resource.

Advertised Resources in Use

You can use the **rmntstat** command to find out what remote computers have mounted your advertised resources. This command can print output for all your resources or the one you choose. The syntax is as follows:

rmntstat [**-h**] [*resource*]

where **-h** will print the output without the heading and *resource* can be used to restrict output to information for a particular resource. Here is an example of the output from **rmntstat**:

```
# rmntstat
RESOURCE          PATH              HOSTNAMES
DNEWS             /usr/news         peanuts.linus peanuts.lucy
MDEV              /dev              peanuts.linus
SPECIAL           unknown           peanuts.charlie
```

The output shows the resources your machine has advertised, where those resources are located on the local machine, and the computers that have mounted the resource.

NOTE

If **unknown** appears in the pathname field, it means you have unadvertised the resource, but it is still mounted on the listed remote machines.

Unadvertise

You can unadvertise any of your computer's resources using the **unadv** command. It will remove the resource from the advertise tables on your computer and the domain name server.

The domain administrator can use **unadv** to unadvertise any resource within the domain. (You should only use **unadv** when the machine has gone down, otherwise, the domain and your computer's advertise tables will not match.)

Unadvertising does not remove a currently-mounted resource from a remote computer [see **umount**(1M)]. It does, however, prevent additional machines from mounting the resource. There are two reasons you may want to use this command.

1. Before you can unmount (**umount** or **fumount**) one of your file systems containing an advertised directory, it must be unadvertised.

2. If you want to restrict a previously-shared directory to only local access, you will want to unadvertise it.

Because advertise commands can be set up to run automatically in **init 3**, you may have to remove them if you want them permanently unadvertised. (See the "Advertise Resources" section of this chapter for information on how to modify the */etc/rstab* file.)

The syntax of the **unadv** command is:

unadv *resource*
or
unadv *domain.resource*

where *resource* is used to unadvertise one of your machine's resources, and *domain.resource* is used by a domain name server to unadvertise any resource in its domain. In the second case, the resource name is prefixed by the domain name in which it resides and a period (.).

Forced Unmount

You cannot unmount a local file system using the **umount** command if any part of that file system is mounted remotely. Normally, you should tell each administrator whose machine has mounted such a resource to unmount it. In this way, a resource can be removed in an orderly fashion.

When you have to unmount a local file system immediately, however, you can use the **fumount** command. **fumount** will remove a remotely-mounted resource from all machines that have mounted it. You should only do this in cases where it is urgent that the resource be removed, because you may be cutting off remote processes that are accessing the resource.

The syntax of the **fumount** command is:

fumount [**-w** *sec*] *resource*

where the **-w** option says to wait *sec* seconds before remotely unmounting the resource and *resource* is replaced by the resource name.

When you execute **fumount**, this is what happens:

1. The resource is unadvertised.
2. If the **fumount** command is executed with a grace period of several seconds, the following shell script is run on all client machines currently using the resource.

/usr/nserve/rfuadmin fuwarn *resource sec*

By default, this shell script will write to all terminals on all client machines:

```
resource will be disconnected from the system in
sec seconds.
```

(You can edit **rfuadmin** to tailor the action taken in response to **fumount**.)

3. After the grace period of *sec* seconds, the resource is removed from all remote machines it is mounted on. The following message is then sent to all terminals:

```
resource has been disconnected from the system.
```

4. On each client machine, **rfuadmin** then executes **rmount**. **rmount** will try to remount the resource every 60 seconds until it succeeds.

See **rfuadmin**(1M) and **rmount**(1M) for further information on processing these commands.

Remote Resource Mounting

You can attach another computer's advertised resource to your system using the standard **mount** command. You simply choose an existing directory, preferably empty, or create a directory to use as a mount point and **mount** the resource, using the **-d** option.

When you try to **mount** a remote resource, a request is sent to the computer that advertised the resource. If you have permission to mount the resource, the resource will be added to your mount table and connected to the mount point you specified. You can list the remote resources, as well as local file systems, mounted on your computer using the **mount** command without options.

The form of the **mount** command to mount a remote resource is as follows:

mount [-r] [-c] **-d** *resource directory*

The options are as follows:

-r The **-r** option indicates that the resource should be mounted read-only. If it is not used, the resource is mounted with read/write permissions. (A remote resource can only be mounted read/write if it was advertised that way.)

-d This option is used to indicate that you are mounting a remote resource.

resource This must be replaced by the resource identifier assigned by the computer that advertised the resource.

directory This must be replaced by the full path to the local directory on which you want to mount the resource.

-c This option indicates that remote reads and writes for the remote resource you mount should NOT be cached in the local buffer pool. You will generally want the default (buffer caching on), since caching will cut down on network access and improve RFS performance. (See the "Monitoring" and "Parameter Tuning" sections of this chapter for more information on monitoring client caching activities.)

Automatic Remote Mounts

You can set up your **mount** commands to run automatically when your computer enters the **init 3** state. You do this by adding mount information to the */etc/fstab* file. The format of */etc/fstab* for remote mounts is as follows:

resource directory **-d**[**r**]

resource is replaced by the resource name, *directory* is replaced by the directory where the resource will be mounted, and **-d** says this is a remote mount. You can use **-dr** instead of **-d** if you want to mount the remote resource read-only.

Mounting Guidelines

Below are some guidelines that apply to resources.

- Once you have advertised a resource from your computer, you can:
  - Mount a local file system on a subdirectory of the advertised resource. The new file system will become part of the advertised resource. (You cannot mount directly on the advertised mount point, however.)
  - Mount a remote resource on subdirectories of your advertised resource. The remote resource you mount will not become part of the resource, however. Only your local users will be able to access it. (Remote users will be able to see this mount point directory, but will get a "multihop" error message if they try to access the directory in any way.)

  NOTE You cannot mount a remote resource directly on an advertised directory.

- If a resource was advertised with read-only permissions, you must mount it read-only. If it was advertised read/write, you have a choice of mounting it read-only or read/write.

Mounting Rules

There are some rules you must follow to avoid unexpected results when mounting remote resources.

Rule #1 Mounting over basic directories

A directory containing files that define your local machine should not be used as a mount point for a remote resource. This will result in essential local files being inaccessible to your system.

For example, you shouldn't mount a remote */dev* on your machine's */dev* directory or you will make your machine's console inaccessible (*/dev/console*). As another example, if you mounted an */etc* directory on your *etc* directory, you would cover your local *inittab, passwd,* and *mnttab* files, to name a few. Some other directories that fall into this category are: */, /usr, /usr/bin, /usr/nserve, /usr/net,* and */shlib.*

Rule #2 Mounting spool and work directories

Like Rule #1, Rule #2 has to do with mounting a directory from one computer on the same directory on another computer. In this case the problem is spool files and workspace directories. Applications such as **uucp**(1) and **lp**(1) can run into problems when multiple machines are trying to create spool files or lock files in the same directory. For example, if you share the */usr/spool/locks* directory, by using a tty device for **uucp** on one machine, you would prevent use of a device of the same name on another machine. Also, mounting */tmp* can cause collisions among temporary files.

Rule #3 File systems on remote devices

When a remote machine advertises a directory containing a device and that device contains a file system, you would not be able to mount the file system by simply mounting the resource containing the device. To access the file system on the remote device, the remote machine would have to mount the device locally, then advertise that mount point. (You can access the remote as a raw device, however, by simply mounting the resource containing the device.)

Rule #4 Using remote sticky bit programs

Mounting remote resources that contain executable files with the sticky bit on can improve performance of those files. When executed on your machine, the text portion of the sticky bit program will remain in main memory on your machine, thereby reducing the network overhead on future executions. From your perspective as a client, you should be careful not to mount too many sticky bit programs or you could unknowingly gobble up a lot of memory.

If your machine is a server sharing sticky bit files, you should be aware that they are treated differently from strictly local sticky bit files. Before removing sticky bit programs from an advertised resource, you must unmount the resource from all client machines [**fumount**(1M)], remove the program, then readvertise the resource. You should do this to prevent out-of-date text for recompiled or deleted files from remaining in memory on client machines.

Local Mount Table

You can list the remote resources that are mounted on your computer as you would list local file systems: the **mount** command with no options. Remote resource output from this command will appear in the form:

directory **on** *resource permission* **on** *date*

where *directory* is the name of the directory where the remote *resource* is mounted, the *permission* is **read only/remote** or **read/write/remote**, and *date* is the time and date the resource was mounted.

The following is an example of output from the mount command, with no options. The last two entries in this example are remote resource mounts.

```
$ mount
/ on /dev/dsk/0s1 read/write on Tue Sept 1 09:07:19 1987
/usr2 on /dev/dsk/0s4 read/write on Tue Sept 1 09:07:32 1987
/usr on /dev/dsk/0s3 read/write on Tue Sept 1 09:07:33 1987
/s/codes on LCODE  read/write/remote on Tue Sept 1 09:10:13 1987
/s/timing on TEMPO read only/remote on Tue Sept 1 09:10:27 1987
$
```

Remote Resource Disconnected

When a machine that shares its resources with you goes down or the network connection is broken, resources that you have mounted from the server will be disconnected.

A Remote File Sharing daemon process (**/usr/nserve/rfudaemon**) runs an administrative shell script (**rfuadmin**) to try to clean up when a resource has been disconnected. It then tries to remount the remote resource as soon as it becomes available again.

rfudaemon

The **rfudaemon** process is run automatically when Remote File Sharing is started (**rfstart**) and continues to run until it is stopped (**rfstop**). The **rfudaemon** process waits for one of the following events to occur, then passes that information to the **rfuadmin** administrative shell script.

disconnect When a link is cut to a remote resource, the **rfudaemon** process sends a **disconnect** message and the resource name to the **rfuadmin** shell script.

fumount When a resource is unmounted (**fumount**) by the server, the **rfudaemon** process sends a 3fumount message and the resource name to the **rfuadmin** shell script.

fuwarn When a server sends a message that a resource is about to be unmounted (**fumount**), the **rfudaemon** process sends a **fuwarn** message, the resource name, and the number of seconds before the resource will be unmounted to the **rfuadmin** shell script.

rfuadmin

When links to resources are disconnected, the response to the disconnect is handled at the user level by the **/usr/nserve/rfuadmin** shell script. By editing this shell script, you can tailor the response your system makes when the connection to a remote resource is lost. The **rfudaemon** process starts **rfuadmin** with one of the following arguments.

disconnect *resource*

When **rfuadmin** is started by **rfudaemon** with these arguments, **rfuadmin** sends this message to all terminals using the **wall**(1) command:

```
resource has been disconnected from the system.
```

Then it executes **fuser**(1M) to kill all processes using the resource, unmounts the resource [**umount**(1M)] to notify the kernel, and starts **rmount** to try to remount the resource. The assumption is that the link was either broken by mistake or that as soon as the server makes the resource available again, the client will want to mount it.

fumount *resource*

When **rfuadmin** is started by **rfudaemon** with these arguments, the processing is similar to a disconnect.

fuwarn *resource seconds*

When **rfuadmin** is started by **rfudaemon** with these arguments, **rfuadmin** sends this message to all terminals:

```
resource is being removed from the system in sec seconds.
```

There are many reasons you may want to change the **rfuadmin** shell script. If access to a resource is lost, you may want to respond by trying to mount another resource. You may want to send different messages when a resource is lost.

NOTE

When a resource is disconnected, **rfuadmin** tries to remount the resource using **/usr/bin/rmount**. This command retries the remount every 60 seconds until it succeeds. To change this behavior, you must either edit **/etc/rfuadmin** so it no longer does an **rmount**, or edit **rmount** so it retries a limited number of times [see **rmount**(1M)].

Unmounting

You can unmount any remote resource you have mounted with the **umount** command. The syntax for using **umount** to unmount a remote resource from your computer is as follows:

umount -d *resource*

where *resource* is replaced by the name of the resource you are unmounting.

Before you run **umount**, you should make sure none of your users are using the resource with the **fuser** command. When the **fuser** command is run, it lists the processes on your computer that are accessing a mounted remote resource. It can then be used to kill all processes relating to a resource.

The form of **fuser** for reporting on remote resources mounted on your machine is:

fuser [**-ku**] *resource ...*

where **-u** will list user names in the report of processes that have files open in any directory or subdirectory relating to the resource, and **-k** will kill all processes that have files open in any directory or subdirectory relating to the resource.

Mapping Remote Users

NOTE The ''Complex User ID/Group ID Mapping'' procedure in this chapter is designed to act as a tutorial for setting up ID mapping. This section provides further reference information to support that section.

Your computer has a set of users, defined in the */etc/passwd* file. These users can also be members of groups that are defined in the */etc/group* file. The user and group ID assignments are used by the system to evaluate requests by the user for access to local files, directories, and devices.

When you share your directories with other computers using Remote File Sharing, you have the ability to define the permissions each remote user will have to your resources. You do this by mapping remote users and groups into the permissions of existing users and groups on your computer. You also have the option of mapping remote users and groups into a special ''guest ID'' that doesn't map into permissions of any existing users and groups on your system.

If you do not want to map remote users, you do not have to. The default treats all remote users as the special guest ID, which has the ID number of MAXUID plus one. MAXUID is the maximum ID number defined for the system, so MAXUID+1 is always guaranteed not to overlap with any current or future users (by default, MAXUID+1 is 60001).

When a remote user checks the ownership of one of your resources, the user might see another special ID: MAXUID+2 (or 60002). No files will ever be owned by 60002 on the system where a file resides. MAXUID+2 is simply a way of telling remote users that a file or directory is not owned by them or any other users from their system. For example, if all users on a remote system were mapped to 60001, any files created by one of your local users would appear to be owned by user ID 60002.

How Mapping Works

When you set up your remote user and group mapping for a remote computer, you define how requests from users and groups will be handled. This mapping has an impact on the remote users' access to files and directories on your resources, as well as each remote user's view of ownership.

For example, say you map user ID **101** from machine **abc** into user ID **115** on your machine. When **101** from **abc** tries to create a file in a directory of one of your advertised resources, your machine will translate the request from **abc**'s **101** into a request from **115**. If local ID **115** has permissions to create a file in that directory, then the file will be created.

If you tried to **stat** the file on your machine (**ls -l**, for example) you would see that user ID **115** was the owner. However, if a **stat** comes from machine **abc**, your machine would do inverse mapping. Therefore, the user from **abc** would see the file as being owned by user ID **101**.

Inverse mapping from the machine that owns the resource (the server) provides the most consistent file system view to a remote user. It could potentially cause confusion, though. Continuing with the example, say that instead of just mapping **101** into **115**, you also mapped **102** from **abc** into **115** on your machine. A file created by **102** would correctly create the file as owned by **115** on your machine. However, when a user from **abc** **stat**s the file, it would always show ownership by the smaller numeric value: **101** user ID.

NOTE

This same result will occur if you gave several local user names the same numeric user ID.

If users are confused when files they create do not seem to belong to them, the situation described above could be the reason. This does not cause any problems with each user's ability to access the resource. However, it could break some programs that are dependent on local IDs. The most consistent way to map, however, is one-to-one remote to local IDs.

Mapping Components

You must use the **idload**(1M) command to do the user and group mappings. This command reads the user and group mapping rules you create, reads your computer's */etc/passwd* and */etc/group* files, if needed, and maps the remote users into your users' permissions. If you are using remote user and group names to map into your computer, you must have access to user and group lists from the remote computers, so **idload** can read the files and translate those names into the appropriate numeric ID numbers.

Rules Files

The rules files you create will tell **idload** how to map remote users. Both files are in */usr/nserve/auth.info* under the names, *uid.rules* for user rules and *gid.rules* for group rules.

Figure 9-6 shows how the user rules file can be structured. The format of the group rules files is exactly the same. All lines in each file are optional.

```
global
default local_id | transparent
exclude [remote_id-remote_id ...] | [remote_id]
map [remote_id:local ...]

host domain.nodename ...
default local | transparent
exclude [remote_id-remote_id ...] | [remote_id ...] | [remote_name ...]
map [remote:local ...] | remote | all
```

Figure 9-6: Format of *uid.rules* and *gid.rules* files

The following notation is used in the previous figure:

local_name = a local user name
local_id = a local user ID number
remote_id = a remote user ID number
remote_name = a remote user name
local = a local name or ID number
remote = a remote name or ID number

A rules file is divided into blocks of information. Each block is either a **global** or **host** block. There is only one **global** block per file, but there can be one **host** block for each computer mapped.

global This line starts the block of global information. Each line of definitions after **global** and before the first **host** line will be applied to all computers that are not explicitly defined in **host** blocks. You can use **default, exclude**, and **map** inside **global** blocks.

You cannot map or exclude names in **global** blocks. You must use ID numbers.

host *domain.nodename ...*

This line starts a block of information for a particular computer. Each line of definitions following this line and before the next **host** line will be applied to the *domain.nodename* specified. You can use **default, exclude**, and **map** inside **host** blocks.

If you want to map more than one computer from a single set of *passwd* and *group* files, you can put several computer names on one line. In this case, **idload** will read the *passwd* and *group* files for the first computer referenced (if you map by name) and use the information in those files for all computers that are referenced.

A computer can only be mapped once in each rules file.

Each of the following lines of information can appear in either a **host** block or a **global** block. A name or an ID should only be mapped once in each block. If one is mapped more than once, the first reference is in effect and the others will produce warning messages from **idload**.

1. **default** *local* | **transparent**

   One **default** line can be put in each block to indicate how to handle remote users and groups that are not explicitly mapped or excluded.

   transparent means use the same numeric ID on your machine that the user had on the remote machine for undefined users. So if a request comes from remote uid **101**, that request will have the permissions of local uid **101**.

   local is replaced by a local user name or ID number. By default, all remote users will be mapped into the permissions of the local user indicated by name or ID. If a default line does not appear in a block, MAXUID+1 permission will be assigned.

2. **exclude** [*remote_id-remote_id*] | [*remote_id*] | [*remote_name*] ...

   Optional **exclude** lines can go into a block to exclude certain users from the default mapping. Zero or more ranges of ID numbers (*remote_id-remote_id*), single *remote_names*, or single *remote_id* numbers can be excluded. (*remote_name* is not available in the global block.)

   A user who is excluded will still have access to your resources but will only have permissions of the MAXUID+1 user. All **exclude** lines must go before any **map** lines in a block.

3. **map** [*remote:local*] | *remote* | **all**

   You can use **map** lines in each block to assign local permissions to particular remote users. There are several ways to use the **map** command. You can set any remote user's permissions to any local user's permissions by either local user *id#* or *name;* separate the two with a colon (:). By entering a single *remote_id* or *remote_name,* the remote user who matches will have the permissions of the local user of the same ID or name. For example:

   ```
   map mcn
   ```

   would give the remote **mcn** the same permissions of the local user **mcn**.

   The literal entry **all** maps all users by user name into the permissions of users with the same name on your computer.

Multiple **map** lines are valid. You cannot map by remote name in **global** blocks.

NOTE

map all and mapping by name are not allowed in a global block. **map all** will usually produce warning messages, since multiple administrative logins will have uid 0, and **idload** will try to map each one to 0. There is no harm in this.

idload Command

Once the rules files are created, use the **idload** command to read your rules files and create mapping translation tables. When you run **idload**, the rules in **global** blocks and any **host** blocks that have resources currently mounted immediately take effect. All other **host** block rules will take effect when the remote machine mounts one of your resources.

The syntax of **idload** is:

idload [**-n**] [**-k**] [**-g** *g_rules*] [**-u** *u_rules*] [*directory*]

The options are as follows:

-n This is the "no update mode" option. When it is used, **idload -n** will print the mapping that would result from the rules files without putting them into effect.

-k This option shows the mapping that is currently active on your machine. (Note that there will be mapping ready to take effect that is not shown as active when you do not currently have a connection to a remote machine.)

-g *g_rules* This option lets you use a group rules file other than */usr/nserve/auth.info/gid.rules* as input for group mapping rules.

-u *u_rules* This option lets you use a user rules file other than */usr/nserve/auth.info/uid.rules* as input for user mapping rules.

directory This option indicates that some directory other than */usr/nserve/auth.info* contains the *domain/nodename* directories where the *passwd* and *group* files for each remote computer reside. If it is not used, */usr/nserve/auth.info* will be assumed.

Each time you set up or change your rules files, first run **idload** with the **-n** option. The results will show you the mapping that will occur when the command is run to actually load the IDs. You must then run **idload** for the rules to go into effect.

Remote Computer passwd and group Files

If you are mapping remote users by name, you will need lists of these users from each remote computer. These lists should be copies of the */etc/passwd* and */etc/group* files from each computer.

If **idload** finds a request for a remote user name in a **host** information block, it will check the directory for that computer for *passwd* and *group* files. The pathname to the remote computer's directory will be

/usr/nserve/auth.info/domain/nodename

on your system, where *domain* and *nodename* are replaced by the remote computer's domain and the remote computer's nodename, respectively (unless you overrule this using the **-g** and **-u** options).

NOTE

Mapping by name can be a very useful feature. However, if you map only by ID number or local name, and avoid mapping by remote names, you will avoid the need to coordinate distributing and updating remote *passwd* and *group* files and rerunning **idload**.

Example Rules Files

This section describes some strategies you can use to map users. It describes the easiest way to deal with remote user permissions and progresses to the most complicated ways. Read through each example to decide what strategy is best for your computer.

No Mapping

If you do not run **idload** to map users, all remote users will have the permissions of the user ID number MAXUID +1, which is the maximum ID number defined on your system plus one. Because there are no users on your system with that user ID number, remote users will only have access to files created by your users that are open to all users.

Mapping Remote IDs

If you map remote users using remote ID numbers and local ID numbers and names, you do not need to get any *passwd* and *group* files from remote computers. The following displays contain some simple examples of mapping that only involve remote ID numbers.

In Figure 9-7, all remote user IDs will be mapped into the same user ID permissions on your computer, except for **root** (ID number **0**), which would only have special guest permissions. This would apply to all remote computers.

CAUTION

The `exclude 0` line is strongly recommended to prevent possible security breaches from **root** users on other systems.

```
global
default transparent
exclude 0
```

Figure 9-7: **uid.rules** File: Setting Global Defaults

In Figure 9-8, users have the same permissions as in the previous example, except remote user IDs **0** through **100** will have MAXUID +1 permissions, and any user ID **732** would have the same permission as local user ID **106.**

```
global
default transparent
exclude 0-100
map 732:106
```

Figure 9-8: **uid.rules** File: Global Mapping by Remote ID

In Figure 9-9, the users from computer **lucy** in domain **peanuts** will not be mapped by the global rules. Instead, all users will have the permissions of local user **mpg** except that user IDs 0 through 50 will have MAXUID +1 permissions.

```
global
default transparent
exclude 0-100
map 732:106

host peanuts.lucy
default mpg
exclude 0-50
```

Figure 9-9: **uid.rules** File: Host Mapping by Remote ID

Mapping Remote Names

If you want to use specific remote user names to map into your local users' permissions, you will need to have access to *passwd* and *group* files from those computers on your system. Below are some examples of ways you can map remote user names.

map all

If you have the same set of user names on different machines, but the user IDs differ, you may want to use **map all** as shown in Figure 9-10.

In Figure 9-10, each user name from computer **lucy** in domain **peanuts** will have the same permissions as the same user name on your computer. The only exceptions will be users **mary**, **root**, and **uucp**, who will have MAX-UID +1 permissions.

```
global
default transparent
exclude 0

host peanuts.lucy
exclude mary 0 uucp
map all
```

Figure 9-10: **uid.rules** File: Mapping by Name With **map all**

map name:name

You can also map particular remote user names into local user names or user IDs on your computer. Figure 9-11 is an example.

```
global
default transparent
exclude 0

host peanuts.lucy
default transparent
exclude 0
map mcn:jcb ral gwn:103
```

Figure 9-11: **uid.rules** File: Mapping Specific Users by Name

Here all users from the computers will be mapped into their same user ID with the following exceptions. Remote user **mcn** will have the permission of local user **jcb**, remote user **ral** will have permissions of local user **ral**, and remote user **gwn** will have permissions of local user ID **103.**

List Current Mapping

There are two ways to list the mapping you have set up: **idload -n** and **idload -k.** The **-n** option inspects the rules files and prints a listing of what would be in effect were you to load them. The **-k** option prints the mapping that is currently in effect in the kernel.

Figure 9-12 shows the result of **idload -n** used for the example shown previously. (The **gid.rules** file simply has the global block set at **default transparent.**) The **-n** option says to print the mapping that is set up in the rules file. You should do this before you run **idload** without options so you can see the mapping that will take effect.

```
# idload -n

TYPE  MACHINE        REM_ID     REM_NAME   LOC_ID         LOC_NAME

USR   GLOBAL         DEFAULT    n/a        transparent    n/a
USR   GLOBAL         0          n/a        60001          guest_id
USR   peanuts.lucy   DEFAULT    n/a        transparent    n/a
USR   peanuts.lucy   0          n/a        60001          guest_id
USR   peanuts.lucy   100        mcn        105            jcb
USR   peanuts.lucy   102        gwn        103            n/a
USR   peanuts.lucy   191        ral        101            ral

GRP   GLOBAL         DEFAULT    n/a        transparent    n/a
```

Figure 9-12: Output From **idload -n**

If you were to then run **idload**, the mapping shown above would take effect. If you were then to run **idload -k**, and the machine called **peanuts.lucy** did not have a resource mounted, you would see that the output in Figure 9-13 was active.

```
# idload -k

TYPE  MACHINE   REM_ID     REM_NAME   LOC_ID         LOC_NAME

USR   GLOBAL    DEFAULT    n/a        transparent    n/a
USR   GLOBAL    0          n/a        60001          guest_id

GRP   GLOBAL    DEFAULT    n/a        transparent    n/a
```

Figure 9-13: Output From **idload -k**

All mapping to **peanuts.lucy** will become active as soon as you are connected to it.

NOTE

The output from **idload** with the **n** and **k** options could be different if you have changed the rules files, but not yet run **idload** without options. Also, the **k** option will not show mapping for computers that are not currently mounting a resource from your machine, even though the mapping would be in effect as soon as the remote machine mounted one of your resources.

Domain Name Servers

One machine in each RFS domain must be chosen to be the primary name server and zero or more can be secondary name servers. The duties of these machines are described briefly under the "Name Service" heading in the "Overview" section of this chapter. This section describes the "how-to" of being a name server.

Before you run any of these tasks, you will want to know which machines are assigned as name servers and which machine is the current name server. To find the current name server, type:

```
rfadmin
```

To find the name server assignments, type:

```
cat /usr/nserve/rfmaster
```

The line in the *rfmaster* file that has a **P** in the second field designates the primary name server. If an **S** is in the second field, the entry designates a secondary name server. (Lines with **A** in the second field designate the network address of a primary or secondary.)

Primary Name Server

If your machine is the primary domain name server, you are responsible for maintaining domain information. The "Setting Up RFS" section of this chapter describes primary name server responsibilities as you set up your machine and domain. You may refer to the following paragraphs in that section to change your RFS configuration after initial configuration.

- Create *rfmaster* File
- Add/Delete Domain Members
- Resource Sharing with Other Domains
- Multiple Domain Name Service.

NOTE

If you want to change the primary and secondary designations in the *rfmaster* file for a domain that is currently running, you must follow this procedure to make sure those changes are properly put in place.

1) Stop Remote File Sharing on all primary and secondary domain name servers for the domain (**rfstop** or **init 2**).

2) Change the *rfmaster* file on the old primary and the new primary.

3) Start the primary designated in the new *rfmaster* file (**rfstart** or **init** 3).

4) Start the secondaries designated in the new *rfmaster* file (**rfstart** or **init** 3).

Once changed on the name servers, each individual computer will pick up the change the next time it starts Remote File Sharing.

Secondary Name Server

Because a secondary is only intended to take over domain name service temporarily, its main responsibility is to pass name server responsibility back to the primary as soon as possible. It does not happen automatically! Most domain maintenance (adding new computers or changing the *rfmaster* file) cannot be done while the secondary is acting domain name server. The secondary simply maintains information that machines need to mount and advertise resources.

To pass name server responsibility back to the primary once it is again running RFS, type the following from the secondary:

```
rfadmin -p
```

The **rfadmin -p** command will pass the domain name server information to the primary or to one of the other computers listed in the domain's *rfmaster* file if it cannot contact the primary. (Note that name service will automatically be passed off when the current name server goes down.)

Recovery

As a domain name server, computers in your domain rely on your machine for information on domain resources and domain member machines. Remote File Sharing is designed to recover quickly when communication is cut between machines and the name server. The following sections describe Remote File Sharing events that can occur and the recovery mechanisms designed to handle them.

Primary Goes Down

All essential domain records are maintained on the primary domain name server. The primary regularly distributes the most critical of these records to secondary domain name servers. (These records do not include files and directories under */usr/nserve/auth.info*.)

If the primary goes down, domain name server responsibilities are passed to the first secondary name server listed in the *rfmaster* file. The secondary is only intended to take over temporarily. The reason is that a secondary has limited name service capabilities. This is done to maintain the definitive domain records on the primary. Changing the name server does not affect any currently mounted resources.

While a secondary is acting domain name server, these functions cannot be done:

- Maintaining domain member lists

  Computers cannot be added or deleted from domain member lists while a secondary is acting domain name server.

- Changing RFS passwords

  Neither the secondary nor another computer can change RFS authentication passwords while a secondary is acting domain name server.

The secondary will maintain lists of advertised resources for the domain and continue basic name server functions so Remote File Sharing activities can continue. In most cases, the computers in the domain shouldn't be aware the primary is down. When the primary comes back up, the secondary should pass name server responsibilities back to the primary using the **rfadmin -p** command.

NOTE

When a primary crashes without properly shutting down Remote File Sharing and passing name server responsibilities to a secondary in an orderly fashion, the advertise table on the secondary may contain some errors. Resources from the primary may still be listed as available and recently advertised resources from other computers may not appear on the list. You can fix the domain advertise table using **unadv** and **adv -m** commands from the domain name server.

Primary and Secondaries Go Down

If all primary and secondary name servers go down at once, all information on advertised resources will be lost. Active mounts and links, however, are not disturbed. The problem is that when the primary comes back up, each computer will still think its resources are advertised but the primary will have no record of these advertised resources.

As soon as the primary is running, each computer can make sure its advertised resources are in sync with those listed on the primary in one of two ways:

- Readvertise with **-m**

  This is a less drastic way to update the advertise tables on the primary. Readvertise each resource using the **adv -m** command from the computer where the resource resides. This command will get the primary and remote computer's advertise tables back in sync.

- Restart Remote File Sharing

  You can bring down Remote File Sharing, bring it back up, and then readvertise your resources. You can do this automatically by going from **init 3** to **init 2** to **init 3**.

Monitoring

This section describes the commands used to monitor Remote File Sharing activity, the reports they produce, and possible action you can take to make sure that your system is operating at peak efficiency. In general, these reports can help you decide if you want to:

- Change parameter settings to better match the way your system is used
- Move resources from machines with heavy RFS traffic to machines with lighter traffic
- Use sticky bit programs across the network
  (See the "Mounting Guidelines" section of this chapter for special rules relating to sharing sticky bit programs.)

A description of all Remote File Sharing tunable parameters and suggested initial settings appear at the end of this section.

The **-D** option of **sar** is used to produce RFS-specific information along with standard **sar** reports (**c**, **u**, and **b** options).

Remote System Calls (sar –Dc)

Your computer collects data each time a system call sends a message across a Remote File Sharing network to access a remote file. You can print this information using **sar -Dc** (see Figure 9-14).

The report produced by **sar -Dc** contains the average system calls per second; average read and write system calls per second, including average characters read and written per second; and average **exec** per second (see Figure 9-14).

Information is divided into three categories: incoming requests (another computer's request for your resources), outgoing requests (your computer's request for a remote resource), and strictly local system calls.

```
$ sar -Dc
lucy lucy 3.0 2 computer   02/14/86

00:00:04  scall/s  sread/s  swrit/s  fork/s  exec/s   rchar/s  wchar/s
01:00:04
   in          4        1        2             0.00       350      220
   out         3        2        1             0.00       240      300
   local     133       30       12     0.73    1.33     11202     3813
02:00:04
   in          4        1        2             0.00       350      220
   out         3        2        1             0.00       240      300
   local     133       30       12     0.73    1.33     11202     3813
03:00:02
   in          4        1        2             0.00       350      220
   out         3        2        1             0.00       240      300
   local     133       30       12     0.73    1.33     11202     3813
04:00:02
   in          4        1        2             0.00       350      220
   out         3        2        1             0.00       240      300
   local     133       30       12     0.73    1.33     11202     3813

Average
   in          4        1        2             0.00       350      220
   out         3        2        1             0.00       240      300
   local     133       30       12     0.73    1.33     11202     3813
$
```

Figure 9-14: Output From **sar -Dc**

NOTE

Some statistics will not reflect the actual number of messages sent across the network, since the client-caching feature allows some remote read requests to be satisfied from data in local buffers. Outgoing scall/s, sread/s, and rchar/s fields include statistics for these read "hits" of remote data in the client cache. Though these reads do not result in actual messages to the remote machine, they are still categorized as outgoing, since they access remote data.

The following paragraphs describe how information from the **sar -Dc** report can be useful to you. If performance is poor, you can see how efficiently system read and write calls to and from your computer are using the Remote File Sharing network. For incoming (in) and outgoing (out) system calls, divide the characters read or written by the reads and writes, respectively.

If your computer is attempting more than about 30 remote system calls per second (in and out scall/s), you are probably nearing capacity. Performance problems will probably result from this much demand. Remote **exec**s also put a heavy demand on a computer. Selective use of sticky bit programs can help improve performance.

You may want to consider moving resources to machines where they are most in demand. (See the **fusage** command to determine what resources are being used most heavily.)

CPU Time (sar -Du)

You can list the percent of total central processing unit (CPU) time spent on system calls from remote computers (%sys remote) with the **sar -Du** command (see Figure 9-15).

```
$ sar -Du
lucy lucy 3.0 2 computer   02/14/86

00:00:04      %usr      %sys     %sys      %wio    %idle
                       local   remote
01:00:04         7        21       10        28       44
02:00:04        11         9       10         4       76
03:00:02         8        18       10        17       57
04:00:02         2         4       10         1       93
05:00:03         1         4       10         1       93
06:00:02         2         5       10         2       91
07:00:02         1         4       10         1       94
08:00:02         2         5       10         2       91
08:20:02        26        16       10        11       48
08:40:02        18        11       10         9       62
09:00:17        25        21       10        13       41
09:20:18        23        21       10        11       45
09:40:20        21        24       10        15       39
10:00:09        21        29       10        17       33
10:20:14        29        28       10        13       31
10:40:18        19        20       10         7       54

Average          9        12       10         8       71
$
```

Figure 9-15: Output From **sar -Du**

If the percent of CPU time spent servicing remote system calls is high, your local users may be suffering. (However, if the computer is a server machine, you would expect %sys remote to be high.)

To reduce the time spent servicing remote requests, you may want to place the resource(s) in demand on another computer [see the **fusage**(1M) command] or limit resource access by changing some of the tunable parameters. (See the section titled "Parameter Tuning.") You may also want to make sure clients are doing I/O in an efficient way (see **sar -Dc**).

Client Caching (sar –Db and sar –C)

The client-caching feature of RFS improves RFS performance by reducing the number of times data is retrieved across the network. With client caching, the first read of data will bring the data into local buffers. Once data is in the local buffer, it will remain there so subsequent reads can get the data locally.

Client caching is assigned by default on a systemwide basis (RCACHE-TIME parameter) and when you mount a remote resource. You will almost always want to take advantage of the improved performance of client caching. There are only two very rare occasions when you may not want to use client caching.

- If buffer space is limited on your system, you may choose to turn off client caching for some resources or the entire system.
- If you are using programs that do their own private network buffering, you may not want to use client caching.

You can produce two **sar** reports to monitor caching activities.

Caching Buffer Usage

The **-b** option of **sar** reports the buffer pool usage for local (disk) reads and writes. The **sar -Db** option reports the same information, plus information on buffer pool usage of locally mounted remote resources (see Figure 9-16).

```
$ sar -Db

charlie charlie 3.1 2 computer    09/03/86

14:37:15 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
14:37:18
   local      2      40      93       1       3      64       0       0
   remote     1      11      92       1       1       0
14:37:21
   local      2      39      92       1       3      63       0       0
   remote     0      10      94       1       1       0
14:37:24
   local      2      40      93       1       3      64       0       0
   remote     1      12      93       1       1       0

Average
   local      2      40      93       1       3      64       0       0
   remote     1      11      93       1       1       0
```

Figure 9-16: Output From **sar -Db**

The fields on this report are as follows:

bread/s The number of read buffer misses per second. (Each miss results in a read message to the server.)

lread/s The number of read cache accesses per second.

%rcache Read cache hit ratio (100 – ((breads)/(lreads) * 100)).

bwrit/s Number of write buffer *misses* per second. All writes are sent to the server. Cache buffers affected by the writes are updated (write-through policy). This field indicates the numbers of lwrites that did not require a write-through. If data did not require a write-through it means that no data affected by the lwrit was present in the cache. (The information in this field has no performance implications when compared to using caching versus not using caching.)

lwrit/s The total remote write cache accesses per second.

%wcache Write cache hit ratio (100 – ((bwrits)/(lwrits) * 100)).

pread/s Not reported for remote use.

pwrit/s Not reported for remote use.

Cache Consistency Overhead

Information on the overhead related to maintaining cache consistency is listed with **sar** -C (see Figure 9-17).

```
$ sar -C

charlie charlie 3.1 2 computer    09/03/86

14:36:56 snd-inv/s snd-msg/s rcv-inv/s rcv-msg/s dis-bread/s blk-inv/s
14:36:59       0.0       1.1       0.0       1.5         0.0       0.2
14:37:02       0.0       0.6       0.0       0.5         0.0       0.4
14:37:05       0.3       0.6       0.0       0.5         0.0       0.1

Average        0.1       0.9       0.0       0.8         0.0       0.2
```

Figure 9-17: Output From **sar** -C

The fields on this report are as follows:

snd-inv/s The number of invalidation messages sent by the server per second to inform client machines about changes to server files.

snd-msg/s The total number of outgoing RFS messages sent per second.

rcv-inv/s The number of invalidation messages received by client from the server. Each message informs the client that the contents of one or more of its cache buffers may have been modified by a write on the server. The client machine reacts by invalidating data in the affected buffers, so the buffers can be used for other purposes.

rcv-msg/s The total number of incoming RFS messages received per second.

dis-bread/s When an invalidation message is received, caching is turned off until the writing process closes or until a time interval has elapsed (set by the tunable parameter RCACHETIME). This counter tracks the number of buffer reads that normally would be eligible for caching in a resource with caching turned on, but that are not added to the buffer pool because caching for this resource is temporarily turned off. It indicates the penalty of running uncached and provides a basis for tuning the RCACHETIME parameter.

blk-inv/s The number of buffers removed from the client cache as a result of receiving an invalidation message while a remote file is open or reopening a remote file that has been modified since the last close on the client.

Server Processes (sar –S)

Every request from a remote computer to access your resources is handled by a server process. When there are too many requests for the servers to handle, they are delayed and placed on the request queue. Requests leave the request queue when servers are available. Information on server availability and requests awaiting service are listed with **sar –S** (see Figure 9-18).

```
$ sar -S

lucy lucy 3.0 2 computer    02/14/86

00:00:04  serv/lo-hi   request   request   server   server
            3 - 6      %busy     avg lgth  %avail   avg avail
01:00:04    3              0         0       100        3
02:00:04    3              0         0       100        3
03:00:04    4             80         8        20        2
04:00:04    6            100        25         0        0

Average    15             50        15        70        2
$
```

Figure 9-18: Output From **sar -S**

As an administrator you can set the number of server processes available to service remote system calls (see "Parameter Tuning"). There are two server variables you can set: MINSERVE and MAXSERVE. MINSERVE is the number of servers that are initially running to service remote requests.

MAXSERVE is the maximum number of servers that may ever exist. If demand goes beyond what the MINSERVE servers can handle, extra servers can be dynamically allocated so the total number of servers can be as high as the value of MAXSERVE. These processes disappear when they are no longer needed.

Information from **sar -S** can be used to tune your server parameters as in Figure 9-19.

Too Few Servers

If the receive queue is almost always busy (request %busy), you may want to raise the number of servers. Here is how to decide the parameter to raise:

- Raise the MAXSERVE if the total average servers is high.
- Raise the MINSERVE if the total average servers is low.

Too Many Servers

If servers are available nearly 100% of the time (server %avail), you may have allocated too many servers. To decide which parameter to lower:

- Check the number of total servers. If this number is near the MINSERVE value, you can lower MINSERVE. Try reducing it by 50% or by the number of idle servers.
- Check the total servers that are idle. If this number is near the MAXSERVE value, you can lower MAXSERVE. Try reducing it by 50%.

Resource Usage (fusage)

You can find out how extensively remote computers are using your resources with the **fusage** command. It reports how many kilobytes were read and written from your resources, broken down by remote computers that have access to the resources. The form for **fusage** for reporting on a resource you have advertised is:

```
fusage advertised-directory
```

where *advertised-directory* is the full pathname to one of the directories you have advertised. The **fusage** command with no options produces a full report of data usage for all disks and advertised directories on your system, as shown in Figure 9-19.

```
# fusage

FILE USAGE REPORT FOR charlie

        /dev/dsk/0s1        /

                            /
                                 charlie       649 KB
                               Clients           0 KB
                                 TOTAL         649 KB

        /dev/dsk/0s3        /usr

                            /usr
                                 charlie       563 KB
                               Clients           0 KB
                                 TOTAL         563 KB
```

Figure 9-19: Output From **fusage**

If a remote computer's requests for your resources are high, it may be causing performance problems on your computer. With the output from **fusage**, you can see what resources are being particularly hard hit. You may then decide to move the resource. You may want to move or copy a resource to a computer that is constantly accessing it.

Remote Disk Space (df)

You can use the standard **df** command with a remote resource name to see the space left on the disk on which the remote resource resides. The form of the command to report on a remote resource is:

df *resource*

where *resource* is the name of a remote resource mounted on your machine. (The **df** command with no options will produce information for all mounted remote resources, plus all locally mounted devices.) Figure 9-20 contains an example of the **df** command using resource names as options.

```
# df USERsrc USERmail

/usr/src     (USERsrc      ):     5436 blocks      2202 i-nodes
/usr/mail    (USERmail     ):     5436 blocks*     2202 i-nodes
```

Figure 9-20: Output From **df**

NOTE

When multiple remote resources are reported that reside on the same disk, all listings of space on that disk, after the first, will be noted with an asterisk.

If you have write permission to a resource, you have as much access to file system space as a user on the system who owns a resource. This command will tell you the potential disk space available for you to write in. (Note that the space reported will only be for the top file system related to each resource.)

Parameter Tuning

There are several parameters you can tune to best suit the way you use Remote File Sharing. The RFS parameters control the amount of resources you devote to RFS service. Each network transport provider may also have some tunable parameters that may affect performance characteristics of that particular network. See the network documentation for your network for more details.

Refer to the "Remote File Sharing Parameters" section in Chapter 5, Customizing Your Computer, for a description of Remote File Sharing tunable parameters.

Appendix A

Appendix A

Appendix A: Control Command Equivalents for the Interface

Appendix A: Control Command Equivalents for the Interface

Control Command Sequences

Figure A-1 defines the control command equivalents for editing and keyboard navigation in the interface. These equivalents can be used from a terminal other than the console keyboard. To issue one of these sequences, hold down the Ctrl (control key) and then press the character. For sequences that include two characters, press and hold the Ctrl key and the first character at the same time, and then press the second character.

KEY	SEQUENCE
Screen Labeled Keys	Ctrl f1— Ctrl f8
Command Line	Ctrl z
Screen Refresh	Ctrl l
Beg	Ctrl b
End	Ctrl e
Home	Ctrl f b
Home Down	Ctrl f e
Up Arrow	Ctrl u
Down Arrow	Ctrl d
Right Arrow	Ctrl r
Left Arrow	Ctrl l

Figure A-1: Control Command Sequences

KEY	SEQUENCE
Page Down	Ctrl w
Page Up	Ctrl V
Scroll Down	Ctrl fd
Scroll Up	Ctrl fu
Tab	Ctrl i
Backtab	Ctrl t
PREV	Ctrl p
NEXT	Ctrl n
Backspace	Ctrl h
Delete Char	Ctrl x
Delete Line	Ctrl k
Insert Char	Ctrl a
Insert Line	Ctrl o
MARK	Ctrl fm

Figure A-1: Control Command Sequences (Cont)

Appendix B

Appendix B

Appendix B: Additional Commands for the Interface

Appendix B: Additional Commands for the Interface

Command Line Editing

Figures B-1 through B-4 contain commands that can be used in AT&T Administration, but the keys do not exist on your computer keyboard.

Command	Purpose
BEG	moves the cursor to the beginning of the line.
DELETE LINE	erases the entire line.
CLEAR EOL	erases from the current position to the end of the line.
HOME DOWN	moves the cursor to the last character on the line.

Figure B-1: Editing on the Command Line

Navigating Within Menus

Command	Purpose
BEG	moves the cursor to the first item on the list, even if it is not displayed.
HOME DOWN	moves the cursor to the last currently visible item in a single-column list and to the bottom-left in a multi-column list.
NEXT	same as → and, in addition, it wraps from the last to first item.
PREV	same as ← and, in addition, it wraps from the first to last item.
SCROLL DOWN	rolls the contents down one line.
SCROLL UP	rolls the contents up one line.

Figure B-2: Navigating Within Menus

Navigating Within Forms

Command	Purpose
HOME DOWN	moves the cursor to the last character of the current field.
BEG	moves the cursor to the first character of the first field in the form.

Figure B-3: Navigating Within Forms

Editing Fields

Command	Purpose
DELETE CHARACTER	deletes the character under the cursor and closes the gap.
DELETE LINE	deletes all fields of the current line.
CLEAR EOL	clears the current line from the current position to the end of the line.

Figure B-4: Editing Fields

Appendix C

Appendix C

Appendix C: Change Other User's Password

Appendix C: Change Other User's Password

Change Password Procedure

To change the password for other users, use the following procedure.

1. Log in and use the **/bin/su** command to obtain **root** permissions by typing:

```
$ /bin/su Enter
```

   If you are already logged in and in the interface, escape the interface and then enter the **/bin/su** command.

2. Type the **root** password and strike Enter.
3. When the # prompt appears, type

```
# passwd <login name> Enter
```

   and respond to the prompts. The <login name> is the login name of the password that you want to change.

4. When prompted for the new password,

```
New password:
```

   enter the new password you want.

   The password you enter will not be displayed on the screen.

You will receive an error message in the following circumstances:

- if you enter the old password incorrectly,
- if the new password is not six character long,
- if the new password does not have two alphabetic characters and at least one special character in the first eight,
- if the password resembles the login name by being a reverse or circular shift,
- if the new password does not differ from the old password by 3 or more characters,
- if the new password includes a space or a ":", or
- if you enter the new password incorrectly the second time.

5. When prompted to repeat the new password,

```
Re-enter new password:
```

type your password again.

If the two password entries are the same, the password is assigned. If the two password entries don't match, the message

```
They don't match; try again.
New password:
```

appears. If this message appears, type the new password again and then re-enter the new password again.

Appendix D

Appendix D

Appendix D: Adding Basic Networking

Appendix D: Adding Basic Networking

Basic Networking Procedures

Adding Basic Networking involves doing the following basic steps.

1. Choose to physically connect your computer to another computer by adding one of the following:

   A Direct Link — Physically connect null modem cable from the built-in serial port on your computer to a port on another computer. See "Physical Connection of computer to DTE Direct Link" in this Appendix for details.

   A Modem — For the modem selected, set the appropriate options per "Recommended Switch Settings" in this Appendix or per modem documentation. Then physically connect the modem using RS-232 connectors with customized modem cable to the computer. See "Physical Connection of computer to Modem (DCE)" in this Appendix for details.

2. Logically connect the modem or direct link to the UNIX operating system. This involves using the administration menu to update the appropriate support files to reflect the presence of a direct link or modem. See "Basic Networking Software and Direct Links" in this Appendix and Chapter 8, Basic Networking Administration, for details.

Direct Links and Modems

This section discusses the following configurations:

- computer to Data Terminal Equipment (DTE) direct link
- AT&T computer to Data Communications Equipment (DCE) such as a modem.

Your computer will connect with any other machine with an RS-232 port. The modems supported with your computer will be any kind of auto dial modems.

An advantage of using a direct link is that the link is always available and the time required to access the link is short. Direct links would be beneficial only when:

- The two machines transfer large amounts of data on a regular basis
- The two machines are located no more than several hundred cable feet apart.

NOTE The procedure for setting up direct links through additional serial ports on expansion cards may differ. Refer to the expansion card documentation for that information.

The amount of cable used to link two machines is dependent on the environment the cable is run. The standard for RS-232 connections is 50 feet or less. As the cable length is increased, noise on the lines may become a problem. This means that the transmission rate must be decreased or limited distance modems should be placed on each end of the line. Normally, you should not use more than 1000 cable feet to connect the two machines.

The advantage of using a modem is that a port is not dedicated to only one computer. You can also be networked to a remote computer located anywhere in the world where the telephone network exists. The disadvantages are that the port of the remote computer is often busy and the transmission rate is slower.

Physical Connection of Computer to DTE Direct Link

Connecting a computer to another (DTE) RS-232-C device (e.g., another computer, UNIX PC, etc.) requires the use of a null-modem cable that must be constructed as follows:

Pin 1 to 1
Pin 2 to 3
Pin 3 to 2
Strap pin 4 to 5 in the same plug
Pin 6 to 20
Pin 7 to 7
Pin 8 to 20
Pin 20 to 6
Pin 20 to 8.

In Figure D-1, **in** means external source and **out** means computer is source. In wiring asynchronous cables from modem to built-in port, the pins have the following meanings:

Pin	Description
1	Frame Ground
2	Data **in**to computer
3	Data **out** of computer
4	Clear to send **in** (must be positive to emit data—will float positive)
5	Request to send **out** (normally positive)
6	Data terminal ready **out** (operates if minor device number includes 128)
7	Signal Ground
8	Data Carrier detect **in** (must be positive to receive data—will float positive)
20	Data set ready **in**

Figure D-1: Pin Descriptions for Null-Modem and Modem Cable

Wiring for Direct Link

Null-modem cables are commercially available for the direct link. If you desire to customize your own null-modem cable, nine leads must be wired for a connection to be made as shown in Figure D-2. Do not attach wiring to unused signals.

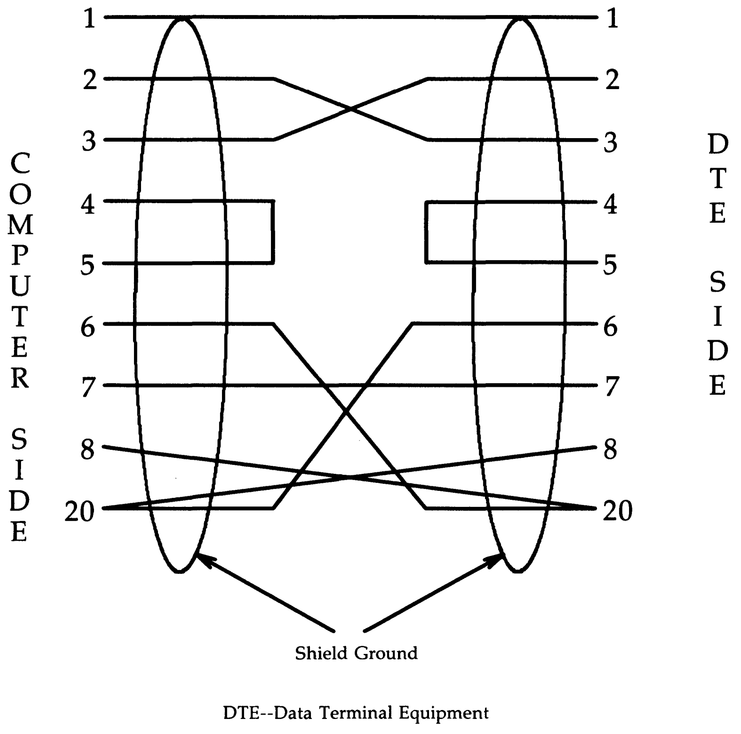


Figure D-2: Connector Wiring Diagram for DTE Direct Link to Computer

Figure D-3 shows a simple illustration of how an computer connects to a DTE direct link.

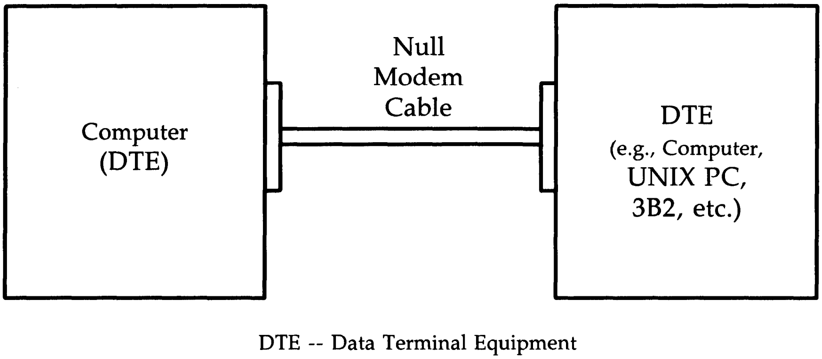

Computer (DTE)
Null Modem Cable
DTE
(e.g., Computer, UNIX PC, 3B2, etc.)
DTE -- Data Terminal Equipment

Figure D-3: Physical Connection of Computer to DTE Device (Direct Link)

Basic Networking Software and Direct Links

Ideally, systems that have a direct link should run common and current releases of the UNIX system to have the full set of capabilities available. (Bidirectional ports that are supported by the **uugetty** program were introduced with UNIX System V, Release 2.0, Version 1.) However, lack of commonality does not prevent utilization of the Basic Networking feature. This section describes the software files that must be modified on your computer in order to accommodate a direct link connection. You may want to consult the documentation provided with your machine if you're linking directly to a remote machine other than an computer.

The following support files must be updated to reflect the presence of a Direct Link:

- **/usr/lib/uucp/Devices**
- **/etc/inittab**
- **/usr/lib/uucp/Systems**

Refer to the "Setting Up Mail" in Chapter 4, System Administration, for information on setting up these files.

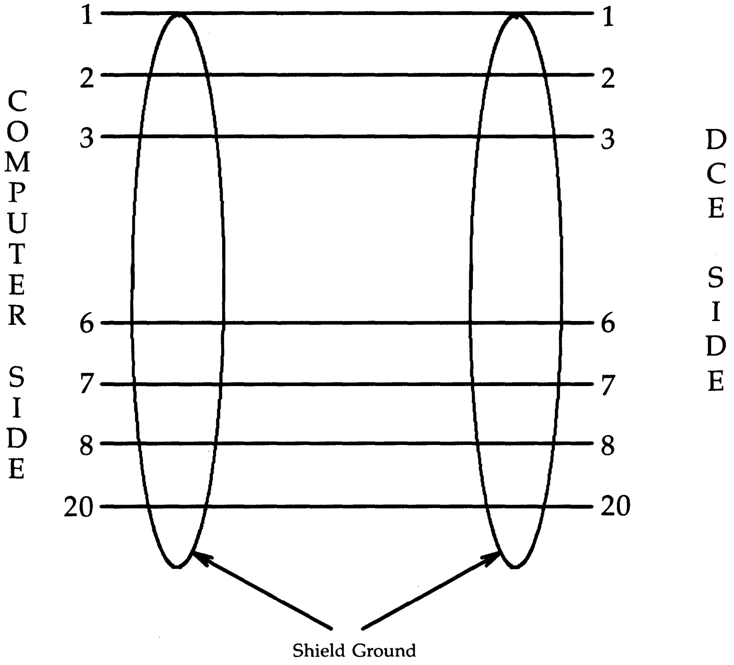
Physical Connection of Computer to Modem (DCE)

A DCE device such as a modem can connect to your computer with an RS-232 cable. The computer's serial connector must have a DTE configuration and the modem is required to have a DCE configuration. The pin descriptions for modem cable are shown is Figure D-1. The following are the pin connections for the RS-232 modem cable:

Pin 1 to 1
Pin 2 to 2
Pin 3 to 3
Pin 6 to 6
Pin 7 to 7
Pin 8 to 8
Pin 20 to 20.

Wiring for Modems

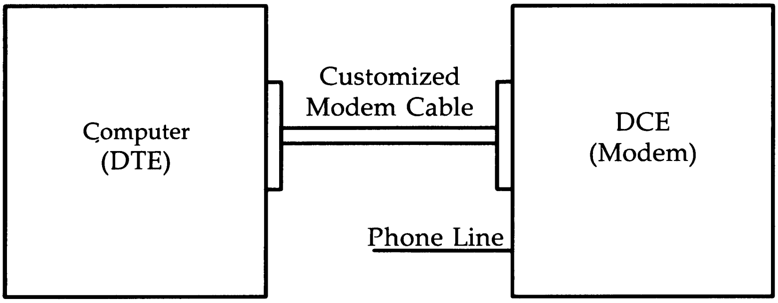
Wire to pins only used at both ends. Do not attach wiring to unused signal. Seven leads must be wired to customize modem cable as shown in Figure D-4.



DCE--Data Communication Equipment

Figure D-4: Connector Wiring Diagram for Connecting Modem to Computer

Figure D-5 shows a simple illustration of how an computer connects to a DCE device such as a modem.



DTE--Data Terminal Equipment

DCE--Data Communication Equipment

Figure D-5: Physical Connection of Computer to DCE Device

Setting Up Modems

Initial Modem Installation

An initial modem setup occurs the first time the modem is installed. The steps that should be followed are:

1. Set up the modem option switches.
2. Make sure the modem power switch is off.
3. Mechanically connect (i.e., attach the cables) the modem to the computer and the telephone line.
4. Plug the modem in.
5. Turn on the power to the modem.
6. Do procedure Serial Port Setup found in Chapter 4, System Administration to set up the serial port.
7. Return to Appendix D, Adding Basic Networking after you have set up the serial port.

The software commands necessary to configure the modem will now be executed. These commands are sent to the modem, so the modem must be connected to the computer and the power to the modem must be on. In addition, a number of configuration files on the computer must be modified. These steps may take a minute.

In addition to initializing the modem during the Serial Ports Setup, the modem needs to be initialized at boot time. This may add up to 15 seconds to the system boot time.

Reinitializing the Modem

There are circumstances which may require the modem to be reinitialized. This may happen if the modem loses power after the system is booted. If, for example, the modem does not automatically answer calls when it is configured as a Host or Both Caller and Host, if the speaker is active, if the modem dials via pulse dialing, or if it just doesn't seem to be working correctly, it may be necessary to reinstall.

To reinstall, try the following:

1. Disconnect the power to the modem.
2. From the Administration menu, highlight Peripherals Setup and strike Enter.
3. From the Peripherals Setup menu, highlight Serial Ports Setup and strike Enter.
4. From the Serial Ports Setup form, move the cursor to the "Device Type:" field and strike the CHOICES function key.
5. From the Device Types pop-up menu, highlight None and strike Enter.
6. Strike SAVE. You'll return to the Peripherals Setup menu.
7. Reconnect and turn the modem power on.

8. From the Peripherals Setup menu, highlight Serial Ports Setup and strike Enter.
9. With the cursor resting on the "Serial Port Number" field, strike CHOICES.
10. From the Port Number pop-up menu, highlight the serial port number you want and strike Enter.
11. Move the cursor to the "Device Speed:" field and strike CHOICES.
12. From the Device Speed pop-up menu, highlight the device speed you want and strike Enter.
13. Move the cursor to the "Device Type:" field and strike the CHOICES function key.
14. From the Device Types pop-up menu, highlight Modem and strike Enter.
15. Strike the SAVE function key.
16. From the Connect to Modem form, strike the CHOICES function key.
17. From the Modems pop-up menu, highlight the modem name you want and strike Enter.
18. From the Device Connection menu, highlight the correct device connection you want and strike Enter.
19. Strike the SAVE function key.
20. When the confirmation message appears telling you the serial port is now set up for a modem, strike CONT.

These steps are necessary since many modems require options that are set by the software. These steps send the options that are set by the software to the modem when the modem loses them because of a loss of power or other circumstances.

Recommended Switch Settings

The following modems are hardware-configured. They have switch settings that must be manually set in order for the modems to work correctly. Modems that are software-configured will have the switch settings automatically set through the software setup scripts.

Hardware configured modems that have a carrier detect (CD) switch must have that switch set low or off.

AT&T 2212C

- Caller Only

  Internal switches set to factory defaults

- Host Only

  Internal switches set to factory defaults

- Both Host and Caller

  Internal switches set to factory defaults.

AT&T 2224B

- Caller Only

  All switches set to factory defaults

- Host Only

  All switches set to factory defaults

- Both Host and Caller

  All switches set to factory defaults.

NOTE

The volume control for the speaker is located behind the front panel. If the speaker comes on when the modem is in use, flip open the front panel and move the speaker loudness control to its lowest position.

When installing this modem or changing the speed, be sure that the speed indicated on the front panel switch matches the speed set in the menu. The modem will originate calls at the speed indicated in the menu. However, it will only answer calls at the speed indicated by the front panel switch. If the menu and front panel switch do not agree, calls may be lost without there being any other indication of an error.

AT&T 4000 Models 1A01, 1A02, 4024

There are no hardware switches in the standalone (external) AT&T 4000 modems. All options are set in software by the setup scripts. The AT&T 4000 1A01 and 1A02 modems are two different versions of the same modem. Both of these modems are supported by the computer. The version number may be found on the bottom of the modem.

AT&T 4000 Model 4112

When configuring this modem use the COM2 device. The following are the 4 switch settings for the 4112 modem:

1. off
2. off
3. off
4. on Strap CD off.

Hayes SMARTMODEM 1200

- Caller Only

1.	up	Modem disconnects on loss of DTR
2.	up	English result codes
3.	down	Send result codes
4.	up	Echo characters
5.	down	Auto answer disable
6.	up	Strap CD off
7.	up	See manual; up = single line
8.	down	Enable command recognition.

- Host Only

1.	up	Modem disconnects on loss of DTR
2.	up	English results codes
3.	down	Send result codes
4.	up	Echo characters
5.	up	Auto answer enabled
6.	up	Strap CD off
7.	up	See manual; up = single line
8.	down	Enable command recognition.

- Both Host and Caller

1.	up	Modem disconnects on loss of DTR
2.	up	English result codes
3.	down	Send results codes
4.	up	Echo characters
5.	up	Auto answer enabled
6.	up	Strap CD off
7.	up	See manual; up = single line
8.	down	Enable command recognition.

NOTE

The Hayes SMARTMODEM 1200 must be reinstalled if the modem loses power after the computer is rebooted.

Hayes SMARTMODEM 2400

There are no option switches on the Hayes SMARTMODEM 2400.

NOTE

The Hayes SMARTMODEM 2400 must be reinstalled if the modem loses power after the computer is booted.

Penril 300/1200 AD

- Caller Only
    - A. Front panel switches
        1. The HS button should be struck.
        2. All other buttons should be out.
    - B. The internal switches should all be set to the factory defaults.
- Host Only
    - A. Front panel switches
        1. The HS button should be struck.
        2. All other buttons should be out.
    - B. The internal switches should all be set to the factory defaults.

- Both Host and Caller

  A. Front panel switches

  1. The HS button should be struck.

  2. All other buttons should be out.

  B. The internal switches should all be set to factory defaults.

Ventel EC1200-31

There are several modems manufactured by Ventel that are marketed under the model number EC1200-31. These modems can be supported using factory default settings or by setting the Hayes compatible "AT" dialer switch. With either mode you choose, the carrier detect switch (CD) must be set low.

- Caller Only

A. External Switch Settings

1.	open	Use factory default.
2.	open	Use factory default.
3.	open	Use factory default.
4.	open	Use factory default.

B. Internal Switch Settings

1.	open	Modem disconnects on loss of DTR.
2.	close	Strap CD on only when carrier is present.
3.	open	Enables Ventel compatible command recognition.
4.	open	Speaker off.

- Host Only

A. External Switch Settings

1. open Use factory default.
2. open Use factory default.
3. open Use factory default.
4. open Use factory default.

B. Internal Switch Settings

1. open Modem disconnects on loss of DTR.
2. close Strap CD off.
3. open Enables Ventel compatible command recognition.
4. open Speaker off.

- Both Host and Caller

  A. External Switch Settings

  1. open Use factory default.
  2. open Use factory default.
  3. open Use factory default.
  4. open Use factory default.

  B. Internal Switch Settings

  1. open Modem disconnects on loss of DTR.
  2. close Strap CD off. The light will come on only when a valid carrier is detected by the modem.
  3. close Hayes compatible AT dialer enabled. Modem assumes Hayes verbose on power up.
  4. open Speaker off.

NOTE

When the Ventel EC1200-31 is powered on, the MB/HO light will sometimes stay on. The modem should be powered off then on until this light is out.

Ventel 1200-EC

- Caller Only

A. External Switch Settings

1.	open	English result codes
2.	open	Send result codes
3.	open	Echo characters
4.	close	Don't auto answer.

B. Internal Switch Settings

1.	open	Modem disconnects on loss of DTR.
2.	open	Strap CD on only when carrier is present.
3.	close	Enables Hayes compatible command recognition.
4.	open	Speaker off.

- Host Only

A. External Switch settings

1.	open	Don't care - use factory default.
2.	close	Don't send result codes.
3.	close	Don't echo character.
4.	open	Auto answer.

B. Internal Switch Settings

1.	open	Modem disconnects on loss of DTR.
2.	open	CD on only when carrier is present.
3.	close	Don't care - use factory default.
4.	open	Speaker off.

C. Both Host and Caller

D. External Switch Settings

1.	open	English result codes
2.	open	Send result codes
3.	open	Echo characters
4.	open	Auto answer.

E. Internal Switch Settings

1	open	Modem disconnects on loss of DTR.
2.	open	Strap CD on only when carrier is present.
3.	close	Enables Hayes compatible command recognition.
4.	open	Speaker off.

NOTE

When the Ventel 1200-EC is powered on, the MB/OH light will sometimes stay on. The modem should be powered off then on until this light is out.

Ventel MD212

- Caller Only

  A. All front panel switches should be out.

  B. All internal switches should be set to the factory defaults.

- Host Only

  A. All front panel switches should be out.

  B. All internal switches should be set to the factory defaults.

- Both Host and Caller

  A. All front panel switches should be out.

  B. All internal switches should be set to the factory defaults.

Appendix E

Appendix E

Appendix E: fsck Error Messages

Appendix E: fsck Error Messages

The following are the error messages you might receive when using **fsck**.

Initialization

Before a file system check can be performed, certain tables have to be set up and certain files opened. This section describes the opening of files and the initialization of tables. Error conditions resulting from command line options, memory requests, opening of files, status of files, file system size checks, and creation of the scratch file are listed below. The **fsck** program terminates on initialization errors.

Legal Options

Legal **fsck** options are *-f, -b, -y, -n, -s, -S, -t, -q, -b,* and *-D*. The *-y* option is recommended for **fsck**. This option will answer yes to all questions prompted by **fsck** and requires no intervention by you. Another recommended option is *-s* which forces rebuilding the free list in optimal order. The free list gets disorganized with use. Rebuilding the free list improves performance on subsequence created files. Use the following command line for **fsck**:

```
/etc/fsck -s -y file_system_name
```

See the **fsck**(1M) command in the *User's/System Administrator's Reference Manual* for additional information.

Bad -t option

The **-t** option is not followed by a filename. The **fsck** program terminates on this error condition.

Invalid -s argument, defaults assumed

The **-s** option is not suffixed by 3, 4, or blocks-per-cylinder:blocks-to-skip. The **fsck** program assumes a default of 400 blocks-per-cylinder and 7 blocks-to-skip.

Incompatible options: -n and -s

It is not possible to salvage the free-block list without modifying the file system. The **fsck** program terminates on this error condition.

Cannot fstat standard input

The attempt to **fstat** standard input failed. This error condition indicates a serious problem that may require additional assistance. The **fsck** program terminates on this error condition.

Cannot get memory

The request for memory for virtual memory tables failed. This error condition indicates a serious problem that may require additional assistance. The **fsck** program terminates on this error condition.

Cannot open checklist file: F

The default file system **checklist** file *F* (usually **/etc/checklist**) cannot be opened for reading. The **fsck** program terminates on this error condition. Check access modes of *F*.

Cannot stat root

The request for statistics about the root directory "/" failed. This error condition indicates a serious problem that may require additional assistance. The **fsck** program terminates on this error condition.

Cannot stat F

The request for statistics about the file system *F* failed. The **fsck** program ignores this file system and continues checking the next file system given. Check access modes of *F*.

F is not a block or character device

The **fsck** program has been given a regular filename by mistake. It ignores this argument and continues checking the next file system given. Check the file type of *F*.

Cannot open F

The file system *F* cannot be opened for reading. The **fsck** program ignores this and continues checking the next file system given. Check the access modes of *F*.

Size check: fsize X isize Y

More blocks are used for the i-node list *Y* than there are blocks in the file system *X*, or there are more than 65,535 i-nodes in the file system. The **fsck** program ignores this file system and continues checking the next file system given.

Cannot create F

The request to create a scratch file *F* failed. The **fsck** program ignores this file system and continues checking the next file system given. Check the access modes of *F*.

CAN NOT SEEK: BLK B (CONTINUE)

The request for moving to a specified block number *B* in the file system failed. The occurrence of this error condition indicates a serious problem that may require additional assistance.

Possible responses to CONTINUE prompt are:

YES	Attempt to continue to run file system check. Often, however, the problem persists. This error condition does not allow a complete check of the file system. A second run of **fsck** should be made to recheck this file system. If the block was part of the virtual memory buffer cache, **fsck** will terminate with the message "Fatal I/O error."
NO	Terminate program.

CAN NOT READ: BLK B (CONTINUE)

The request for reading a specified block number *B* in the file system failed. The occurrence of this error condition indicates a serious problem that may require additional assistance.

Possible responses to CONTINUE prompt are:

YES Attempt to continue to run file system check. Often, however, the problem persists. This error condition does not allow a complete check of the file system. A second run of **fsck** should be made to recheck this file system. If block was part of the virtual memory buffer cache, **fsck** will terminate with the message "Fatal I/O error."

NO Terminate program.

CAN NOT WRITE: BLK B (CONTINUE)

The request for writing a specified block number *B* in the file system failed. The file system should not be opened for writing.

Possible responses to CONTINUE prompt are:

YES Attempt to continue to run file system check. Often, however, the problem persists. This error condition does not allow a complete check of the file system. A second run of **fsck** should be made to recheck this file system. If block was part of the virtual memory buffer cache, **fsck** terminates with the message "Fatal I/O error."

NO Terminate program.

Phase 1: Check Blocks and Sizes

This phase concerns itself with the i-node list. This part lists error conditions resulting from checking i-node types, setting up the zero-link-count table, examining i-node block numbers for bad or duplicate blocks, checking i-node size, and checking i-node format.

UNKNOWN FILE TYPE I=I (CLEAR)

The mode word of the i-node *I* indicates that the i-node is not a special character i-node, regular i-node, or directory i-node.

Possible responses to CLEAR prompt are:

YES	Deallocate i-node *I* by zeroing its contents. This invokes the UNALLOCATED error condition in Phase 2 for each directory entry pointing to this i-node.
NO	Ignore this error condition.

LINK COUNT TABLE OVERFLOW (CONTINUE)

An internal table for **fsck** containing allocated i-nodes with a link count of zero has no more room.

Possible responses to CONTINUE prompt are:

YES	Continue with program. This error condition does not allow a complete check of the file system. A system run of **fsck** should be made to recheck this file system. If another allocated i-node with a zero link count is found, this error condition will be repeated.
NO	Terminate the program.

B BAD I=I

I-node *I* contains block number *B* with a number lower than the number of the first data block in the file system or greater than the number of the last block in the file system. This error condition may invoke the EXCESSIVE BAD BLKS error condition in Phase 1 if i-node *I* has too many block numbers outside the file system range. This error condition invokes the BAD/DUP error condition in Phase 2 and Phase 4.

EXCESSIVE BAD BLKS I=I (CONTINUE)

There is more than a tolerable number (usually 10) of blocks claimed by other i-nodes.

Possible responses to CONTINUE prompt are:

YES Ignore the rest of the blocks in this i-node and continue to check using the next i-node in the file system. This error condition does not allow a complete check of the file system. A second run of **fsck** should be made to recheck this file system.

NO Terminate the program.

B DUP I=I

I-node *I* contains block number *B* that is already claimed by another i-node. This error condition may invoke the EXCESSIVE DUP BLKS error condition in Phase 1 if i-node I has too many block numbers claimed by other i-nodes. This error condition invokes Phase 1B and the BAD/DUP error condition in Phase 2 and Phase 4.

EXCESSIVE DUPS BLKS I=I (CONTINUE)

There is more than a tolerable number (usually 10) of blocks claimed by other i-nodes.

Possible responses to CONTINUE prompt are:

YES Ignore the rest of the blocks in this i-node and continue to check using the next i-node in the file system. This error condition does not allow a complete check of the file system. A second run of **fsck** should be made to recheck this file system.

NO Terminate the program.

DUP TABLE OVERFLOW (CONTINUE)

An internal table in **fsck** containing duplicate block numbers has no more room.

Possible responses to CONTINUE prompt are:

YES	Continue with program. This error condition does not allow a complete check of the file system. A second run of **fsck** should be made to recheck this file system. If another duplicate block is found, this error condition will repeat.
NO	Terminate the program.

POSSIBLE FILE SIZE ERROR I=I

The i-node *I* size does not match the actual number of blocks used by the i-node. This is only a warning. If the **-q** option is used, this message will not print.

DIRECTORY MISALIGNED I=I

The size of a directory i-node is not a multiple of the size of a directory entry (usually 16). This is only a warning. If the **-q** option is used, this message will not print.

PARTIALLY ALLOCATED INODE I=I (CLEAR)

I-node *I* is neither allocated nor unallocated.

Possible responses to CLEAR prompt are:

YES	Deallocate i-node *I* by zeroing its contents.
NO	Ignore this error condition.

Phase 1B: Rescan for More DUPS

When a duplicate block is found in the file system, the file system is rescanned to find the i-node that previously claimed that block. This part lists the error condition when the duplicate block is found.

B DUP I=I

I-node *I* contains block number *B* that is already claimed by another i-node. This error condition invokes the BAD/DUP error condition in Phase 2. I-nodes with overlapping blocks may be determined by examining this error condition and the DUP error condition in Phase 1.

Phase 2: Check Path Names

This phase concerns itself with removing directory entries pointing to error-conditioned i-nodes from Phase 1 and Phase 1B. This part lists error conditions resulting from root i-node mode and status, directory i-node pointers in range, and directory entries pointing to bad i-nodes.

ROOT INODE UNALLOCATED. TERMINATING

The root i-node (always i-node number 2) has no allocated mode bits. The occurrence of this error condition indicates a serious problem that may require additional assistance. The program stops.

ROOT INODE NOT DIRECTORY (FIX)

The root i-node (usually i-node number 2) is not directory i-node type.

Possible responses to FIX prompt are:

YES Replace the root i-node type to be a directory. If the root i-node data blocks are not directory blocks, a *very* large number of error conditions will be produced.

NO Terminate the program.

DUPS/BAD IN ROOT INODE (CONTINUE)

Phase 1 or Phase 1B has found duplicate blocks or bad blocks in the root i-node (usually i-node number 2) for the file system.

Possible responses to CONTINUE prompt are:

YES Ignore DUPS/BAD error condition in root i-node and attempt to continue to run the file system check. If root i-node is not correct, then this may result in a large number of other error conditions.

NO Terminate the program.

I OUT OF RANGE I=I NAME=F (REMOVE)

A directory entry *F* has an i-node number *I* that is greater than the end of the i-node list.

Possible responses to REMOVE prompt are:

YES	The directory entry *F* is removed.
NO	Ignore this error condition.

UNALLOCATED I=I OWNER=O MODE=M SIZE=S MTIME=T NAME=F (REMOVE)

A directory entry *F* has an i-node *I* without allocate mode bits. The owner *0*, mode *M*, size *S*, modify time *T*, and filename *F* are printed. If the file system is not mounted and the **-n** option is not specified, the entry will be removed automatically if the i-node it points to is size 0.

Possible responses to REMOVE prompt are:

YES	The directory entry *F* is removed.
NO	Ignore this error condition.

DUP/BAD I=I OWNER=O MODE=M SIZE=S MTIME=T DIR=F (REMOVE)

Phase 1 or Phase 1B has found duplicate blocks or bad blocks associated with directory entry *F*, i-node *I*. The owner *O*, mode *M*, size *S*, modify time *T*, and filename *F* are printed.

Possible responses to REMOVE prompt are:

YES	The directory entry *F* is removed.
NO	Ignore this error condition.

DUP/BAD I=I OWNER=O MODE=M SIZE=S MTIME=T FILE=F (REMOVE)

Phase 1 or Phase 1B has found duplicate blocks or bad blocks associated with directory entry *F*, i-node *I*. The owner *O*, mode *M*, size *S*, modify time *T*, and filename *F* are printed.

Possible responses to REMOVE prompt are:

YES	The directory entry *F* is removed.
NO	Ignore this error condition.

BAD BLK B IN DIR I=I OWNER=O MODE=M SIZE=S MTIME=T

This message only occurs when the **-q** option is used. A bad block was found in DIR i-node *I*. Error conditions looked for in directory blocks are nonzero padded entries, inconsistent "." and ".." entries, and embedded slashes in the name field. This error message indicates that the user should at a later time either remove the directory i-node if the entire block looks bad or change (or remove) those directory entries that look bad.

Phase 3: Check Connectivity

This phase concerns itself with the directory connectivity seen in Phase 2. This part lists error conditions resulting from unreferenced directories and missing or full **lost+found** directories.

UNREF DIR I=I OWNER=O MODE=M SIZE=S MTIME=T (RECONNECT)

The directory i-node *I* was not connected to a directory entry when the file system was traversed. The owner *O*, mode *M*, size *S*, and modify time *T* of directory i-node *I* are printed. The **fsck** program forces the reconnection of a nonempty directory.

Possible responses to RECONNECT prompt are:

YES Reconnect directory i-node *I* to the file system in directory for lost files (usually **lost+found**). This may invoke **lost+found** error condition in Phase 3 if there are problems connecting directory i-node *I* to **lost+found**. This may also invoke CONNECTED error condition in Phase 3 if link was successful.

NO Ignore this error condition. This invokes UNREF error condition in Phase 4.

SORRY, NO lost+found DIRECTORY

There is no **lost+found** directory in the root directory of the file system; **fsck** ignores the request to link a directory in **lost+found**. This invokes the UNREF error condition in Phase 4. Check access modes of **lost+found**.

SORRY, NO SPACE IN lost+found DIRECTORY

There is no space to add another entry to the **lost+found** directory in the root directory of the file system; **fsck** ignores the request to link a directory in **lost+found**. This invokes the UNREF error condition in Phase 4. Clean out unnecessary entries in **lost+found** or make **lost+found** larger.

DIR I=I1 CONNECTED, PARENT WAS I=I2

This is an advisory message indicating a directory i-node *I1* was successfully connected to the **lost+found** directory. The parent i-node *I2* of the directory i-node *I1* is replaced by the i-node number of the **lost+found** directory.

Phase 4: Check Reference Counts

This phase concerns itself with the link count information seen in Phase 2 and Phase 3. This part lists error conditions resulting from unreferenced files; missing, or full **lost+found** directory; incorrect link count for files, directories, special files; unreferenced files and directories; bad and duplicate blocks in files and directories; and incorrect total free-i-node counts.

UNREF FILE I=I OWNER=O MODE=M SIZE=S MTIME=T (RECONNECT)

I-node *I* was not connected to a directory entry when the file system was traversed. The owner *O*, mode *M*, size *S*, and modify time *T* of i-node *I* are printed. If the **-n** option is omitted and the file system is not mounted, empty files will be cleared automatically. Nonempty directories are not cleared.

Possible responses to RECONNECT prompt are:

YES Reconnect i-node *I* to file system in the directory for lost files (usually **lost+found**). This can cause a **lost+found** error condition in Phase 4 if there are problems connecting i-node *I* to **lost+found**.

NO Ignore this error condition. This invokes a CLEAR error condition in Phase 4.

SORRY. NO lost+found DIRECTORY

There is no **lost+found** directory in the root directory of the file system; **fsck** ignores the request to link a file in **lost+found**. This invokes the CLEAR error condition in Phase 4. Check access modes of **lost+found**.

SORRY. NO SPACE IN lost+found DIRECTORY

There is no space to add another entry to the **lost+found** directory in the root directory of the file system; **fsck** ignores the request to link a file in **lost+found**. This invokes the CLEAR error condition in Phase 4. Check size and contents of **lost+found**.

(CLEAR)

The i-node mentioned in the immediately previous error condition cannot be reconnected.

Possible responses to CLEAR prompt are:

YES	Deallocate i-node mentioned in the immediately previous error condition by zeroing its contents.
NO	Ignore this error condition.

LINK COUNT FILE I=I OWNER=O MODE=M SIZE=S MTIME=T COUNT=X SHOULD BE Y (ADJUST)

The link count for i-node *I*, that is a file, is *X* but should be *Y*. The owner *O*, mode *M*, size *S*, and modify time *T* are printed.

Possible responses to ADJUST prompt are:

YES	Replace link count of file i-node *I* with *Y*.
NO	Ignore this error condition.

LINK COUNT DIR I=I OWNER=O MODE=M SIZE=S MTIME=T COUNT=X SHOULD BE Y (ADJUST)

The link count for i-node *I*, that is a directory, is *X* but should be *Y*. The owner *O*, mode *M*, size *S*, and modify time *T* of directory i-node *I* are printed.

Possible responses to ADJUST prompt are:

YES	Replace link count of directory i-node *I* with *Y*.
NO	Ignore this error condition.

LINK COUNT F I=I OWNER=O MODE=M SIZE=S MTIME=T COUNT=X SHOULD BE Y (ADJUST)

The link count of *F* i-node *I* is X but should be *Y*. The filename *F*, owner *O*, mode *M*, size *S*, and modify time *T* are printed.

Possible responses to ADJUST prompt are:

YES	Replace link count of i-node *I* with *Y*.
NO	Ignore this error condition.

UNREF FILE I=I OWNER=O MODE=M SIZE=S MTIME=T (CLEAR)

I-node *I*, that is a file, was not connected to a directory entry when the file system was traversed. The owner *O*, mode *M*, size *S*, and modify time *T* of i-node *I* are printed. If the **-n** option is omitted and the file system is not mounted, empty files will be cleared automatically. Nonempty directories are not cleared.

Possible responses to CLEAR prompt are:

YES	Deallocate i-node *I* by zeroing its contents.
NO	Ignore this error condition.

UNREF DIR I=I OWNER=O MODE=M SIZE=S MTIME=T (CLEAR)

I-node *I*, that is a directory, was not connected to a directory entry when the file system was traversed. The owner *O*, mode *M*, size *S*, and modify time *T* of i-node *I* are printed. If the **-n** option is omitted and the file system is not mounted, empty files will be cleared automatically. Nonempty directories are not cleared.

Possible responses to CLEAR prompt are:

YES	Deallocate i-node *I* by zeroing its contents.
NO	Ignore this error condition.

BAD/DUP FILE I=I OWNER=O MODE=M SIZE=S MTIME=T (CLEAR)

Phase 1 or Phase 1B has found duplicate blocks or bad blocks associated with the file i-node *I*. The owner *O*, mode *M*, size *S*, and modify time *T* of i-node *I* are printed.

Possible responses to CLEAR prompt are:

YES	Deallocate i-node *I* by zeroing its contents.
NO	Ignore this error condition.

BAD/DUP DIR I=I OWNER=O MODE=M SIZE=S MTIME=T (CLEAR)

Phase 1 or Phase 1B has found duplicate blocks or bad blocks associated with directory i-node *I*. The owner *O*, mode *M*, size *S*, and modify time *T* of i-node *I* are printed.

Possible responses to CLEAR prompt are:

YES	Deallocate i-node *I* by zeroing its contents.
NO	Ignore this error condition.

FREE INODE COUNT WRONG IN SUPERBLK (FIX)

The actual count of the free i-nodes does not match the count in the super-block of the file system. If the **-q** option is specified, the count will be fixed automatically in the super-block.

Possible responses to FIX prompt are:

YES	Replace count in super-block by actual count.
NO	Ignore this error condition.

Phase 5: Check Free List

This phase concerns itself with the free-block list. This part lists error conditions resulting from bad blocks in the free-block list, bad free-block count, duplicate blocks in the free-block list, unused blocks from the file system not in the free-block list, and the total free-block count incorrect.

EXCESSIVE BAD BLKS IN FREE LIST (CONTINUE)

The free-block list contains more than a tolerable number (usually 10) of blocks with a value less than the first data block in the file system or greater than the last block in the file system.

Possible responses to CONTINUE prompt are:

YES	Ignore rest of the free-block list and continue execution of **fsck**. This error condition will always invoke "BAD BLKS IN FREE LIST" error condition in Phase 5.
NO	Terminate the program.

EXCESSIVE DUP BLKS IN FREE LIST (CONTINUE)

The free-block list contains more than a tolerable number (usually 10) of blocks claimed by i-nodes or earlier parts of the free-block list.

Possible responses to CONTINUE prompt are:

YES	Ignore the rest of the free-block list and continue execution of **fsck**. This error condition will always invoke "DUP BLKS IN FREE LIST" error condition in Phase 5.
NO	Terminate the program.

BAD FREEBLK COUNT

The count of free blocks in a free-list block is greater than 50 or less than 0. This error condition will always invoke the "BAD FREE LIST" condition in Phase 5.

X BAD BLKS IN FREE LIST

X blocks in the free-block list have a block number less than the first data block in the file system or greater than the last block in the file system. This error condition will always invoke the "BAD FREE LIST" condition in Phase 5.

X DUP BLKS IN FREE LIST

X blocks claimed by i-nodes or earlier parts of the free-list block were found in the free-block list. This error condition will always invoke the "BAD FREE LIST" condition in Phase 5.

X BLK(S) MISSING

X blocks unused by the file system were not found in the free-block list. This error condition will always invoke the "BAD FREE LIST" condition in Phase 5.

FREE BLK COUNT WRONG IN SUPERBLOCK (FIX)

The actual count of free blocks does not match the count in the superblock of the file system.

Possible responses to FIX prompt are:

YES	Replace count in super-block by actual count.
NO	Ignore this error condition.

BAD FREE LIST (SALVAGE)

Phase 5 has found bad blocks in the free-block list, duplicate blocks in the free-block list, or blocks missing from the file system. If the **-q** option is specified, the free-block list will be salvaged automatically.

Possible responses to SALVAGE prompt are:

YES	Replace actual free-block list with a new free-block list. The new free-block list will be ordered to reduce the time spent by the disk rotating into position.
NO	Ignore this error condition.

Phase 6: Salvage Free List

This phase concerns itself with the free-block list reconstruction. This part lists error conditions resulting from the blocks-to-skip and blocks-per cylinder values.

Default free-block list spacing assumed

This is an advisory message indicating the blocks-to-skip (gap size) is greater than the blocks-per-cylinder, the blocks-to-skip is less than 1, the blocks-per-cylinder is less than 1, or the blocks-per-cylinder is greater than 500. The default values of 7 blocks-to-skip and 400 blocks-per-cylinder are used. These values were set previously when the **mkfs** (make file system) command was used to make the file system.

Glossary

Glossary

Glossary G:

Glossary

Menu Terminology

This glossary defines terms and acronyms used in this document that may not be familiar to you. The glossary is divided into two parts: terminology for menus and terminology for hardware and software. This division will help you (the experienced user/programmer) skip the hardware and software terms that you already know.

Active Menu — The menu that you're currently working with.

CANCEL — A command that stops the suggested action and removes a form or menu from the screen, so that it's no longer available.

Cursor — The cursor is a movable pointer that designates where your input is echoed on the screen. If you issue a command that requires an argument, the item at the cursor is the item that action is taken on. The cursor occupies on character position.

Feature — Feature is a capability that is provided to you, e.g., user login administration.

Field — Field is an area in a form that you fill in with your choice or response. For example, you fill in the hour field with the correct hour when you set the system clock.

Form — A form is a group of items that appear on frames and collectively perform a task. These items require your input.

Frames — Frames are independently scrollable regions surrounded by a border. Frames occupy any part of the screen ranging from a few lines of the screen to the entire screen.

Function Keys — Function keys are the top row of keys on the keyboard, F1 - F8, that perform the commands displayed in the screen-labeled function keys.

Highlight

Highlighting is a method of selecting an item by moving the cursor to where it covers a task or command that you want to open.

Items

Items are the components of menus that a user selects, or components of forms that prompt for input.

Menu

A menu is a list of selectable items appearing in a frame that you can choose from by highlighting the item and pressing [Enter].

Movable Marker

The movable marker is an inverse video or a ">" that marks items that you selected with the MARK function key. The marker occupies multiple character positions.

Navigation

Navigation refers to the cursor movement between menus and within the same menu.

Pop-up menus

Pop-up menus are temporary frames that list selectable items.

SAVE

The SAVE screen-labeled function key is used to preserve information by recording it in a file on a disk.

Screen

Screen refers to your terminal screen. This screen may have any number of frames.

Screen-Labeled Function Keys

The SLKS are the highlighted areas at the bottom of your screen that display commands and act as labels for the keyboard function keys F1 - F8.

Scroll

Scrolling is an action that causes the contents displayed on the screen to move up or down.

Task

A capability that is a subset of a feature, e.g., adding a user login is a subset of the User Login Administration feature.

Text Frames

Text frames display text to the user but they do not display selectables or request user input via the keyboard.

Hardware And Software Terminology

Absolute Pathname The pathname that is used to specify a command or program from the **root** directory.

Application The software designed to perform a particular kind of work. For example, you use the Word Processing application to create and edit documents you want to print.

Backup A spare copy of data or software that you keep in case the original is damaged or lost.

Bad Track A part of the hard disk known as a track that is not usable.

Character A character is a letter, number, or symbol.

Command A command is an instruction used to tell the computer to perform a function or carry out an activity.

Configuration Configuration is the way that the computer is set up to allow for particular uses or situations.

Copy Copy means to duplicate information.

Daemon Daemon is a program that runs as a background process to handle UNIX system activities, e.g., file transfers, command executions, cleanup routines, etc.

Default Default is a value that the computer uses if you do not specify a value.

Delete Delete means to remove, erase, or discard data.

Encrypt Encrypt means to make a file unreadable by anyone who does not know the password to the file.

Error Message An error message is a response from a program indicating that a problem has arisen or something unexpected has happened, requiring your attention.

Floppy Disk Drive A Floppy Disk Drive is a device that reads and writes information on a floppy disk.

Format (1) Format is used to prepare a new floppy disk or hard disk for use with the computer.
(2) The way data is displayed. Pertains to the way the data appears on your screen or printed copy.

Hard Disk Hard disk is a device that stores operating systems, programs, and data files.

Install Install involves the procedures used to set up the hardware and software of a computer so that it can be used. Installing often includes customizing the system for a particular situation or user.

Meta Character A meta character is a set of characters the UNIX system shell interprets as having a special meaning.

Modem A modem is a device that modulates and demodulates data transmitted over communication lines.

Operating System The software that controls and allocates the resources, such as memory, disk storage, and the screen display for the computer.

Option A UNIX system option is an addition to a command to improve or provide an extra enhancement to the command. The option is usually depicted with a minus (-) sign in front of it.

Partition A partition is a section of the hard disk that is used to store an operating system and data files or programs. By dividing the disk into partitions, you can use the space allocated in a more efficient and organized manner.

Printer A printer is a machine that prints information transferred from a computer.

Printer Interface Program A program used to "set the printer up" for a print request. Each printer has an interface program.

Program
A program is a set of step-by-step instructions that tells a computer how to do a particular task.

Relative Pathname
The pathname that is used to specify a command or program from the current directory.

Software
Software is computer programs that have been stored on a disk or other media.

Spooling
The term "spool" is an acronym for **s**imultaneous **p**eripheral **o**perations **o**n-line. The line printer spooling system allows you to send a file to be printed while you continue with other work.

Syntax
Syntax is the format of a command line.

System
System is a general term for a computer and its software and data.

UNIX system
The UNIX system is a general-purpose, multiuser, interactive, time-sharing operating system, used with your computer.

Utilities
Utilities are a group of programs combined into a package that represent a specific application available with your computer.

Utility
Utility is a program, usually from a set of programs, that represents a specific application available with your computer.

Write-Protect Notch
The write-protect notch is a rectangular cutout on one edge of a floppy disk. If this notch is covered with a piece of special tape that comes with the floppy disk, new information cannot be written on the disk. Data on the floppy disk cannot be altered.

Index

Index

Index

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES

NOTES