HEWLETT
PACKARD

HP 9000
Workstations

Using HP-UX

Using HP-UX

HP 9000 Workstations

HEWLETT
PACKARD

HP Part No. B2910-90001
Printed in USA August 1992

Edition 1
E0892

Legal Notices

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty. A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.

Printing History

The manual printing date and part number indicate its current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The manual part number will change when extensive changes are made.

Manual updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updates or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

August 1992 . . . Edition 1. This edition incorporates material from *A Beginner's Guide to HP-UX*, Edition E0191, with the addition of updated material for HP-UX 9.0, HP VUE, the System Administration Manager, and Instant Ignition. This manual applies to HP 9000, Series 300, 400, and 700 computers.

Printing Conventions

This book uses the following typographical conventions:

If you see ...	It means ...
`computer text`	Text displayed by the computer system. For example, `login:` indicates a login prompt displayed by the system.
italic text	Text supplied by you. For example, `more` *file_name* means that you type `more` followed by a file name of your choice. Italic text is also used for text emphasis and for document titles.
(Key)	Type the corresponding key on the keyboard. For example, (CTRL)-(D) means you hold down the (CTRL) key, and press the (D) key.
`Displayed` `Item`	Select an on-screen item or a corresponding softkey. For example, `Help` shown at the bottom left side of the screen means that pressing the softkey corresponding to that position on the screen (`f1`) will cause a help screen to be displayed.

Contents

4. Working with Files

5. Organizing Files in Directories

6. Using the Shell Command Interpreter

7. Using Shell Processes

1

Introducing Your Workstation

Your new HP workstation uses the HP-UX operating system and the HP Visual User Environment (VUE). HP-UX is a versatile operating system that you can use to run application programs and perform a variety of tasks. HP VUE is a powerful graphical interface to HP-UX that will simplify many of your daily tasks.

Installing Your Workstation

If you have not installed your hardware or started your workstation, please refer to:

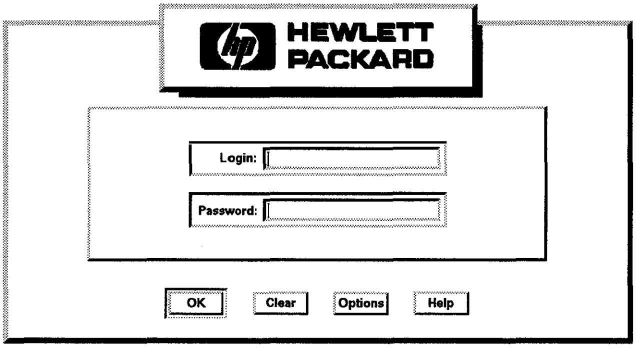
- The *Installation Guide* for your workstation.
- The *Owner's Guide* for your workstation.

Your Workstation Environment

Once your workstation is installed and running, it will display one of two ways to log in.

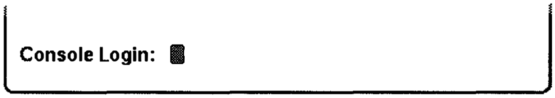
Workstations Running HP VUE

If your workstation is running HP VUE, you will see the HP VUE login screen:



Workstations Not Running HP VUE

If HP VUE is not running, you will see the system console login prompt:



```
Console Login: ▮
```

Choosing an Environment

We recommend that you use HP VUE on your workstation whenever possible. The powerful features of HP VUE make it easier to learn to use your workstation and extend its functionality.

Use HP VUE if you want the following:

- An easy-to-use interface. You'll still be able to type commands if you choose to do work that way.
- The ability to run more than one application at a time. You can run applications that create their own windows and applications that must be run in a terminal.
- The ability to run the applications that are part of HP VUE. For example, the HP VUE Text Editor provides an easy way to edit files. Its Help Manager lets you access extensive online information.

Use the HP-UX shell prompt without HP VUE if you need to do the following:

- Perform maintenance on your system.
- Run a single application that uses the entire display, such as Starbase Graphics or a CAD application.
- Use the X Window System without HP VUE.

If HP VUE Does Not Start Automatically

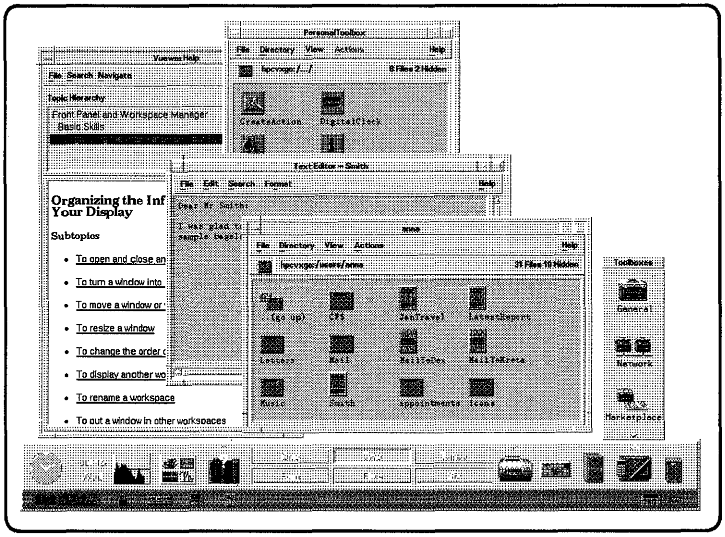
Under certain conditions, such as after a system update, or after a new workstation is added to an HP-UX cluster, HP VUE may not be configured to run automatically, and you will see a system console login prompt when you boot your system. If HP VUE is not running and you want to turn it on, see Chapter 12 for instructions.

Turning HP VUE off

If you need to totally disable HP VUE, see Chapter 12 for instructions.

Introducing HP VUE

HP VUE is a powerful graphical environment and a set of applications for interacting with your computer.



Features of HP VUE

HP VUE includes these features:

- Windows and workspaces. Windows are containers on the screen for applications; they let you run more than one application at a time. Workspaces are a way of providing more room on your display for windows.
- Icon-based file management. Files are represented by icons that can be selected and moved on the display.
- Front Panel and toolboxes for easy access to applications.
- Extensive online help.
- Session management. HP VUE remembers which applications were running when you logged out and restarts them the next time you log in.

- Easy customization for colors, fonts, window behavior, and other aspects of the appearance and behavior of your workstation.
- Easy-to-use Text Editor and Icon Editor.
- Multi-media applications for annotating files.

Two Forms of HP VUE

Regular HP VUE sessions provide all the HP VUE features.

HP VUE Lite is a subset of HP VUE. It features enhanced system performance by omitting full icon-based file management, full session management, and file annotation. Some models and configurations of HP workstations may not give satisfactory performance when running HP VUE. If you wish to exchange some HP VUE features for enhanced performance, you can use HP VUE Lite by selecting it from the `Options` menu on the HP VUE login screen.

Entering Commands

While HP VUE provides an easy way to work with files and applications using icons, there are times you may want or need to enter a command by typing it into a command line. Command lines are provided by HP VUE's terminal window application. See Chapter 2 to learn how to use a terminal window. For information about commands, see Chapter 6 and Appendix A, in this manual.

Additional Login Options

You can use the `Options` menu on the HP VUE login screen to suspend HP VUE in order to run a special program or to perform certain configuration tasks. The `Options` menu also accesses sessions in other languages.

Learning About HP VUE

This manual covers the basics of how to use HP-UX and HP VUE. See *HP VUE User's Guide* for the details of using HP VUE.

You can also use the HP Help System to access online help about HP VUE. To learn more about HP Help, see Chapter 3 in this manual.

Introducing the Console Environment

The console shell prompt and terminal windows provides a command-line interface for running special applications and performing certain system administration functions, working from a shell prompt. The example below shows sample output from the **ls** command, entered at a shell prompt ($):

```
$ ls

bin
news
projects
other
```

Learning About System Commands

Chapter 6, and Appendix A, in this manual, explain basic HP-UX commands that you can use with a shell prompt (command line entry).

If you use shell prompts to run the X11 Windows System, you can also use HP Help to access extensive online information about HP-UX.

Summary of HP VUE and Console Features

Feature	HP VUE	HP VUE Lite	Console
Windows	yes	yes	yes, with X Window System or TSM
Workspaces	yes	yes	no
File management using icons	yes	no	no
Front Panel containing controls for common tasks	yes	yes	no
Toolboxes containing applications	yes	no	no
Text editor	HP VUE Text Editor, `vi`		`vi`
Applications for customizing your workstation	yes	yes	no
Command line	yes	yes	yes
Mailer	yes	yes	yes
Online help	yes	yes	yes, with X Window System
Memory usage	high	medium	low

Finding Information About Your Workstation

For a quick reference to commonly-used HP-UX commands, see Appendix A. For information on using the X Window System, see "Logging In and Out of the X Window System" in Chapter 2, in this manual.

Manuals

To learn more about using your system, continue reading this manual, the *HP VUE User's Guide*, and the *HP VUE Quick Start Guide*. These other manuals may also be useful:

- If you need help with system hardware installation, see the *Installation Guide* and *Owner's Guide* for your system.
- If you have not yet installed your HP-UX system, see *Installing and Updating HP-UX* for your version of HP-UX.
- For administration information, see the *System Administration Tasks Manual*.
- For troubleshooting HP-UX, see *Solving HP-UX Problems*.

Online Help

If you are using HP VUE or the X Window System, you can use HP Help to access information about HP-UX and HP VUE. To use HP Help, refer to Chapter 3, in this manual.

- For information about HP VUE, refer to HP VUE Help, in the HP Help Manager.
- For information about HP-UX, refer to HP-UX Operating System Help, in the HP Help Manager.
- For information about HP-UX documentation, see "Finding HP-UX Information" in HP-UX Operating System help.

2

Logging In and Doing Basic Tasks

Chapter Contents

- Before Logging In the First Time.
- Logging In.
- Performing Key Tasks
- Using Your System Everyday.
- Introducing the HP VUE Front Panel.
- Basics of Using Windows.
- Basics of Using Controls.
- Basics of Using Menus.
- Keyboard Equivalents.
- Where to Go From Here.

Before Logging In the First Time

The *Installation Guide* for your system will give you the specific procedures for installing and booting your system the first time. The *Owner's Guide* for your system gives you further details about your system, including mass storage installation and initial logging in.

This chapter reviews some of the initial procedures and provides initial information on using both HP VUE sessions and HP-UX. For more detailed information about HP VUE procedures, see the *HP VUE User's Guide*.

If you have not completed the installation process, you will need to be prepared with the following information:

- The time zone where your computer is located.
- The System name (host name) for your workstation; any alphanumeric, single-word name with eight or fewer characters.
- The network address number, also called an IP number, for your workstation. This consists of four address fields separated by periods: for example, *255.32.3.10*. You may need to consult with your system administrator for this information. (If your IP number has already been assigned, you can determine it, after boot, by entering `nslookup` *nodename*, at the system prompt).

If your system has HP-UX preloaded on its disk (this is indicated by a label over the power switch), HP-UX will automatically load itself when you first turn the power on. You will still need to furnish the above information. See your *Owner's Guide* for further details.

If your login screen is now displayed, go on to the next section.

Logging In and Out

Once HP-UX is running on your system, you must log in. The process of logging in is one of the ways that HP-UX prevents unauthorized persons from using your system. This is especially important if your system is attached to a network.

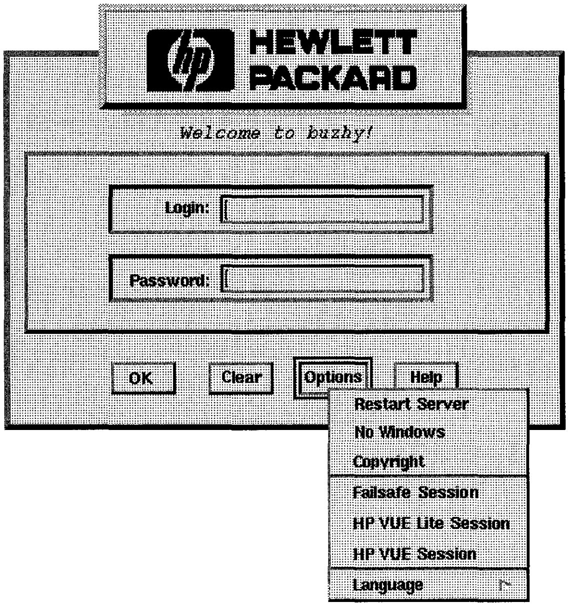
Logging In the First Time

- If you have a pre-loaded operating system, you will see a `Welcome` window displayed along with the HP VUE Front Panel, the first time you log in. The information in this screen will help you explore the capabilities of your system and perform some basic tasks.
- If you are only updating your HP-UX system to the current version and you have installed HP VUE, then you will see the `Welcome to Help Manager` and the HP VUE Front Panel.
- If you are only updating your HP-UX system to the current version and you have *not* installed HP VUE, then you will work with the shell prompt ("`$`").
- When you log in to HP VUE for subsequent sessions, you will see the Front Panel and the File Manager for your home directory.
- When you log in using "`login:`" (or "No Windows") for subsequent sessions, you will see the the shell prompt ("`$`").

Preparing to Log In to an HP VUE Session

The login screen, created by the HP VUE Login Manager provides a place for you to type your login name and password.

The `Options` menu on the login screen allows you to select several alternative types of sessions, such as HP VUE Lite, or a failsafe session. You can also select the language for your session.



The Options menu lists your login options.

- During the login process, if you need help logging in, click the login screen Help button.
- If you choose *not* to use HP VUE after installation, (and you know how to use the command line), you can select Options from the login window menu and No Windows from the VUE login screen, at this time. In that case, enter your login name and password after the appropriate prompts and see Chapter 6 for further information on using the command line.
- If you are *not* using HP VUE, you will log in at the login prompt ("`login:`").

Logging In to an HP VUE Session

1. Select the Login box and type your login name. Press (Return) or choose OK.
2. Type your password. Press (Return) or choose OK.

If the Login Manager does not recognize your name or password, you will see an error message. If this happens, choose Clear and start over.

If your previous session was an HP VUE Lite session, then you must select HP VUE Session from the login screen Options menu before logging in.

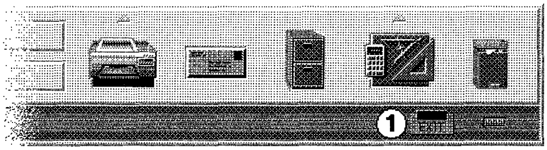
Once you've logged in, the Session Manager starts a session:

- If this is the first time you've logged in, you'll get a new session.
- If you've logged in before, your previous session will be restored.
- If this is the first time you've logged into HP VUE 3.0, but you previously used HP VUE 2.01 on this system, your previous HP VUE 2.01 session will be restored.

Note Your system may be configured to use only HP VUE Lite sessions. If this is the case, you will automatically log into HP VUE Lite.

Logging Out in HP VUE

- Choose the logout control on the Front Panel.



Use the log out control ① to end the session.

- *Or,* choose Log out from the workspace menu.

When you log out of a regular HP VUE session, the Session Manager saves information about your current session so that it can be restored the next time you log in.

Logging Out and Cancelling a Session

Cancelling a session ends the session without saving any new session formatting information. It also provides a way to log out if, for some reason, the Front Panel and Workspace Menu are not working properly.

If you are using a VUE or X Window terminal window, you can use the "three key" method of logging out without saving any new window configuration:

1. Save any open text files and exit from open processes.
2. Press (Shift)-(Ctrl)-(Reset).
   For PC-101 keyboards, use ((Shift)-(Ctrl)-(Pause).

Note If you are working in an HP VUE terminal window, and you want to save a new Workspace Manager configuration, exit using the Exit button at the right of the Front Panel.

For information on running the various types of HP VUE sessions, see the *HP VUE User's Guide*.

Logging In and Out of the X Window System

Your HP-UX contains the filesets to run either the X Window System or HP VUE (Visual User Environment).

If you wish to run just the X Window System, without using the HP VUE workspaces, you can select it when you create your new user account.

You can begin an X Window (Motif) session by typing the following from a shell prompt:

```
x11start
```

This will cause the X Window System to run, with system default settings, for the current login session.

To log out of an X Window System session, press Shift-Ctrl-Reset. (For PC-101 keyboards, use (Shift-Ctrl-Pause)).

For information on using the X Window System, see *Using the X Window System*.

Logging In and Out of HP-UX

When you log in without windows, a command-line login prompt appears:

```
login:
```

1. Type your login name (or `root`).
2. Press Return.

   If you *haven't* yet set a password, you will get a a system prompt (**#** for root, or **$** for user), and you can begin using the system.
3. If you *have* set a password, type it when the system gives the following prompt:

   ```
   Password:
   ```
4. Press Return. The system prompt (**#** or **$**) appears and you can use the system.

Logging Out

With command lines, you can also either use a screen-locking feature to secure your workstation while you temporarily leave it (and leave processes running), or you can log out of your current work session entirely.

Use the `exit` command to log out from the command line:

```
exit
```

To lock the screen from a shell prompt using `lock`, see "Locking and Unlocking the Display with the Shell Prompt".

Caution

If your system has its own disk and you are running a local operating system, do not turn off power to your system without first shutting down the operating system software according to the procedure in “Shutting Down Your System” in Chapter 12. Turning off the power for your stand-alone computer without first doing the shutdown procedure may result in damage to data on your disk. Always execute the shut-down process to completion first.

If you are running your system as a node in a cluster (with the operating system running on a separate server computer) you can, in any case, shut down your computer by turning off the power after you have properly closed files and terminated processes.

Introducing the HP VUE Front Panel

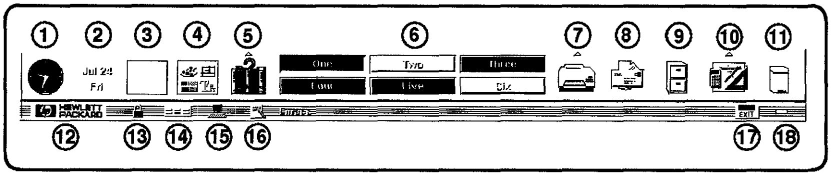
Now that you have been introduced to the techniques of logging in and out and locking your system, you will need some additional details about using the system interfaces to do tasks.

HP VUE provides several options in the level of features provided:

- HP VUE Regular is the fully-implemented VUE interface.
- HP VUE Lite has fewer actions which are immediately accessible.

The following describe the Front Panel configurations for each.

Front Panel Controls (Regular Session)



Top Row Controls (left to right)

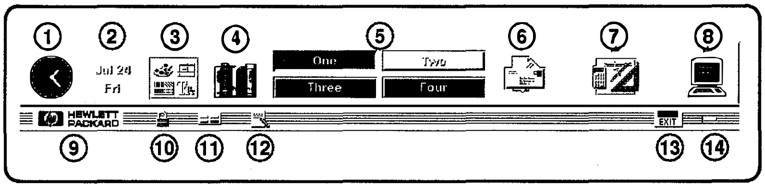
1. **Clock** displays the current workstation time.
2. **Date** displays the current workstation date.
3. **Load** displays workstation activity. This control is actually an application displaying a window in the Front Panel.
4. **Style Manager** starts Style Manager with which you change display appearance, such as colors, and change system device behavior, such as mouse double-click speed.
5. **Help Manager** starts Help Manager. The Help subpanel (accessed by clicking on the up arrow) provides additional online information.
6. **Workspace Switch** displays another workspace.

7. **Printer** displays printer job status on the system default printer. The button is also a drop zone that accepts a file icon. The Printer subpanel can be configured for other printers.
8. **Mailer** starts your electronic mail application. The button is also a drop zone and accepts a file icon.
9. **File Manager** starts a File Manager window showing your home directory.
10. **Toolbox** opens your Personal Toolbox. The Toolbox subpanel opens other Toolboxes that contain actions and utilities.
11. **Trash Can** displays the contents of the trash can. The button is also a drop zone that accepts a file icon.

Bottom Row Controls (left to right)

12. **Logo** gives HP VUE version information.
13. **Lock** locks your workstation, preventing unauthorized input.
14. **Rename Workspace** displays a dialog in which you can rename a workspace.
15. **Terminal** starts a terminal window, providing access to a command-line prompt.
16. **Text Editor** starts Text Editor. The button is also a Drop zone that accepts a file icon.
17. **Logout** Begins the logout process.
18. **Progress Light** blinks to indicate an activity in progress, such as a new window opening.

Front Panel Controls (HP VUE Lite Session)



Top Row Controls (left to right)

1. **Clock** displays the current workstation time.
2. **Date** displays the current workstation date.
3. **Style Manager** starts Style Manager with which you change display appearance, such as colors, and change window behavior.
4. **Help Manager** starts Help Manager. The Help subpanel (accessed by clicking on up arrow) provides access to additional online information.
5. **Workspace Switch** displays another workspace.
6. **Mailer** starts your electronic mail application. The button is also a drop zone and accepts a file icon.
7. **Toolbox** can be configured to run an application of your choice.
8. **Terminal** starts a terminal window, providing access to a command-line prompt.

   Bottom Row Controls (left to right)

9. **Logo** gives HP VUE version information.
10. **Lock** locks your workstation, preventing unauthorized input.
11. **Rename Workspace** displays a dialog in which you can rename a workspace.
12. **Text Editor** starts Text Editor. The button is also a Drop zone that accepts a file icon.
13. **Logout** Begins the logout process.

14. **Progress Light** blinks to indicate an activity in progress, such as a new window opening.

For More Information

To find out more details for manipulating HP VUE Front Panel controls, see the *HP VUE User's Guide.*

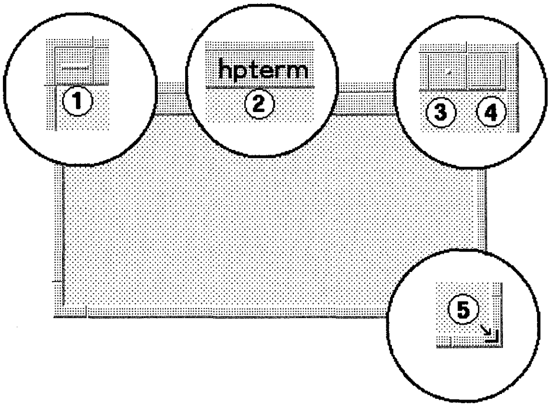
Basics of Using Windows

If you are not running HP VUE, but you are running the X Window System, the following contains basic information on using windows and using HP VUE. If you are new to using windows, you will need to see this section. If you are already familiar with using windows, go on to "Performing Key Tasks", in this chapter. "Keyboard Equivalents" gives information on using keyboard equivalents to mouse actions.

HP VUE provides a graphical user interface for HP-UX, windows, and workspaces. The X Window System provides standard Motif windows and utilities for maintaining them. If you are using either HP VUE or the X Window System, then you will be using the same basic Motif window manipulation and management techniques.

See "Logging In and Out of the X Window System", in this chapter, for information on logging in with the X Window System.

The following figure shows the controls for a window.



Window Controls

① Window menu button. Used to move, size, lower, or remove the window from the workspace.

② Title bar. Used with mouse to move the window to a convenient location on the screen.

③ Minimize button. Turn the window into an icon.

④ Maximize button. Makes window approximately the size of your screen.

⑤ Resize border. Used with mouse to resize window.

By default, clicking your mouse while the pointer is in a window will make the window active. When a window becomes "active," its frame changes color, and you can use the keyboard to execute commands in the window. See "Keyboard Equivalents" for keyboard equivalents to mouse actions.

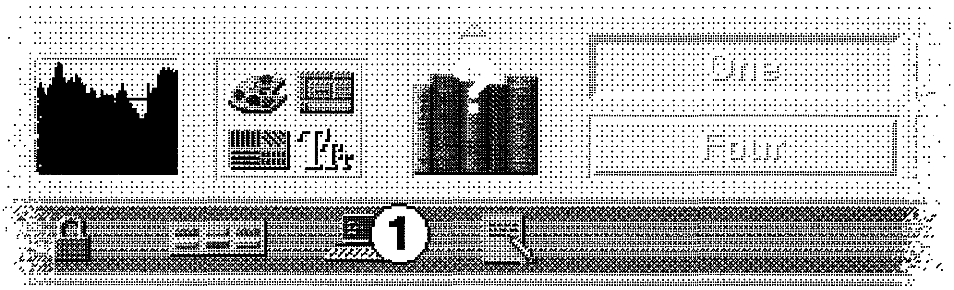
Some application may use only some, or none, of the window frame controls.

Opening and Closing a Terminal Window

For using a shell prompt to enter HP-UX commands, you will generally be using a terminal window. There are two types of terminal windows, HP Term and X Term. Each has slightly different behavior. For example, HP Term windows (Term0 compatible) can utilize softkey representations, whereas X Term emulates a VT102 (ANSI compliant) terminal. For more information, see the *HP-UX Reference* entries *hpterm*(1) and *xterm*(1).

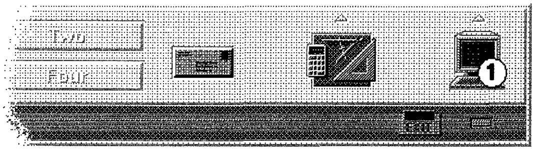
To open a terminal window from a shell prompt, type **hpterm**.

To open a terminal window in HP VUE, click on the Front Panel button that resembles a keyboard and screen. For a regular HP VUE session the button is found in the following location:



The HP VUE Terminal Button Location

For an HP VUE Lite session, you'll find it in the following location:



The HP VUE Lite Terminal Button Location

Opening and Closing an Application Window

Many applications have action buttons on the Front Panel which will create windows specific to those actions. To open an application window (for example, Mailer or File Manager):

- Choose the application button in the Front Panel.

Starting an application opens a window containing the application.

Example

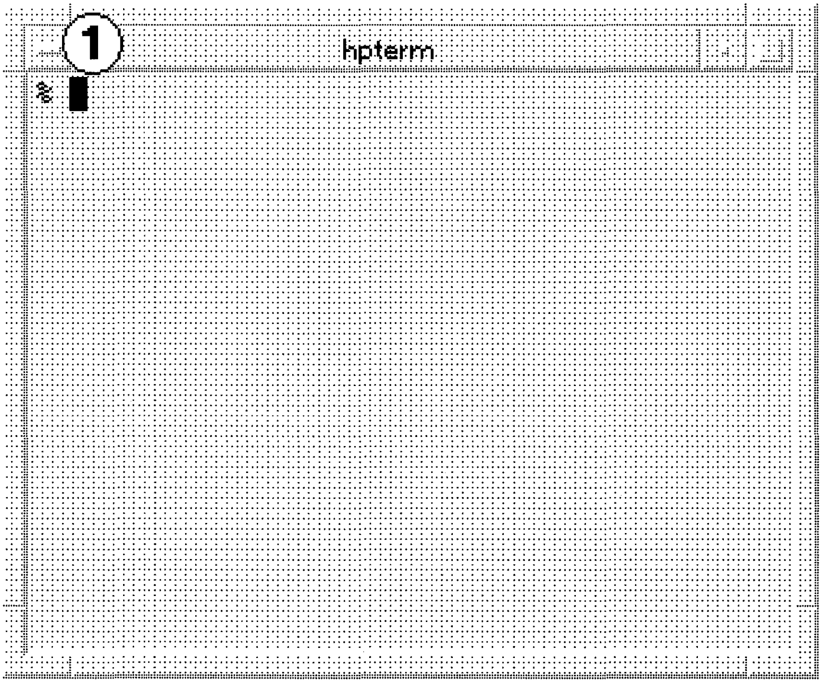
To start File Manager:

- Move the pointer to the Front Panel and choose the File Manager ("file cabinet") icon.

Closing an Application

You can close a window by doing any of the following:

- In general, the easiest way to close a window is by double-clicking the window menu button ▬, at the upper left of the window.



```
$ 
```

The Window Menu Button

- Or open the window menu by clicking on the menu button (−). Then choose `Close`.
- Inside an application, you can choose the application's `Exit` or `Close` command (usually in its File menu).
- With the keyboard alone, you can press Alt+F4.

Closing an application removes it from all workspaces.

Caution Before closing an application, always save your work.

Converting a Window Into an Icon

As you work, your screen can become cluttered with windows. Changing a few of those windows into **window icons** tidies up the workspace. Programs running in window icons continue to run.

To make a window into an icon:

- Click the window's Minimize button (the dot at the right side of the top border).

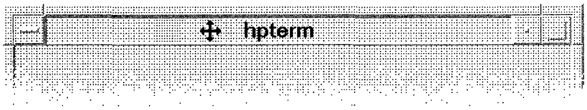
To restore a window to normal size: To turn the window icon back into a window:

- Double-click the icon
- *Or*, choose Restore from the window icon menu (mouse button 3).

Moving a Window or Window Icon

1. Position the mouse pointer:
   - For a window, move the pointer over its title bar.

   - For a window icon, move the pointer over the icon.
2. Hold down mouse button 1 as you drag it to its new location.



Move a window by dragging it by its title bar.

To cancel the move operation, press Esc.

Resizing a Window

1. Place the pointer on a window frame's side or corner.
2. Hold down mouse button 1 as you drag the window outline to the new size.

To cancel the resize operation, press (Esc).

Changing the Order of Overlapping Windows

Windows in a workspace can overlap, just like pieces of paper on a desk. And, like papers, you can change the stacking order, bringing the one needing your attention to the top.

- To bring a window to the top, click on a visible part of the window's frame.
- To bring a concealed window to the top, lower the visible windows by selecting Lower from the Window Menu.

Displaying Another Workspace

- Click the workspace button in the Front Panel.

Basics of Using Controls

To help you manipulate the functions of an application, windows contain standard sets of **controls**, including the following:

- **Buttons** execute commands, start actions, or specify options and settings. Types of buttons include push buttons, toggle buttons, and radio buttons.
- **Text fields** provide an area where you can type information.
- **Lists** display a list of choices from which you can select.
- **Sliders** provide incremental selection from a range of values.

Choosing a Push Button

Choosing a push button immediately performs the associated command or action. For example, choosing an OK button applies any changes made to a dialog, then closes it.

- Move the pointer over the button and click mouse button 1.

Selecting a Toggle or a Radio Button

Generally, selecting a toggle button merely specifies the option or setting to be used when a command, such as OK, is performed. You can select more than one toggle button in a group.

Selecting a radio button also specifies an option or setting, but only one radio button in a group can be selected at a time.

- Move the pointer over the button and click mouse button 1.

To select the button, move the pointer over a selected button and click mouse button 1. The outline of the button changes to indicate that it has been “pressed”. To unselect it, choose another button.

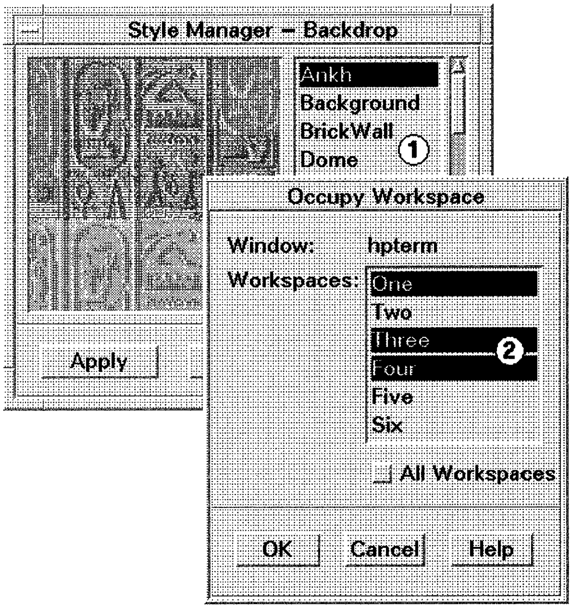
Choosing a List Item

A selected list item highlights and is acted upon when a command, such as OK, is chosen.

1. Click the list item.
2. Choose a command, such as OK.

In some lists, double-clicking an item selects the item and chooses the command indicated.

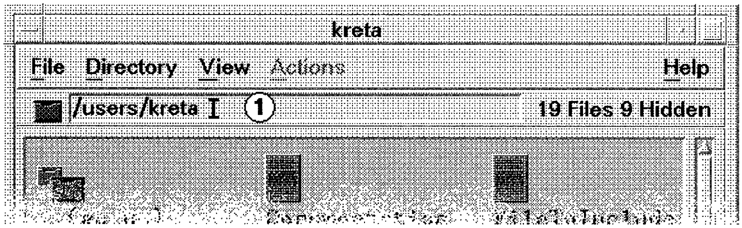
In case of multiple-selectable lists, click on OK after all the selections are made.



① Single-selection list.
② Multiple-selection list.

Entering Text Into an Empty Field

- Click the field to get the text insertion cursor.
- Start typing.



① **A field that shows a blinking text insertion cursor is ready for you to enter text.**

Editing Text in a Field

1. Select the text in the field to edit:
   - **Character(s):** Move the pointer to the first character and drag the pointer to the last character.
   - **Word**: Double-click the word.
   - **Line**: Triple-click the line.
   - **Multi-line field**: Quadruple-click the field.
2. Type the replacement text.

Using the Editing (White) Keys

Arrow	Move the text insertion cursor.
Backspace	Delete the character before the cursor.
Delete char or Del	Delete the character following the cursor.
CTRL+Delete char or CTRL+Del	Delete all the characters from the cursor to the end of the current line.

Cutting and Pasting Text

Cutting and pasting eliminates the need to retype text. You can cut from and paste into:

- Text fields.
- Terminal windows.
- Editor windows.

Cutting Text

1. Move the pointer to the start of the text block.
2. Hold down mouse button 1 and drag the pointer to the end of the text block.

Pasting Text

1. Put the text insertion cursor at the target location.
2. Click mouse button 2 to paste the text into the field.

To unselect text, click mouse button 1 in an empty area of the window that has the text selected, or press (ESC).

Using Sliders

- Drag the slider by moving the mouse while holding down mouse button 1.

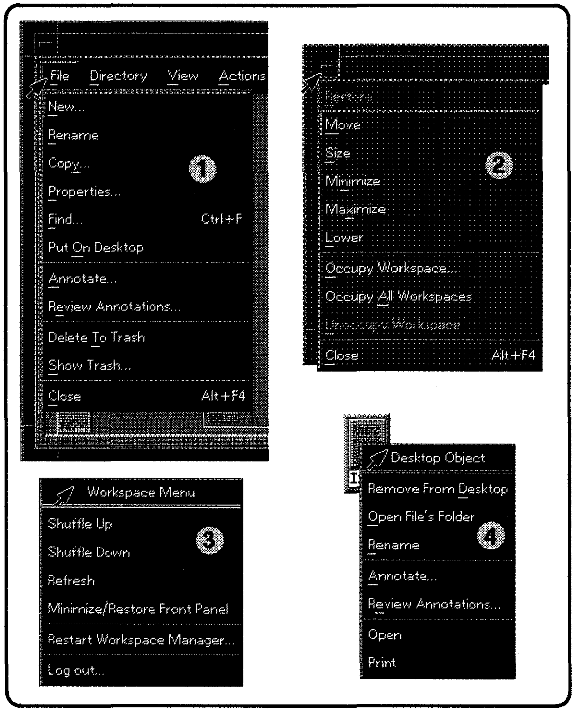
Basics of Using Menus

A menu is a list of frequently-used commands that help you manage windows and operate software applications.

Choosing a menu command performs an associated action. For example, choosing `Minimize` from a window menu turns the window into an icon.

Some of the most common menus are:

- Window menus.
- The HP VUE Workspace menu.
- Software application menus:
  - □ Menus that "pull down" from the application's menu bar.
  - □ Menus that "pop up" on an application or on a Desktop object.



Sample Menus

① An application's menu pulled down from a menu bar.
② A window menu.
③ The HP VUE Workspace menu.
④ A Desktop object menu popped-up from the object (mouse button 3).

Menu commands that have darkened labels are inactive. These commands are only available under certain conditions. For example, the Restore command is available in a window menu only when the window is an icon.

Choosing a Command from a Window Menu

1. Click mouse button 1 on the window menu button to display the menu. *Or*, if the window is an icon, click the icon.
2. Click a menu command.

Or, you can use the dragging method:

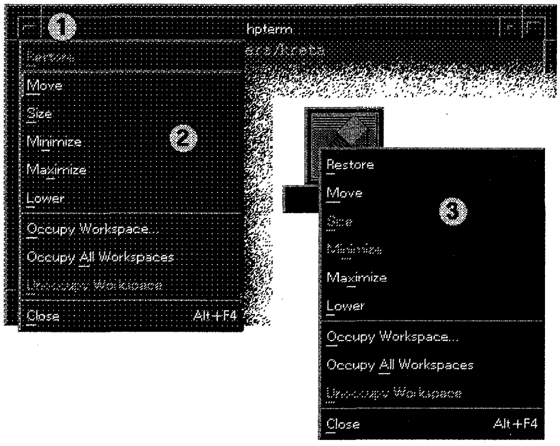
1. Hold down mouse button 1 as you drag the pointer through the menu.
2. When you reach the command, release the mouse button.

To close the menu without choosing a command, click anywhere off the menu.

To close the menu without choosing a command, press (ESC).

The Window Menu

The window menu provides easy access to services that control the behavior of the window.



① Window menu button.
② Window menu.
③ Window icon menu.

Restore
Turns an icon back into a window. This item is active only when the window is an icon.

Size
Interactively changes the size of the window. This item is inactive in the icon's menu.

Minimize
Turns the window into an icon.

Maximize
Enlarges the window to its greatest allowable size.

Lower
Puts the window on the bottom of the stack of overlapping windows in a workspace.

Occupy Workspace
Displays a dialog to select the workspaces in which the window should appear.

Occupy All Workspaces
Displays the window in all workspaces.

Unoccupy Workspace
Removes the window from the current workspace. This item is inactive if the window is displayed in only one workspace.

Close	Closes the window, removing it from the workspace

Choosing a Command from the Workspace Menu

1. Move the pointer over the workspace backdrop.
2. Hold down mouse button 3 as you drag the pointer through the menu.
3. When you reach your choice, release the mouse button.

The Workspace menu can only be accessed with the mouse.

The Workspace menu contains commands that help manage the workspace.

Shuffle Up	Puts the bottom window (in a stack of overlapping windows) on the top of the stack.
Shuffle Down	Puts the top window (in a stack of overlapping windows) on the bottom of the stack.
Refresh	"Repaints" the screen should the video display become unreadable.
Minimize/Restore Front Panel	Turns the Front Panel into an icon. When selected a second time, restores the Front Panel.
Restart Workspace Manager	Stops then restarts the Workspace Manager after you have customized configuration files.
Log out	Begins the logout process, the same as pressing the exit button in the Front Panel.

Choosing a Command from Other Menus

Clicking Method

1. Display the menu:
   - For a menu bar menu, click the menu name using mouse button 1.
   - For a pop-up menu, click inside the application using mouse button 3.
   - For an object's pop-up menu, click the object with mouse button 3.
2. Click on the command.

To close the menu without choosing a command, move the pointer off the menu, and click the mouse button a second time.

Dragging Method

1. Hold down the mouse button as you drag the pointer through the menu:
   - For a menu bar menu, use mouse button 1.
   - For a pop-up menu, use mouse button 3.
2. Release the mouse button when you reach the command.

To close the menu without choosing a command, move the pointer off the menu and release the mouse button.

Performing Key Tasks

After you know how to log in and out and the basics of window management, there are several other key tasks you will need to know about to prepare for everyday operation:

- Creating a new user account
- Setting or changing a password.
- Locking or unlocking your display.

Note The Welcome Window in the Help Manager (? icon with books) provides assistance with these tasks.

If you already know how to do these tasks, go on to "Where to Go From Here", in this manual.

Creating a New User Account in HP VUE

If you are using a system administrator, that person may have already set up a user account for you. If you are doing your own system administration, you will need to do the following to set up a user account.

The home directory or "account" in which you work is where you log in at every new session and where most of your default files are found. As "owner" of this directory and its subdirectory, you also have control over who can access any of its files.

If you have not already created a user account, do the following steps to create one:

1. Log into HP VUE as `root`. (You will need the `root` password).
2. Click on the up arrow (the small triangle) over the Toolbox control icon at the right of the Front Panel. A slide-up menu ("subpanel") appears.
3. Choose the `General` icon on the Tools subpanel. You will see a window which lists applications in the General Toolbox. Select the `System_Admin` folder.
4. The System Administration window appears. Select `Sam`.

Caution If you are not already logged in as `root` you will be asked for your root password to use SAM. `root` is a separate login account providing unlimited permissions on your system. This means that you need to take actions more carefully when in the `root` account. `root` is only used to do system administration tasks, and, for security reasons, it should use a password which is different from your everyday user password. Chapter 3 and Chapter 12 give you more detail on using SAM.

5. At the opening menu, choose `Users and Groups->` by clicking on it and then clicking on `Open` (or just double-click on the item).
6. At the next screen, choose `Users`. You will see a screen displaying a list of logins and real names.
7. Select `Add` from the `Actions` pull-down menu. You will see a form `Add a User Account`.
8. Fill in your login name, choice of start-up program and environment (if different from the defaults given), and the optional information.

Note At this point you can select X Windows as your login default environment, if you so desire.

9. Choose `OK` when you are finished.
10. You will be asked to select a password. (See “Selecting a New Password” for password requirements. If you wish, you can select a temporary password and reset it later). Type the password and click on `OK`. Re-enter the password, as requested. The re-entered password must match the first.
11. Choose `OK`
12. When the “Task Completed” message appears, choose `OK`.
13. In the Users window, select `Exit` from the `List` menu (or double-click on `-`).

14. Choose the `Exit SAM` button.

Creating a User Account with SAM

If you are not running HP VUE, you can set up a user account by accessing SAM directly and use it in a similar way, in graphical or "character terminal" display mode. Note that, if you are not already logged in as `root` you will be asked for your root password to use SAM.

1. Type `sam`, as `root`.
2. At the opening menu, choose `Users and Groups->` by clicking on it and clicking on `Open` (or press (Return), if you are using a keyboard).
3. At the next screen, select and open `Users`. You will see a screen displaying a list of logins and real names.
4. Select `Add` from the `Actions` pull-down menu. You will see a form `Add a User Account`.
5. Fill in your login name, choice of start-up program and environment (if different from the defaults given), and the optional information.

Note At this point you can select X Windows as your login default environment, if you so desire.

6. Choose `OK` when you are finished.
7. You will be asked to select a password. (See "Selecting a New Password" for password requirements. If you wish, you can select a temporary password and reset it later). Type the password and choose `OK` (or press (Return)). Re-enter the password, as requested, and choose `OK`. The re-entered password must match the first.
8. Choose `OK`.
9. When the "Task Completed" message appears, choose `OK`.

10. In the Users window, select `Exit` from the `List` menu (or double-click on `-`).

11. Choose the `Exit SAM` button.

Setting or Changing a Password with HP VUE

To change or create a new password for your new user account or for `root`, you can also use the HP VUE Toolbox:

- Choose the Toolbox control icon. You will see the Personal Toolbox. (The `ChangePassword` action can be accessed either here or in the General Toolbox, which appears on the Toolbox subpanel).
- Choose `ChangePassword`. You will see a `Changing password` screen.
- Enter your old password, at the prompt.
- Enter the new password. (See "Selecting a New Password", if necessary).
- Re-enter the new password, as requested.

Setting or Changing a Password with HP-UX

From a command line shell prompt, you can use the `passwd` command directly to set or change a password. Enter the following:

```
passwd
```

You will be prompted for your old password. Then you will be prompted to enter and re-enter your new password. The re-entered password must match the first entry.

See the later section in this chapter, "Selecting a New Password", if you need help with selecting passwords. Use the same procedure to change an old password as to add a new password. If you already have one, you will be prompted appropriately for the old password.

Selecting a New Password

If you have already booted and used your system, you should already have set different passwords for your user account and for `root`.

However, you will also want to change your password from time to time as a matter of good security practice. The following gives the general requirements of setting passwords. See Chapter 14, in this manual, for more details about password selection.

A password must meet four criteria to be valid:

- Contain at least six characters.
- At least two characters must be alphabetic.
- At least one character must be a number (0-9) or a special character (/, ?, !, or other punctuation mark).
- Differ from your previous password by at least three characters.

Your password is case-sensitive, so the password `?Secret` is different from the password `?secret`. Your password can also be as long as you want, but only the first eight characters are checked.

If you are adding many users to your system, see *System Administration Tasks* for the details of controlling access to your system.

Locking and Unlocking Your Display

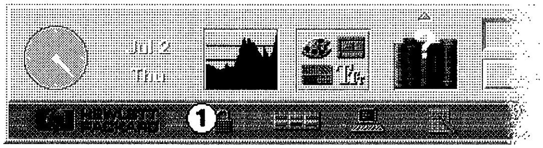
The display lock lets you prevent other people from using your workstation when you are away from it without logging out. You can set the appearance of the locked display.

The system timeout automatically turns off the display after a certain amount of time elapses in which you haven't used the keyboard or mouse. You can set:

- The amount of time until timeout.
- Whether or not to lock the display at timeout.

Locking and Unlocking the Display in HP VUE

1. Choose the lock control in the Front Panel. The screen will change to a lock display.
2. To unlock the screen, enter your password (it will not show on the screen.)



Choose the lock control ① to lock your display.

Locking and Unlocking the Display with the Shell Prompt

If you want to leave your workstation for a short time without logging out, you can use the `lock` command. You will be asked to enter a password and repeat it. Enter the following:

```
lock
```

At the prompts, enter and repeat the password. Re-enter the password to unlock.

Keyboard Equivalents

All mouse inputs can be replicated by keyboard equivalents. See *HP VUE User's Guide* for detailed keyboard commands to run HP-VUE. In general, you can press Extend\char-Space Bar (or Alt-Space Bar) to display the window menu. Pressing the underlined letter of a command on the menu will produce the indicated action. Then you can press the down arrow until the command is illuminated; then press Enter to activate it.

If you use the keyboard, note the following:

- Alt is the same as Extend char on some keyboards.
- Enter is the same as Return on some keyboards.

Outside a Window

Key(s)	Moves to ...
Alt+Tab	Next window or window icon.
Shift+Alt+Tab	Previous window or window icon.
Alt+F6	Next window belonging to an application or between the Front Panel and a subpanel.
Shift+Alt+F6	Previous window belonging to an application or between the Front Panel and a subpanel.

Within a Window

Tab	Next tab group.
Shift+Tab	Previous tab group.
▼	Next control in a tab group.
▲	Previous control in a tab group.

Within a Menu

▼	Next menu command.
▲	Previous menu command.

Where to Go From Here

This chapter has provided you with some preparation for using HP VUE windows and shell prompts in order to perform basic tasks. The next chapter will help you in using the online help information provided with your system.

- If you have HP VUE and need to learn more about using it, see the VUE help described in the next chapter. Also see *HP VUE User's Guide.*
- If you do not have HP VUE or have decided to remove it and work with the command line, this manual will help you with all the basic aspects of using HP-UX. For information on obtaining help, see the HP-UX help and information sources described in the next chapter.
- For advanced information, refer to *System Administration Tasks*, *HP-UX Reference*, *User's Guides*, or to the online "Man Pages". The *Ultimate Guide to the vi and ex Text Editors* has advanced information on using these text editing tools.

3

Getting Help

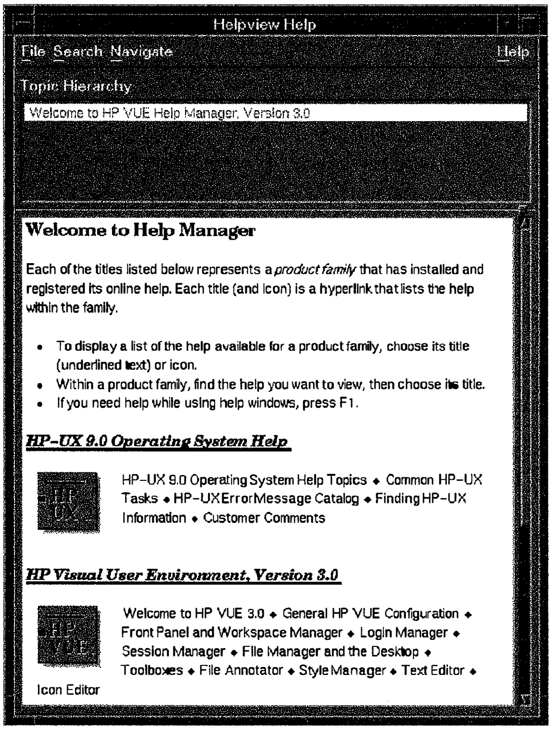
Chapter Contents

Getting Online Help

If you are using HP VUE or the X Window System, you can use the HP Help System. HP Help provides a wealth of help information that you can browse, search, display, and print. This information displays in a window called the Help Manager.

- *To start Help Manager in HP VUE:* Choose the Help control (the books and question mark) in the Front Panel. (Be careful not to choose the up arrow at the top of the control.)
- *To start Help Manager from the command line or in the X Window System:* Type `helpview &` at a command line prompt (this runs `helpview` as a background process so that you can do other work at the same time).

You should see a window similar to this:



Welcome to HP VUE Help Manager, Version 3.0

Welcome to Help Manager

Each of the titles listed below represents a *product family* that has installed and registered its online help. Each title (and icon) is a hyperlink that lists the help within the family.

- To display a list of the help available for a product family, choose its title (underlined text) or icon.
- Within a product family, find the help you want to view, then choose its title.
- If you need help while using help windows, press F1.

HP-UX 9.0 Operating System Help

HP-UX 9.0 Operating System Help Topics ◆ Common HP-UX Tasks ◆ HP-UX Error Message Catalog ◆ Finding HP-UX Information ◆ Customer Comments

HP Visual User Environment, Version 3.0

Welcome to HP VUE 3.0 ◆ General HP VUE Configuration ◆ Front Panel and Workspace Manager ◆ Login Manager ◆ Session Manager ◆ File Manager and the Desktop ◆ Toolboxes ◆ File Annotator ◆ Style Manager ◆ Text Editor ◆ Icon Editor

Figure 3-1. HP VUE Help Manager Main Window

Once you have started Help Manager, you can learn how to use it by pressing (F1), the "help key." Or choose `Using Help` from the Help menu.

Features of Help Manager

Help Manager includes the following features to make it easier for you to locate and understand information:

- Keyword search.
- Navigational tools such as a "Backtrack" mechanism and a "History" that maintains a list of the topics you have already seen.
- "Hyperlinks" to access related help topics and additional detail.
- Graphics.
- The ability to print help topics.
- An indicator showing your current location in the hierarchy of topics.

Contents of Help Manager

The Help Manager display contains a two-level hierarchy. Help topics for various products are displayed at the first level. (See Figure 3-1). You may have to scroll the Help Manager main window to see all the topics. When you choose one of these topics, a second window opens to display the subtopics within the chosen topic.

Help Manager includes the following topics (it might also include additional help, depending on what optional products you have purchased):

- *HP Visual User Environment Help, Version 3.0* provides information on HP VUE. Refer to this help to learn about HP VUE and its features.
- *HP-UX 9.0 Operating System Help* provides information on using the HP-UX operating system. Refer to this help to perform common HP-UX tasks, determine the cause of HP-UX errors, and locate information about HP-UX manuals.
- If you have a pre-loaded operating system (Instant Ignition), *Welcome Messages* provides an introduction to your new workstation and the HP VUE interface. It also explains how to create user accounts and change passwords.

Getting Context-Sensitive Help for Specific Items

If you are using HP VUE, you can get help for the context in which you are currently working:

3

Table 3-1. Getting Context-Sensitive Help

To get help on this ...	Do this ...
The HP VUE Front Panel	Choose the up arrow above the Help control (the books and question mark) in the Front Panel. The Help subpanel appears (refer to Figure 3-2). Choose `On Front Panel` in the Help subpanel.
A particular item in HP VUE	Either of the following: 1. Move the cursor to the object or icon for which you want help. Then press `F1`, the "help key." *or* 1. Choose the up arrow above the Help control (the books and question mark) in the Front Panel. The Help subpanel appears (refer to Figure 3-2). 2. Choose `On Item` in the Help subpanel. 3. The cursor changes to a question mark (`?`). Choose the object or icon for which you want help.
An HP VUE application	Choose a command from the application's Help menu.
An HP VUE dialog box	Choose the dialog's Help button.

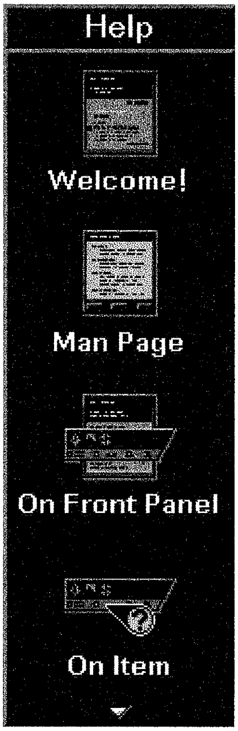


Figure 3-2. The Help Subpanel

Accessing the HP-UX Manual Pages

The *HP-UX Reference* contains reference entries (also called manual pages or "man pages") for every HP-UX command.

These manual pages provide command syntax and a detailed description of the command and its options and arguments. The description may include examples of command usage and provide other information such as system files used and related commands.

Displaying the HP-UX Manual Pages with HP VUE

HP VUE includes a manual page viewer:

1. Choose the up arrow above the Help control (the books and question mark) in the HP VUE Front Panel. The Help subpanel appears (refer to Figure 3-2).
2. Choose `Man Page` in the Help subpanel. A dialog box appears.
3. Type the name of the manual page you want to see. For example, type `cp` to learn more about the *cp* command. Type `man` to learn more about the *man* command (see "Displaying the HP-UX Manual Pages from the Command Line").
4. Choose `Show Man Page` or press (Enter). After a few seconds, an information display appears. You may need to scroll down to read the entire display.

   For command syntax, refer to the "`SYNOPSIS`" section of the manual page. Brackets, [ ], in a syntax statement indicate that the enclosed parameter is optional.
5. To print the manual page, choose the `Print` button.
6. When you have finished looking at the manual page, choose the `OK` button, then choose the `Exit` button in the manual page dialog box.

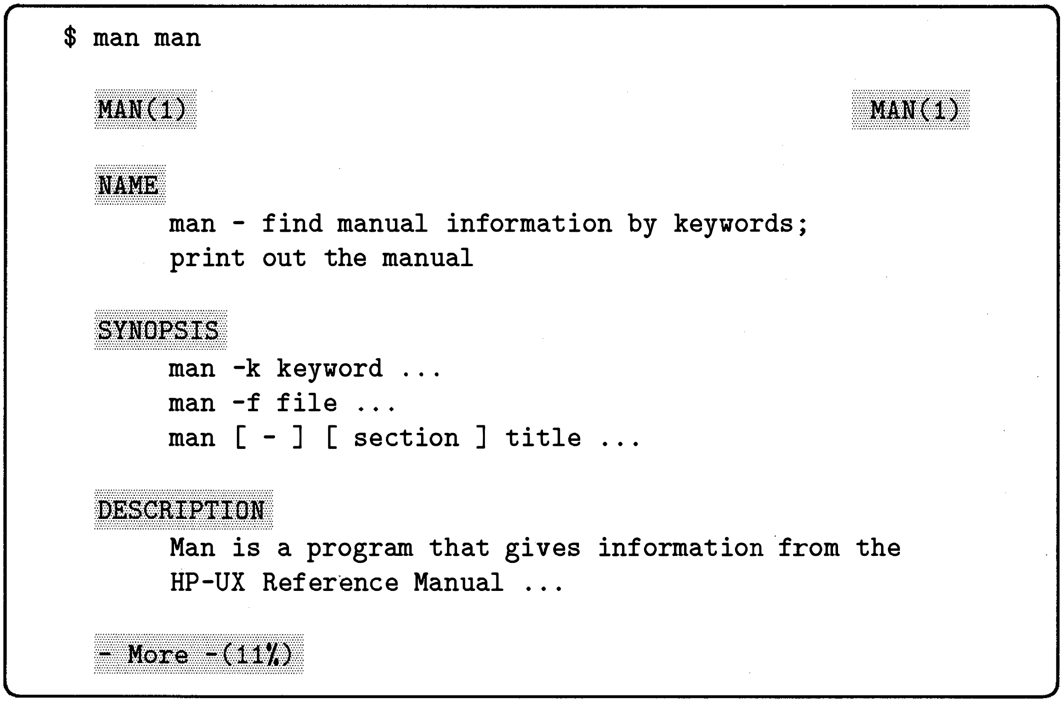
Displaying the HP-UX Manual Pages from the Command Line

Type **man** *command_name* at the command prompt. For example, to learn more about the **cp** command type:

```
$ man cp
```

After a few seconds, an information display appears.

For example, you can even look at the **man** manual page to learn more about the **man** command itself:



```
$ man man

  MAN(1)                                                   MAN(1)

  NAME
       man - find manual information by keywords;
       print out the manual

  SYNOPSIS
       man -k keyword ...
       man -f file ...
       man [ - ] [ section ] title ...

  DESCRIPTION
       Man is a program that gives information from the
       HP-UX Reference Manual ...

  - More -(11%)
```

The message `- More -(11%)` means you've viewed 11% of the file, and 89% remains. (Some systems will just display `- More -`). At this point, you can do any of the following:

- Step through the file a page at a time by pressing the space bar.
- Scroll through the file a line at a time by pressing Return.
- Quit viewing the reference page by pressing q.

For command syntax, refer to the "`SYNOPSIS`" section of the manual page. The above example shows that the `man` command actually has three different command syntaxes:

```
man -k keyword ...
man -f file...
man [ - ] [section] title ...
```

Brackets, [ ], in a syntax statement indicate that the enclosed parameter is optional. (Note that using the "`-`" will cause the `man` display to scroll past as standard output, rather than being processed through `more`).

To print a `man` page for a command named *command*, enter the following:

```
man command | col -b | lp
```

Finding HP-UX Documents

You can view a complete catalog of available HP-UX manuals, including the part number, intended audience, tasks described, and ordering information.

1. Choose the Help control (the books and question mark) in the Front Panel. (Be careful not to choose the up arrow at the top of the control.)
2. Choose *HP-UX 9.0 Operating System Help.*
3. Choose *Finding HP-UX Information.*
4. Choose the appropriate topics in *Finding HP-UX Information* until you locate the manual.

Finding Titles and Part Numbers

To see an alphabetized list containing only current manual titles and part numbers, enter `man manuals`.

Online Manuals: The LaserROM/UX Optional Product

If your system is equipped with CD-ROM, you can purchase an optional online documentation system, LaserROM/UX, to view the HP-UX manuals electronically.

LaserROM/UX has the following features:

- Keyword search with relevance ranking.
- Navigational tools such as an index, table of contents, and "hyperlinks" to more detailed information.
- The ability to create your own "bookshelves" (customized groupings of most-frequently used books), "bookmarks", and annotations.
- The ability to print part or all of a document.

LaserROM is also supported on a number of terminals, including `hp262x`, `hp239x`, `vt100` and `wy50` (`wyse 50`).

Your Hewlett-Packard sales engineer can furnish more information on obtaining LaserROM/UX.

Other Online Sources of Information

Your system also includes other sources of information:

- SAM, the HP System Administration Manager: SAM allows you to perform many system administration tasks without using the underlying HP-UX commands that are associated with the task. SAM can also save you time and keystrokes.

  SAM has complete online help information to help you perform system administration tasks. For more information about SAM, refer to Chapter 12 of this guide.

- Release Notes: This is the online version of the Release Notes which came with your system. It contains all the late information and undocumented changes for your release of HP-UX. Release Notes is found in the `/etc/newconfig` directory, and is named by its release number. For example, `90RelNotes` is the Release Notes for HP-UX 9.0.

- Newconfig: The directory `/etc/newconfig` contains information and new versions of HP-UX product configuration files, as well as shell scripts which may have been customized (localized) on your system. The contents of this directory will vary depending on which products you have loaded on your system. In most cases, old versions of these files, in their regular locations in the file system, are not overwritten by the update process. See the `README` file in `/etc/newconfig` for information on the contents of this directory.

4

Working with Files

Chapter Contents

Creating and Listing Files

A **file** is a named area on your system where you can store information. In using HP-UX commands and applications, you will create files containing a variety of text or data. There are two ways you can use to manipulate and work with files and directories:

If You are Running HP VUE:

You can get a File Manager window view of your curent directory to work with by clicking on the File Manager button (the "file cabinet" icon) at the right side of the Front Panel. The *HP VUE User's Guide* gives you information on using the File Manger.

If You are Using Shell Prompts:

If you are using the HP-UX shell prompt, this chapter tells you how to use the HP-UX directory structure and related commands.

Creating a File with cat

Using the `cat` command, you can create a file containing text. The examples in this chapter assume you've created a text file named **`myfile`**. To create the file, use the `cat` command as follows:

```
$  cat > myfile
```

After you type this command, the cursor sits on a line by itself: you can now type text into the empty file. (Press Return at the end of each line you type.) When you have finished entering text, press Return. Then hold down CTRL and press D. The `cat` command stops and returns you to the command line prompt.

You can use the `cat` command to create your own version of **`myfile`**. For example, you might create the file as follows:

```
$ cat > myfile
The text I am typing will be stored in "myfile."Return
I press RETURN at the end of each line.Return
When I'm finished, I hold down the CTRL key and press D. Return
CTRL-D
```

Note You can also create and edit files using a text editor such as `vi`. To learn how to use this editor, see "Using vi: Commands and Text Entry" in Chapter 8 in this manual.

Listing Files Using ls

To verify that `cat` created `myfile`, run the `ls` command, which lists the names of your files. Running the `ls` command with the file name will confirm that that file exists, but won't list other files.

```
$ ls myfile         The ls command lists myfile.
myfile
```

(Viewing the file's contents is discussed in the section "Viewing and Printing Files", in this chapter)

Naming Files

When you choose a file name, you need to follow certain rules regarding the length of the name and the types of characters you include. If a file name begins with a dot (.), it is "invisible" and the **ls** command normally will *not* list it. To see invisible file names, run **ls** with the **-a** option.

Choosing a File Name

When you choose a file name, remember these rules:

- Generally, file names can contain up to 14 characters (or bytes, in non-ASCII character sets), which can be any combination of the following:
  - □ Uppercase or lowercase letters (A through Z; a through z)
  - □ Digits (0 through 9)
  - □ Special characters, such as: +, -, _, .

  Based on these rules, the following are valid file names:

  ```
  money          Acct.01.87     CODE.c
  lost+found     112.3-data     foo_bar
  ```

- HP-UX interprets uppercase and lowercase letters differently in file names. Thus, the following file names all are different:

  ```
  money      Money      MoneY      MONEY
  ```

Note Some systems may be configured to accept file names longer than 14 characters. However, before you create files with longer names check with your system administrator. If your system is not configured correctly, the longer file names will be truncated to 14 characters, and may cause difficulties.

Invisible File Names

A file name in which the first character is a dot (.) is an **invisible file name**, since the **ls** command does *not* normally display it. Use invisible file names if you don't want or need certain files displayed when you run **ls**.

To illustrate, you have an invisible start-up file that the system runs when you log in. In HP-UX terminology, this file is called a **login script**. It is used to customize your working environment. To learn more about login scripts, see the chapter, "Customizing Login Scripts," in this manual.

Note that the behavior of your system, including the name of your login script, is determined by the type of command interpreter, or **shell**, that your system is using. Common shells used with HP-UX include the Bourne, Korn, C, and Key Shells. The Key Shell is an interactive shell which uses all the same functions as the Korn Shell.

To force `ls` to list invisible file names, including the name of your login script, run it with the `-a` option:

```
$ ls -a                      Use -a to see invisible file names.
.profile    myfile           This is the Bourne Shell, so .profile is
                             shown.
```

If your shell is ...	Then your start-up file is ...
Bourne Shell	`.profile`
C Shell	`.login`
Korn/Posix Shell	`.profile`
Key Shell	`.profile`

Viewing and Printing Files

Using the `more` command, you can look at the contents of a text file. If your system is appropriately configured, you can print a text file using the `lp` command.

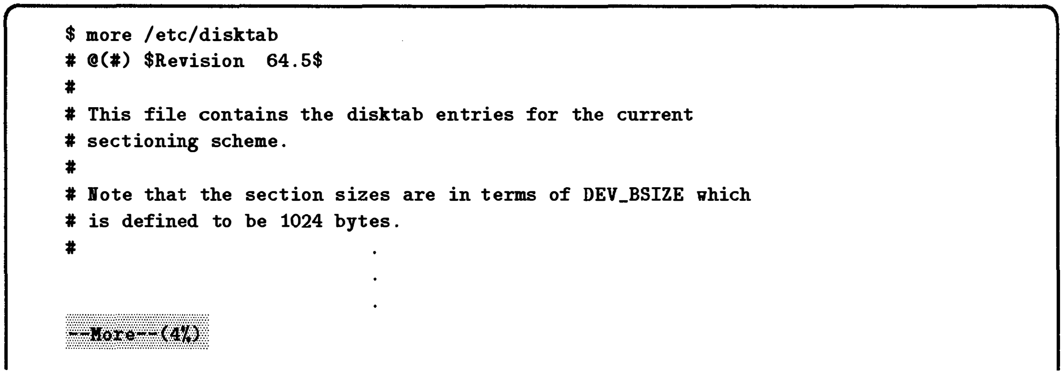
Viewing a File with more

The `more` command displays a text file's contents on the screen. For example, the following `more` command displays the contents of `myfile` (which you created in the section, "Creating and Listing Files"):

```
$ more myfile
The text I am typing will be stored in "myfile."
I press RETURN at the end of each line.
When I'm finished, I hold down the CTRL key and press D.
$
```

If the file contains more lines than are on your screen, `more` pauses when the screen is full. With a longer file, press space to continue looking at additional screens, and press Q when you are finished. Then `more` returns you to the system prompt.

Try running `more` on the system file `/etc/disktab`:



```
$ more /etc/disktab
# @(#) $Revision  64.5$
#
# This file contains the disktab entries for the current
# sectioning scheme.
#
# Note that the section sizes are in terms of DEV_BSIZE which
# is defined to be 1024 bytes.
#                     .
                      .
                      .
--More--(4%)
```

The "--More--(4%)" message at the bottom of the screen means you've viewed 4% of the file thus far, and 96% of the file remains to be viewed. At this point, you can do any of the following:

- Scroll through the file a page at a time by pressing the space bar.
- Scroll through the file a line at a time by pressing (Return).
- Quit viewing the file and leave `man` by pressing (q).

Displaying the First and Last Lines of a File

- To see the first line of a file without opening the file in `vi` or VuePad, you can use the `head` command:

  `head` *filename*

  This will display, by default, the first ten lines of *filename*. (These first ten lines will include blanks). For example:

```
                         CONFERENCE NOTES


Attendees:

 Mary
 Sam
 Nina
 George
 Raphael
 Sergei
```

- To see the last ten lines (default value) of your file, use the `tail` command, as follows:

```
tail %filename%
```

  You will see a display of the last ten lines (including blanks) of *filename*.

Printing a File with lp

If your system is appropriately configured, you can print a text file using the lp (*l*ine *p*rinter) command. Before using the lp command you may need to find out whether your system is set up so that you can use the lp command. If it is not, you can find information on configuring printers in *System Administration Tasks* and *Installing Peripherals*.

You can also use the System Administration Manager (SAM) to configure your system for a printer, after the hardware is connected. See "Getting Information on Printers" in Chapter 12, in this manual, for information about using SAM for this task.

If lp does work on your system, you may also need to find out the location of the printer, on an extensive system. When you have this information, print myfile by running the lp command:

```
$ lp myfile
```

If the lp command is working properly, it should display a message indicating that it sent your file to the printer. For example:

```
request id is lp-number (1 file)
```

The *number* is an i.d. number assigned to the print job by the lp command. If you don't see this message, or if you get an error message, consult your system administrator. You should get a printout with your username displayed on the first page. The time required for a printout depends on the number of tasks being run by the system and the speed of the printer itself.

Getting Printer Information with lpstat

To display a report on the printer status, including the order of your print job in the printer que, type:

```
$ lpstat
```

To get this information with HP VUE, you can click (once) on the Printer icon on the Front Panel.

Cancelling a Print Job with `cancel`

To cancel a print job, enter the `cancel` command, with the i.d. number for your job:

```
$ cancel request_id
```

Renaming, Copying, and Removing Files

To change a file's name, use the `mv` ("*mo*v*e*") command; to make a copy of a file, use the `cp` ("*c*o*p*y") command; to remove a file, use the `rm` ("*r*e*m*ove") command. The examples in this section assume you have created the file `myfile`, as described in "Creating and Listing Files."

Renaming Files with mv

Using the `mv` command, you can rename the file `myfile` to `foofile` as follows:

```
$ mv myfile foofile
```

To verify that `mv` renamed the file, use the `ls` command:

```
$ ls
foofile
```

To rename `foofile` back to `myfile`, type:

```
$ mv foofile myfile
$ ls                    Using ls, verify that the action was successful.
myfile
```

Caution When renaming files, take care not to rename a file to the name of a file which already exists in that directory. If you do this, the file that already has the name will be lost.

For example:

```
$ ls
afile    bfile      If you had these files ...
$ mv afile bfile    And you rename afile to bfile ...
$ lsg               Look what happens ...
bfile               The previous bfile is replaced with the old afile.
```

(The `mv` command can also be used to move files to different locations on the system. This concept is discussed further in Chapter 5 in this manual, "Using Directories to Organize Your Files.")

Copying Files with cp

Copy a file when you want to make a new version of it while still keeping the old version around. For example, to make a new copy of **myfile** named **myfile2**, type:

```
$ cp myfile myfile2
```

Now when you use the **ls** command, you will see the following:

```
$ ls
myfile      myfile2
```

Use **more** to view **myfile2**. You will find that it is the same as **myfile**.

Caution If you copy a file to an existing file, the existing file will be lost.

Removing (Deleting) Files with rm

If you have files that are no longer needed, you should remove (delete) them. Deleting unnecessary files leaves more room for other files on your system. For example, suppose you've finished using **myfile2**, and it is no longer needed. To remove **myfile2**, type:

```
$ rm myfile2
```

To see that **myfile2** was removed, use **ls**:

```
$ ls     The directory listing shows only the other file now remains.
myfile
```

See Chapter 5 for information on how to remove directories and contents.

Comparing the Contents of Two Files: diff

If two text files are known to be similar, and you want to determine what the differences are or which one has been changed:

- First run ll and look at the time at which each file was most recently saved, as in the following example:

  ```
  -rw-r--r--   1 jth      users      1759  Mar 17 15:53 test1
  -rw-r--r--   1 jth      users      2130  Mar 17 15:47 test2
  ```

  4

  You can immediately tell that `test1` was saved more recently than `test2`, as it has the more recent time (and its size was also changed).

- You can tell exactly what the differences are between `test1` and `test2`, by running the command `diff`:

  ```
  diff test1 test2
  ```

For example, if the file contents of `test1` was:

```
Mary had a little lamb.
It's fleece was
white as snow
```

And the contents of `test2` was:

```
Mary had a little lamb.
It's fleece was
blue as sky.
```

The command will display something like the following indicating the differences it found, by line number, and (with the < and >) pointing to which file the difference occurred in:

```
3c3               The relevant line numbers
< white as snow   The version in test1
---
> blue as sky.    The version in test2
```

Joining Two Files

To append to an existing file, you use the `cat` command with two greater-than signs (`>>`). The file name following the `>>` identifies the file to which the contents of the first file is appended. If that file exists, the new data is appended to the end of the file. If the file does not exist, it is created. The command format is:

`cat` *filename2* `>>` *filename1*

where *filename2* is the file whose output is redirected, and *filename1* is the name of the file which is appended to.

This also works with the output of commands. The following example executes the `date` command with the output redirected to append to the `whoison` file:

```
$ date >> whoison                            Append output to whoison.
$ more whoison                               Display contents of whoison.
pat         console      Oct  9 08:50        Output from previous example.
terry       tty01        Oct  9 11:57
kim         tty02        Oct  9 08:13
Tue Oct  9 13:20:16 MDT 1990                 Newly appended output from
                                             date.
$
```

Finding Out Who Can Use Your Files

Files are assigned **access permissions** that control who has permission to read or alter files. If you don't have the necessary access permissions to a file, you may not be able to rename or copy it. If this is the case, the system will display a message indicating that you can't perform the command.

Three classes of users (in various combinations) can access files: *owner*, *group*, and *other*. Each class may access files in various ways: *read permission*, *write permission*, and *execute permission*. You can use the `ll` command to view these file access permissions for individual files.

Access to files is restricted by classes of users. The three basic classes of users are:

- *owner*—Usually the person who created the file (for example, you).
- *group*—Several users who have been grouped together (along with you as the owner of the file) by the system administrator (for example, the members of your department).
- *other*—Any other user on the system.

Each of the above classes can access files in any of these three ways:

- *read permission*—Users with this type of permission can view the contents of a file.
- *write permission*—Users with this type of permission can change the contents of a file.
- *execute permission*—Users with this type of permission can execute (run) the file as a program by typing the file name at the command line prompt.

Using the ll Command to Display File Permissions

The `ll` (*l*ong *l*isting) command displays the permissions for *owner*, *group*, and *other*; `ll` also displays the name of the file's owner and group.

Here is a closer view with all permissions indicated (note that the permissions are in sets of three):

```
 rwx   rwx   rwx
  |     |     |
owner group other
```

To see the permissions, owner name, and group name on `myfile`, for example, type the following:

```
$ ll myfile
```

When you press Return, you should see something like this:

```
-rw-r--r--    1  leslie  users   154        Nov 4 10:18        myfile
permissions      owner   group   size          date            file name
```

The first dash on the left indicates that `myfile` is a file (if `myfile` were a directory, you would see a `d` in place of the dash). The next nine positions indicate read, write, and execute permissions for *owner*, *group*, and *other*. When filled, with all permissions granted for a directory, the permissions look like:

```
drwxrwxrwx
```

If a permission is not allowed, a dash appears in place of the letter.

In the example (`-rw-r--r--`), *owner* (`leslie`) has read and write permission (`rw-`); *group* (`users`) and *other* have only read permission (`r--`).

For More Information

For the procedures to allow you to change permissions on your file and directories see Chapter 14.

Chapter Command Summary

Table 4-1. Commands

To Do This ...	Type This ...
Create a file	`cat >` *filename*
Terminate keyboard input for `cat`	CTRL-D
List visible files in current directory	`ls`
List visible and invisible files in current directory	`ls -a`
View a file	`more` *filename*
Print a file	`lp` *myfile*
Get information on a print job	`lpstat`
Cancel a print job *lpno*	`cancel` *lpno*
Rename ("move") a file	`mv` *fromfile tofile*
Copy a file	`cp` *fromfile tofile*
Delete (remove) a file	`rm` *filename*
Find out access permissions	`ll` *filename*

5

Organizing Files in Directories

- Understanding a Directory Hierarchy.
- Determining Your Location in an HP-UX Directory Hierarchy.
- Specifying Files and Directories: Absolute Path Names.
- Specifying Files and Directories: Relative Path Names.
- Creating Directories.
- Changing Your Current Directory.
- Moving and Copying Files Between Directories.
- Removing Empty Directories.
- File Name Shorthand: Wildcard Characters.
- Displaying Directory Permissions.

Understanding a Directory Hierarchy

HP-UX directories can contain files and other directories. In addition, directories are hierarchically organized. That is, a directory has a parent directory "above" and may also have sub-directories "below." Similarly, each sub-directory can contain other files and also can have more sub-directories. Because they are hierarchically organized, directories provide a logical way to organize files. They are organized the same way in HP-UX and HP VUE, but they may be displayed differently.

With the help of directories, you can organize your files into manageable, logically-related groups. For example, if you have several files for each of several different projects, you can create a directory for each project and store all the files for each project in the appropriate directory.

The structure of an HP-UX directory resembles an inverted tree. These directories (shown in the figure as ovals) usually contain more directories, which in turn create the branching "tree" structure of a typical home directory:

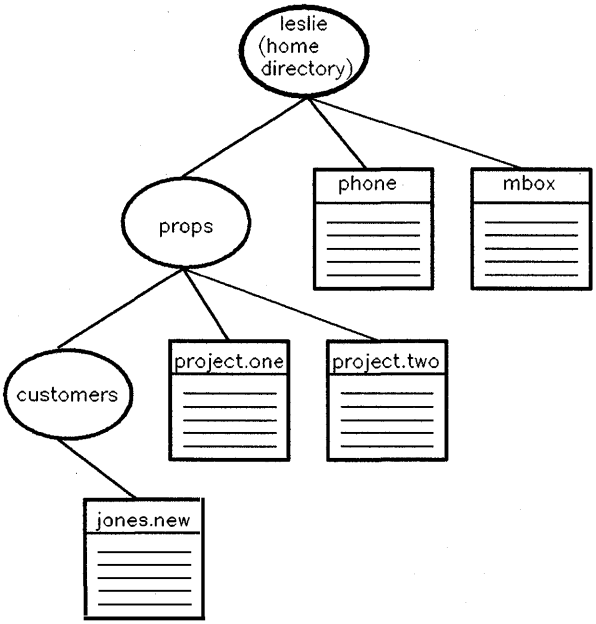


Figure 5-1. A Typical HP-UX Directory Structure

Each directory also contains files (represented below as boxes), which hold actual text, data, or code. At the top of the inverted tree structure is the **root directory**, represented in path names as **/**. Figure 5-2 shows a broader part of a system's directory structure.

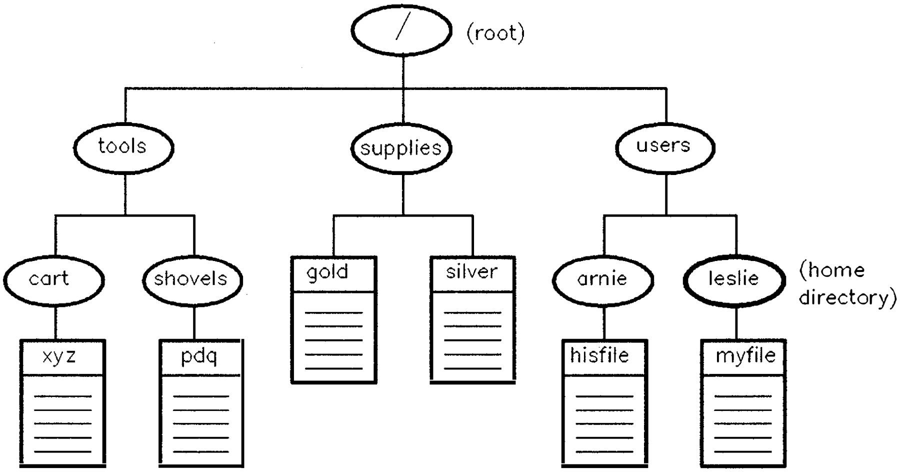


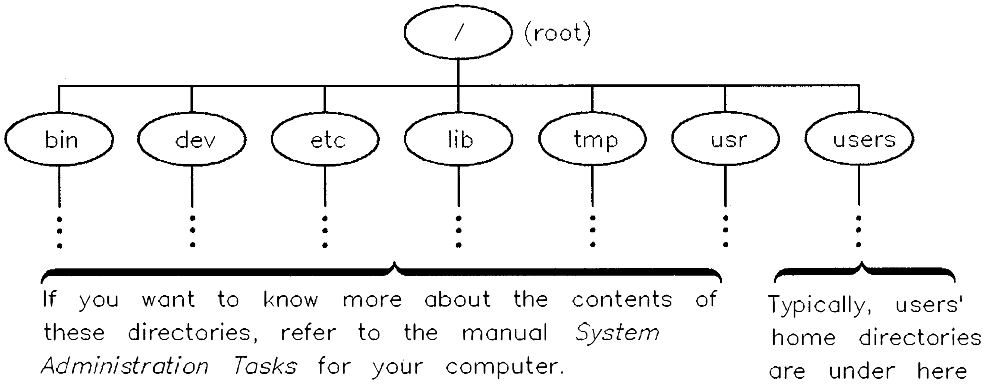
Figure 5-2. A System Directory Structure

Determining Your Location in an HP-UX Directory Hierarchy

This section discusses the HP-UX directory structure and how you specify the location of a file in the structure. All directories fall under the topmost **root** directory, which is denoted by a slash (**/**). When you use HP-UX, you are working in a directory called the **current working directory**. And when you log in, HP-UX places you in your **home directory**.

Figure 5-3 shows the two highest levels of a typical HP-UX directory structure. Each directory, including the root, may contain logically-organized files, as well as more directories.

5



/
(root)
bin
dev
etc
lib
tmp
usr
users
If you want to know more about the contents of these directories, refer to the manual *System Administration Tasks* for your computer.
Typically, users' home directories are under here

Figure 5-3. The HP-UX Directory Structure

When using HP-UX, you are always positioned "in" a directory. The directory you are performing tasks in is known as your **current working directory**. Moreover, whenever you log in, HP-UX places you in a working directory called your **home directory**.

Here is a sample directory hierarchy for a user named Leslie. When Leslie logs in, she is in her home directory, **`leslie`**.

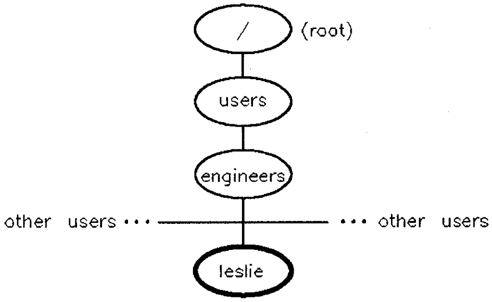


Figure 5-4. Leslie's Home Directory

Specifying Files and Directories: Absolute Path Names

When specifying only files which are in your current working directory, you can refer to them just by their file names. But when referring to directories and files *outside* your current working directory, you must use **path names**, which tell HP-UX how to get to the appropriate directory. An **absolute path name** specifies a path from the root to the directory. A **relative path name** specifies a path from your current directory to another directory.

A path name specifies where a particular file or directory can be found within the directory structure by specifying the directories you need to pass through to get there.

There are two kinds of path names: *absolute* and *relative*.

Absolute Path Names

Absolute path names specify the path to a directory or file, starting from the root directory at the top of the inverted tree structure. The root directory is represented by a slash (/). The path consists of a sequential list of directories, separated by slashes, leading to the directory or file you want to specify. The last name in the path is the directory or file you are pointing to.

To determine the absolute path to your current directory, use the **pwd** (*p*rint *w*orking *d*irectory) command. The **pwd** command displays the "path" from the root directory to your current working directory.

Here is an example of an absolute path, displayed with the **pwd** command:

```
$ pwd
/users/engineers/leslie
```

This specifies the location of the current directory, `leslie`, by starting from the root and working down.

Figure 5-5 shows the absolute path names for various directories and files in a typical directory structure:

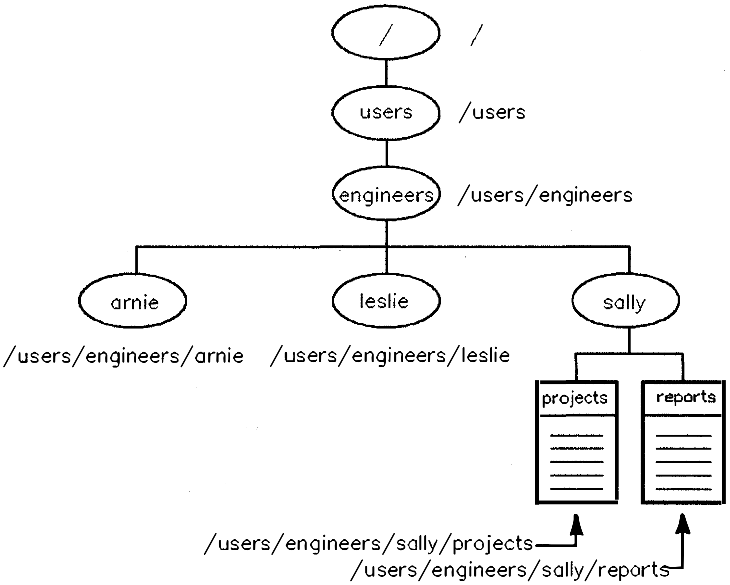


Figure 5-5. Absolute Path Names

Specifying Files and Directories: Relative Path Names

You can use a *relative* path name as a shortcut to the location of files and directories. Relative path names specify directories and files starting from your current working directory (instead of the root directory).

You will frequently find it convenient to use relative path names. The following table shows some common path name shortcuts.

Table 5-1. Examples of Relative Path Names

This relative path name . . .	**Means . . .**
.	The current directory.
..	The parent directory (the directory above the current directory).
../..	Two directories above the current directory.
directory_name	The directory below the current directory.

For example, suppose the current directory (as shown in Figure 5-6) is `/user/engineers/leslie`. To list the files in the directory above (which is `/user/engineers`), enter:

```
$ ls ..
arnie  leslie  sally   you get a listing of /user/engineers
```

On the other hand, to get a listing of the files in a directory immediately below your current directory, simply enter the directory name. For example, to get a listing of the files in the `projects` directory, below the current directory `/user/engineers/leslie`, you would enter:

```
$ ls projects
$                 The projects directory is empty!
```

Figure 5-6 shows relative path names for various directories and files starting from the current directory, **/users/engineers/leslie**.

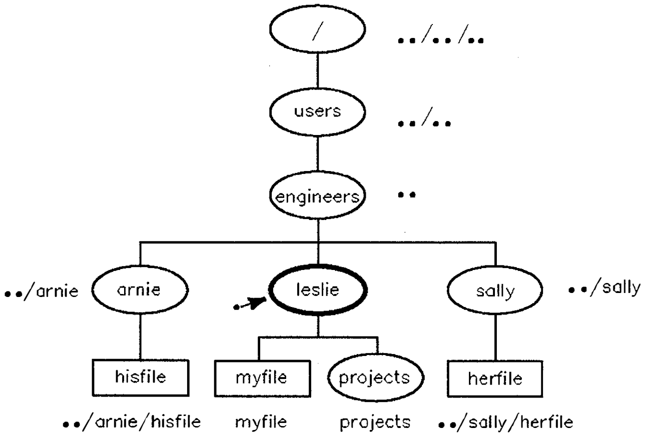


Figure 5-6. Relative Path Names from /users/engineers/leslie

Creating Directories

To create a directory, use the `mkdir` command. To get a directory listing that differentiates files from directories, use the `lsf` command instead of `ls`.

The `mkdir` (make directory) command creates a new directory. After you create a directory, you can move files into it, and you can even create more directories underneath it. For example, to create a sub-directory in your current working directory named `projects`, type:

```
$ mkdir projects
```

To verify that it did what you expected, you can use either the `ls` or `lsf` command. Both commands display the new directory, but `lsf` appends a slash (/) to the end of directory names to differentiate them from file names. For example:

```
$ ls                List files, directories in your current working directory.
myfile  projects    It did what you expected.
$ lsf
myfile  projects/   The lsf command appends a slash to directory names.
```

Figure 5-7 shows the resulting directory structure.

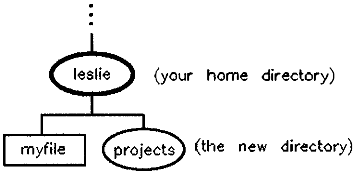


Figure 5-7. Creating the "projects" Directory

The general form of the **mkdir** command is as follows:

```
mkdir new_dir_path
```

where *new_dir_path* is the path name of the directory you want to create. For example, to create a new directory named **old** under the projects directory, type:

```
$ mkdir projects/old
```

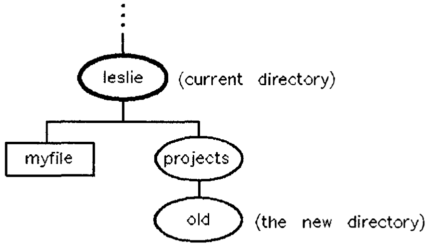


Figure 5-8.
Structure after Creating "old" under the "projects" Directory

Finally, let's create one more directory, named **new**, and verify with **lsf**:

```
$ mkdir projects/new
$ lsf projects
new/      old/          Files and directories are listed alphabetically.
```

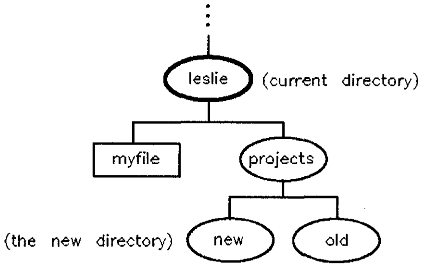


Figure 5-9. Structure after Creating "new" under "projects"

Changing Your Current Directory

Now that you've learned how to create directories under your home directory, you're ready to learn how to move into different directories, using the `cd` command.

Using the `cd` ("change directory") command, you can change your current working directory. For example,

```
$ cd projects
```

moves you into the directory `projects` (which you created in the section "Creating Directories"). To verify that you have, in fact, changed your current working directory, use the `pwd` command, which displays your current directory. For example, if your home directory was `/users/leslie`, then, after you run the "`cd projects`" command, `pwd` would display the following:

```
$ pwd
/users/leslie/projects
```

When you're in the new directory, you can list its contents using `lsf`:

```
$ lsf                         Here are the directories you created earlier.
new/      old/
```

To move into the directory `new` under `projects`, type:

```
$ cd new
$ pwd                            Verify where you are.
/users/leslie/projects/new       It did what you expected.
```

Now if you run `lsf`, it won't display anything because there are no files or directories under `new`:

```
$ lsf
$
```

Remember that `..` is the relative path name for the parent directory of your current working directory. So to move up one level, back to `projects`, type:

```
$ cd ..
$ pwd                        Show your current working directory.
```

```
/users/leslie/projects   It was successful.
```

If you run cd without a path name, it returns you to your home directory as the following example illustrates:

```
$ cd
$ pwd            Are you back home?
/users/leslie    Yes!
```

Experiment with the cd and pwd commands to move around your directory structure. If you become lost, don't panic; just remember that you can run

```
$ cd
```

to return to your home directory. You can also get to any directory using its absolute path name. For example, to change to the projects directory in the example hierarchy, enter:

```
cd /users/leslie/projects
```

Figure 5-10 illustrates how various cd commands change your current working directory. The example assumes you're starting at the directory /users/leslie/projects, and that your home directory is /users/leslie.

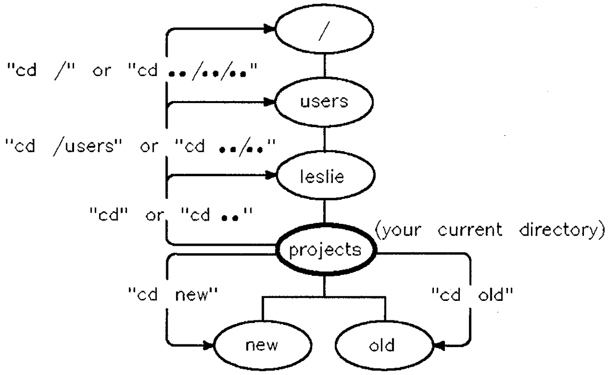


Figure 5-10. Effect of Various "cd" Commands

Moving and Copying Files between Directories

The mv command lets you move a file from one directory to another. With the cp command, you can copy a file into a different directory.

Moving Files

In addition to renaming files, the mv command can be used to move files from one directory to another. For example, to move myfile into the projects directory, type:

```
$ cd                              Move to your home directory first.
$ mv myfile projects
```

Now verify that it did what you intended:

```
$ lsf                             List your current working directory.
projects/                         Where did myfile go?
$ lsf projects                    Look in the projects directory.
myfile     new/      old/         myfile is there, indicating success.
```

Remember that a single dot (.) for a path name represents your current working directory. Therefore, to move myfile from the projects directory back to your current working directory, type:

```
$ mv projects/myfile  .           Don't forget the dot.
$ lsf                             List your current working directory.
myfile     projects/              It worked: myfile is back.
$ lsf projects                    List projects.
new/       old/                   The file myfile isn't there anymore.
```

The general form of the mv command is as follows:

mv *from_path to_path*

where *from_path* is the file name or path name of the file you want to move, and *to_path* is the name of the path where you are moving the file.

Copying Files

To copy a file into a different directory, use the cp command. For example, to make a copy of myfile named myfile2 in the projects directory, type:

```
$ cp myfile projects/myfile2

$ lsf
myfile    projects/      The file myfile still exists.
$ lsf projects
myfile2   new/   old/    The copy (myfile2) is in the projects directory.
```

To make a new version of myfile2 named myfile3 in your current directory, type:

```
$ cp projects/myfile2 myfile3
$ lsf
myfile    myfile3    projects/
```

The general form of the cp command is as follows:

cp *from_path to_path*

where *from_path* is the file name or path name of the file you want to copy, and *to_path* is the path name of the directory or file to which you are copying.

Caution *When moving or copying files, be careful not to destroy an existing file.* For example, if you type the following cp command:

```
$ cp myfile3 projects/myfile2
```

Then, a copy of myfile3 is moved into projects/myfile2, overwriting myfile2. *The previous contents of* myfile2 *are lost.*

If you copy a file to a directory, even if the directory has the same name as the file, the directory will not be destroyed. But a file of the same name *in* that directory would be.

As a general rule, before using mv or cp, use ls or lsf to ensure that the target file name to which you want to move or copy doesn't already exist.

Removing Directories

You can remove an unused directory with the rmdir command. If the directory has any visible or invisible files or subdirectories in it, you will have to first remove these before rmdir will work. You can do this in one step, using the rm command with the -rf option, but you must first be sure you aren't removing any wanted files or subdirectories.

After you have removed a directory, you can no longer use it, and it will no longer appear in an ll or other listing of the directory above it.

Removing a Directory with rmdir

Before removing a directory with rmdir, you must remove its files and any directories under it. For example, suppose you want to remove the projects directory and the files it contains. Figure 5-11 shows how this structure might look:

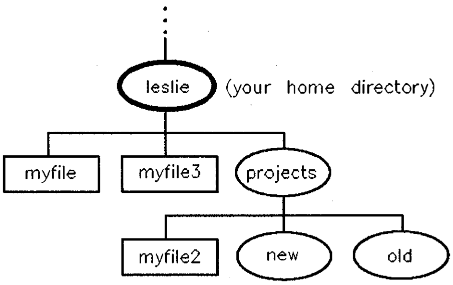


Figure 5-11. The "projects" Directory Structure

To remove this structure, run the following sequence of commands:

```
$ cd                                  Move back to your home directory

$ lsf                                 List the files and directories.
myfile    myfile3    projects/

$ rmdir projects                      Try to remove projects.
rmdir: projects not empty             It won't let you.
```

```
$ cd projects
```

Change directory to `projects`*.*

```
$ lsf
myfile2    new/   old/
```

List its contents.

```
$ rm myfile2
```

Remove the file `myfile2`*.*

```
$ lsf
new/   old/
```

The file `myfile2` *is gone.*
Remove the directory `new`*. If it's empty,* `rmdir` *removes it.*

```
$ rmdir new
```

```
$ lsf
old/
```

The action was successful.

```
$ rmdir old
```

Now remove the directory `old`*. If it's empty,* `rmdir` *removes it.*

```
$ lsf
```

There is no message; the action was successful.

```
$ cd                            Now move back to your home directory ...

$ rmdir projects                And remove projects.
$ lsf                           This will verify that it worked.
myfile    myfile3
```

Removing a Directory and Contents with rm -rf

To avoid the complexity of emptying a directory before you can remove it, you can remove a directory *and all its files and directories* in one action by typing the following:

```
rm -rf dirname
```

Caution Use `rm -rf` with great caution, since it does remove a directory and all its contents, irretrievably, in one action.

File Name Shorthand: Wildcard Characters

Wildcard characters provide a convenient shorthand for specifying multiple file or directory names with one name. Two of the most useful wildcard characters are * and ?. The * matches any sequence (string) of characters (including no characters), and the ? matches any one character.

The * Wildcard

The * wildcard means "any characters, including no characters." Suppose you have created the following files in your current working directory:

```
$ lsf
myfile    myfile2    myfile3    xenic    yourfile
```

To list only the file names beginning with "myfile," type:

```
$ lsf myfile*
myfile    myfile2    myfile3
```

Even though xenic and yourfile exist, lsf displays only the file names that start with myfile. If you wanted to list file names containing "file," type:

```
$ lsf *file*
myfile    myfile2    myfile3    yourfile
```

The ? Wildcard

The ? wildcard means "any single character." Although you probably won't use the ? wildcard as much as *, it is still useful. For instance, if you want to list only the files that start with myfile and end with a single additional character, type:

```
$ lsf myfile?
myfile2   myfile3
```

The ? wildcard character matches *exactly one character.* Thus, myfile didn't show up in this listing because it didn't have another character at the end.

Using the * Wildcard Character with mv, cp, and rm

Wildcard characters are often useful when you want to move or copy multiple files from one directory to another. For example, suppose you have two directories immediately below your current directory, named `new` and `old`, and these directories contain the following files:

```
$ lsf new
myfile    myfile2
$ lsf old
myfile3   myfile4
```

To move all the files from the directory `new` into the directory `old`, type:

```
$ mv new/* old
$ lsf new                                 The files are no longer in new.
$ lsf old                                 They are in the directory old.
myfile   myfile2   myfile3   myfile4
```

You can do a similar operation with the `cp` command. For example, to copy all the files from `old` into `new`, type:

```
$ cp old/* new
```

Similarly, you can use wildcard characters with the `rm` command.

Caution Be careful when using wildcards that you don't accidentally remove files you need.

For example, to remove all the files in the directory `new`, type:

```
$ rm new/*
$ lsf new
$            All the files are gone!
```

For More Information . . .

See *regexp*(5) in the *HP-UX Reference* for general features of `*` and `?`. For additional features relating to individual shells: if you use the Korn Shell, refer to *ksh*(1); if you use the C shell, refer to *csh*(1), both in the *HP-UX Reference*.

Permission to Access Directories

Three classes of users (in various combinations) can access directories: *owner*, *group*, and *other*. Each user class can access directories in various ways: read permission (`r`), write permission (`w`), search permission (`x`). Search permission means that you can search the contents of the directory (for example, you can view the contents of files in the directory with `more`). Use the `ll` command to view directory permissions.

Finding Out Who Can Use Your Directories

In "Finding Out Who Can Use Your Files" in Chapter 4, you learned that your files are accessible by three basic classes of users. Directories are accessible by the same three classes:

- *owner*—Usually the person who created the directory. For example: you.
- *group*—Several users (including you) who have been grouped together by the system administrator. For example: you and the members of your department.
- *other*—Any other user on the same system.

Similar to permissions for files, each of the above classes may have read or write permission on a directory. Although you can't "execute" a directory, directories have "search" permission, which means that you can access the contents of files in the directory with such commands as `more`.

Table 5-2 shows what various types of permissions mean for directories and for files.

Table 5-2. A Comparison of Permissions for Directories and Files

This permission ...	Means this for a directory ...	Means this for a file ...
read (`r`) *permission*	Users can view the names of the files contained in that directory.	Users can view the contents of the file.
write (`w`) *permission*	Users can create, rename, or remove files contained in that directory.	Users can change the contents of the file.
execute (`x`) *permission*	Users can search for (access) files contained in that directory. For example, with search permission, users can use `more` to view the contents of files in the directory.	Users can execute (run) the file as a program by typing the filename at the command line prompt.

Displaying Directory Permissions: The ll Command

You can display access permissions for a directory, as for a file, with the **`ll`** command. To display permissions for a specific directory, use the **`ll`** command with the **`-d`** option.

To display permissions showing *owner*, *group*, and *other* for a specific directory, use the `ll` command with the `-d` option. For example to see the permissions on the `projects` directory below the current directory, type the following:

`$ ll -d projects` *Follow the* `ll` *command with a* `-d` *and the directory name.*

When you press Return, you should see something like this:

```
drwxr-x---  1  leslie  users      1032 Nov  28 12:38 projects
```

The first character (`d`) in the long listing above indicates that `projects` is a directory. The next nine positions (three sets of three) indicate the presence or absence of read (`r`), write (`w`), and search (`x`) permissions for *owner*, *group*, and *other*. If a permission is not allowed, a dash appears in place of the letter. Here is a closer view with all positions indicated:

```
    d        rwx    rwx    rwx
    |         |      |      |
directory  owner  group  other
```

Then, in the original example above (`drwxr-x---`):

owner (`leslie`) has read, write, and search permission (`rwx`); *group* (`users`) has read and search permission (`r-x`); *other* has no access (`---`) to the `projects` directory.

For More Information ...

See Chapter 14, in this manual for information on changing permissions and on the default access permissions for files and directories.

See the *chmod*(1m) reference in the *HP-UX Reference* for more information on using the `chmod` command.

To learn more about the `ll` command, refer to the *ll*(1) reference in the *HP-UX Reference*.

Chapter Command Summary

Table 5-3. Commands

To Do This ...	**Type This ...**
List files; show directories with "/"	`lsf`
Change directory	`cd` *directory_path*
Change to home directory	`cd`
Display working directory	`pwd`
Remove an (empty) directory	`rmdir` *directory_name*
Remove a directory and contents	`rm -rf` *directory_name*
Display permissions for a directory	`ll -d` *directory_name*

6

Using the Shell Command Interpreter

Chapter Contents

Understanding Command Syntax

HP-UX has many useful commands which will help you handle data and text, do system administration tasks, and find information. Most of the commands you've used thus far are easy to enter, that is, they are either a command without any arguments (`whoami`), or a command whose only argument is a file name (`mkdir projects`). HP-UX commands can also be more complex, having additional options, arguments, or both.

Options change a command's behavior. For example, in Chapter 4, you used the `-a` option to change the behavior of the `ls` command so you could list invisible file names. In general, command options are preceded by a dash (`-`). **Arguments** provide additional information needed by the command. For example, an argument may consist of the file name to run the command on.

Examples Using Options

When used without any options, the `rm` command removes a file without verifying whether you really want to remove it. Suppose, for example, your current working directory contains these files: `myfile`, `myfile1`, `myfile2`, `myfile3`, and `myfile4`. You could remove all these files by typing this command:

```
$ rm my*
$            All the files are removed, no questions asked.
```

For safety, if you want `rm` to prompt you for verification before removing each member of a set of files, you can use the `-i` option:

```
$ rm -i my*
myfile1: ? (y/n)                For each file, the system asks if you
                                really want to remove the file. Type y
                                to remove this file; n to leave it alone.
```

```
myfile1: ? (y/n) y
myfile2: ? (y/n) y
myfile3: ? (y/n) y
myfile4: ? (y/n) n               You don't want to remove this file, after
                                 all.

$ ls
myfile4                          It worked: rm  did not remove myfile4.
```

If you are using `rm` non-interactively and the file does not have write permission (for example, with `444` permission), then a message "*filename:* `444 mode ? (yes/no)`" will be displayed. Respond with `y` if you want to remove the file.

Examples Using Arguments

The `cal` command displays an English calendar for the current month. With multiple command arguments, you can specify which calendar month and year to display. For example, to display a calendar for September, 1992, type the `cal` command as follows:

```
$ cal 9 1992
  September 1992
 S  M Tu  W Th  F  S
       1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

If you just use `1992` as the argument, you will get a complete calendar for that year. (Be sure you include the century, `19`. If you use `92` as the argument, you will get a calendar for the year 92 A.D.)

6

Enclosing Arguments in Apostrophes

When a *single* command argument contains embedded blanks, you must enclose it between apostrophes ('word1 word2') . For example, the following **`grep`** command displays all lines in **`myfile`** containing "**`I am`**":

argument

```
$ grep 'I am' myfile
The text I am typing will be stored in "myfile."
```

Running Multiple Commands on the Same Command Line

Occasionally, you may find it useful to run two or more commands on the same command line. To do so, separate the commands with a semicolon, as illustrated below:

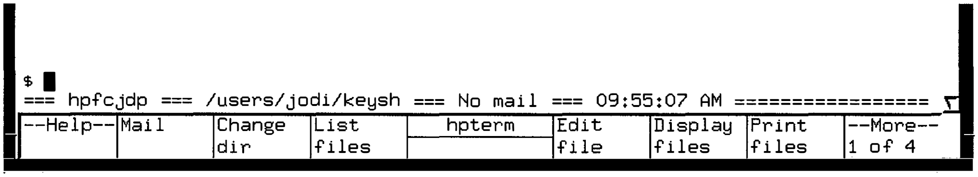
```
$ whoami ; date
leslie                          Output from whoami
Fri Oct  5 15:51:57 MDT 1990    Output from date
```

Entering Commands with the Key Shell

The Key Shell (**keysh**) is an extension of the Korn Shell, using softkey menus and context-sensitive help to help you with command options and syntax. The Key Shell automatically translates softkey commands into HP-UX commands when you press (Return) to enter the command line.

Using the Key Shell Displays

The Key Shell gives you softkey displays at the bottom of your screen which provide a "menu" of basic Korn Shell commands, along with their options in sequence. Get a Key Shell just as you would another shell. For example, to spawn a Key Shell "child" shell, enter the command **/usr/bin/keysh**. (Exit this shell by entering: **exit**.) You will first see a status line like the following



```
$ █
=== hpfcjdp === /users/jodi/keysh === No mail === 09:55:07 AM ================
--Help--  Mail     Change   List        hpterm      Edit     Display  Print    --More--
                   dir      files                   file     files    files    1 of 4
```

Figure 6-1. Key Shell Softkey Display

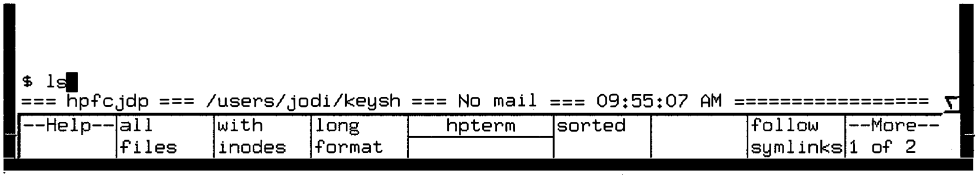
You can enter commands from the Key Shell softkey menu or you can enter standard HP-UX commands as usual. If you enter standard HP-UX commands, the Key Shell will often display an appropriate left-to-right set of menu options in the softkey label area at the bottom of your screen. Each label corresponds to a softkey, f1 through f8. The hpterm at the center separates the softkeys into groups of four. You may select any or none of the options successively by pressing the corresponding softkey.

When you want to see more commands, or more options to go with a command you've already chosen, press the --More-- softkey, f8. This will cause the Key Shell to display the next "bank" of softkeys in sequence, eventually cycling back to the first, if you press f8 repeatedly.

After you make a selection by pressing a softkey, your choice will appear on the command line in "English," just as it appeared in the softkey display, with the correct order and spacing.

Entering a Command with the Key Shell

For example, enter the `ls` command. You will see the following:

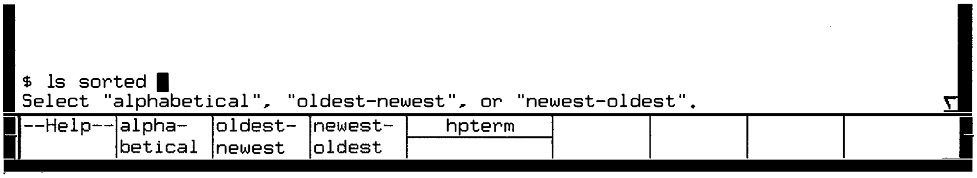


```
$ ls
=== hpfcjdp === /users/jodi/keysh === No mail === 09:55:07 AM =================
```

Figure 6-2. Options Displayed

Many softkey commands require that the user enter a parameter or select an additional softkey before pressing (Return). A "prompt line" underneath the command line will indicate whether you need to enter anything else.

If you select **ls**, and then select the "sorted" option, the Key Shell will ask you to specify *how* you want your file listing sorted:

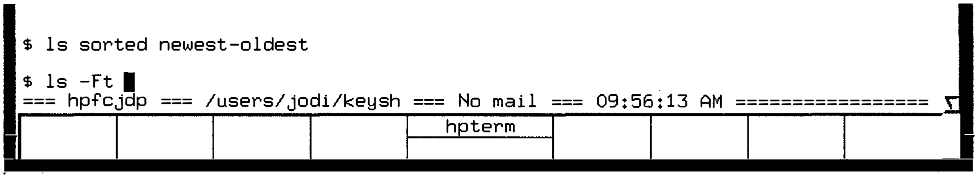


```
$ ls sorted █
Select "alphabetical", "oldest-newest", or "newest-oldest".
```

Figure 6-3. Required Options Requested

At any time, you can use the --Help-- softkey, f1, to find out more about what functions are available to you.

Suppose you have selected "newest-oldest" for the **sort** option above. You can now enter the finished command line by pressing Return. Or, if you want to preview the HP-UX commands to which these "English" words correspond, you can optionally press Insert line and the HP-UX commands will be displayed as shown in Figure 6-4.



```
$ ls sorted newest-oldest

$ ls -Ft █
=== hpfcjdp === /users/jodi/keysh === No mail === 09:56:13 AM ================
```

Figure 6-4. Optional HP-UX Commands Display

Customizing Your Key Shell Softkeys

You can change the Key Shell's configuration (for example, status line or options) using the `Keysh_config` softkey in the `--More-- 4 of 4` display. Any changes you make will be automatically saved in the `.keyshrc` file in your home directory. This file will then be replayed upon subsequent invocations of `keysh`.

If they are not already on, some global options that you can change using `Keysh_config` are:

Table 6-1.

To Enable These Options ...	Enter These Softkeys and Press Return ...
`--Help--` softkey	`Keysh_config` `options` `help` `on`
Automatic prompt messages	`Keysh_config` `options` `prompts` `on`
Visible HP-UX command translations	`Keysh_config` `options` `translations` `on`

6

To turn off any of these options, enter `off` at the end of the entry sequence instead of `on`.

Status-Line Indicators you can change, using the `Keysh_config` softkey are:

Table 6-2.

To Enable Additional Status-Line Indicators	Enter These Softkeys and Press (Return)
Host name	`Keysh_config` `status_line` `host_name` `on`
User name	`Keysh_config` `status_line` `user_name` `on`
Current directory	`Keysh_config` `status_line` `current_dir` `on`
Mail status	`Keysh_config` `status_line` `mail_status` `on`
Date	`Keysh_config` `status_line` `date` `on`
Time	`Keysh_config` `status_line` `time` `on`

In addition, the contents of the **$KEYSH** shell variable, if set, is also displayed on the status line.

6

Restarting and Undoing the Key Shell Configuration Changes

- If you want an existing Key Shell to take on the behavior you've specified in another Key Shell, you will need to enter the following to "replay" the current configuration sequence.

  ```
  Keysh_config restart
  ```

- You can undo any configuration changes you may have made since last invoking a Key Shell by entering the following command:

  ```
  Keysh_config undo
  ```

Pressing undo a second time will "toggle" the configuration back to the previous state again, restoring your changes.

Summary of Key Shell Procedures

The general rules for using the Key Shell are:

- Select any desired softkeys from left to right.
- Use the `--More--` softkey to see more options.
- Optionally, use the Insert line key to preview the translated command-line.
- Use the `--Help--` softkey to find out more functions.

If you make an error, use Back space or CTRL-h to erase the line back to where you want to re-enter command text, just as you would with commands in **ksh**, **csh**, or **sh**.

You can also use the arrow keys, Clear line, Delete line, Insert char, and Delete char to manipulate your command line, in addition to using the editor which is set for your Korn Shell (*see* Chapter 11). Note that Clear line functions to delete the line only from the cursor position to the end of the line. Delete line, however, deletes the entire command line and cancels the command.

6

Chapter Command Summary

Table 6-3. Commands

To Do This ...	**Type This ...**
Delete (remove) a file interactively	`rm -i` *filename*
Run several commands on same line	*command*; *command2*
Show Additional Key Shell choices	`--More--`
Change Key Shell configuration	`Keysh_config`
Find HP-UX command information on-line	`man` *command_name*

7

Using Shell Processes

HP-UX uses command interpreter programs called shells which enable the user to interact easily with the operating system. This chapter introduces you to some of the features of shells.

Chapter Contents

Transferring Data Among Files and Programs

The shell interprets your keyboard commands for the HP-UX operating system to act on. When you log in, you are said to be "in" a **shell**. After the shell interprets a command line, HP-UX loads into memory the corresponding program. When a program is running, it is called a **process**. HP-UX assigns every process a unique number, known as a **process identifier** (**PID**).

How Processes are Created

When you log in, HP-UX starts your shell. During login, HP-UX copies the shell program from system disk into memory. When it is in memory, the shell begins executing, and it becomes a process that lasts until you log out. **Process**, then, refers to the copied program that is actively executing in memory, while **program** is the file stored on the disk.

Similarly, the commands you type create processes. After you type a command line, the following events take place:

1. The shell interprets the command line and searches the disk until it finds the requested program.
2. The shell asks HP-UX to run the program; then control transfers from the shell to HP-UX.
3. HP-UX copies the specified program from a disk file into memory. When the program resides in memory, it begins executing—and a process is created.
4. Each process is assigned a **Process Identifier** or **PID**. You can find out what processes are currently running on your system by typing `ps -ef`.
5. When a program finishes executing, control transfers back to the shell, and the process disappears.

Stopping a Process with kill

Normally, processes can be terminated by entering the following, where *PID* is the identification number for the process you want to get rid of.

```
kill PID
```

The PID for the process is determined by running `ps -ef` and noting the name and process i.d. from the displayed list.

In some cases, the process may ignore the **kill** signal, and you will find it still running after you have issued the **kill** command correctly. If this happens, enter the following:

```
kill -9 PID
```

Run `ps -ef` to confirm that the process has been deleted.

Using Standard Input, Standard Output, and Standard Error

Each process opens three standard "files": standard input (`stdin`), standard output (`stdout`), and standard error (`stderr`). Programs use these as follows:

- **Standard input** is the place from which the program expects to read its input. By default, processes read `stdin` from the keyboard.
- **Standard output** is the place the program writes its output. By default, processes write `stdout` to the terminal screen.
- **Standard error** is the place the program writes its error messages. By default, processes write `stderr` to the terminal screen.

Figure 7-1 illustrates the relationship of these files to the process.

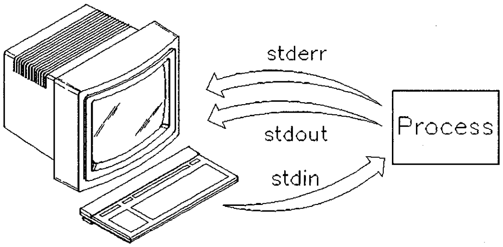


Figure 7-1. Standard Input, Standard Output, and Standard Error

Examples Using the Standard Files

The following three examples illustrate standard output, standard input, and standard error, respectively:

1. Standard Output

```
$ whoami   The whoami command begins the process.
terry      Standard output is displayed on the screen.
$ _        Control returns to the shell.
```

The above example illustrates standard output. In this example, the **whoami** command uses standard output to display the user name of the person typing the command. The prompt returns, indicating that the shell is ready for another command.

2. Standard Input

```
$ sort     The sort command uses standard input.
duffy      Enter standard input at the keyboard.
muffy
daffy
CTRL-D     End of standard input.
daffy      Standard output is displayed on the screen.
duffy
muffy
$ _        Control returns to the shell.
```

The above example uses the **sort** command to sort text typed at the keyboard. Typing CTRL-D ends standard input. The standard output is displayed on the terminal screen.

3. Standard Error

```
$ mroe memo         You have misspelled a command.
mroe: not found     Standard error is displayed on the screen.
$ _                 Control returns to the shell.
```

The standard error example illustrates what happens if you misspell a command. The typing error causes the shell to generate an error message that is sent to standard error. Again, the prompt returns, indicating that the shell is ready for another command.

For More Information ...

Your shell has many productivity-enhancing functions that you will find useful. To learn more about shell processes, including the standard files, see Chapter 11, and refer to *Shells: User's Guide*.

Writing Standard Output to a File

The shell lets you *redirect* the standard output of a process from the screen (the default) to a file. Redirecting output lets you store the text generated by a command into a file; it's also a convenient way to select which files or devices (such as printers) a program uses. To *redirect* a process's output, separate the command and the output file name with a greater-than sign (>) pointing from the command to the output file. To *append* the output to an existing file, use two greater-than signs (>>) pointing to the file to be appended on.

Redirecting Standard Output

You can use output redirection with any command that writes its output to `stdout` (your screen). You can redirect output either to a new file or to an existing file. To redirect output, use a greater-than sign (>). The word following the sign identifies the file name where the `stdout` data is to be written. If the file exists, its previous contents are lost. If the file does not exist, it is created. In its simplest form, the command syntax is as follows:

command > *outfile*

where *command* is the command whose output is redirected, and *outfile* is the name of the file to which the process writes its standard output.

The example below shows output redirection using the `who` command, which displays a list of users currently logged in to the system. Instead of displaying the users on the terminal screen, the output is redirected to the file `whoison`. The `more` command lists the contents of the `whoison` file, showing that the output redirection was successful:

7

```
$ who > whoison                               Redirect output to whoison.
$ more whoison                                Display contents of whoison.
pat         console      Oct  9 08:50
terry       tty01        Oct  9 11:57
kim         tty02        Oct  9 08:13
$
```

Figure 7-2 illustrates where **stdin**, **stdout**, and **stderr** go when output is redirected to a file.

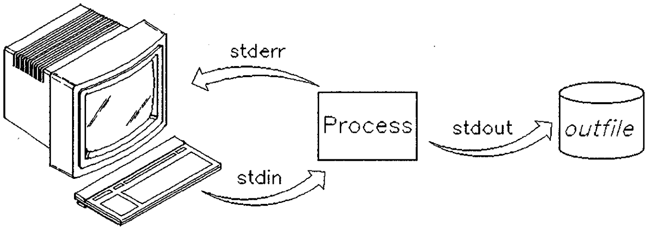


Figure 7-2.
Standard Input, Output, and Error When Output Is Redirected

Using Files for Standard Input

The shell lets you redirect the standard input of a process so that input is read from a file instead of from the keyboard. To redirect the input of a process, separate the command and the input file name with a less-than sign (<) directed at the command name.

Your shell can redirect the data flow into a program so that the keyboard is not used. Input that is normally typed at the keyboard can be redirected to be read from a file. You can use input redirection with any command that accepts input from `stdin` (your keyboard). You cannot apply input redirection to commands such as `who` that do not accept input from `stdin`.

To redirect input, use a less-than sign (<). The word following the sign identifies the file from which the `stdin` data is read. The file must exist for the redirection to succeed. In its simplest form, the command syntax is as follows:

command < *infile*

where *command* is the command whose input is redirected, and *infile* is the name of the file from which the process reads standard input.

The following example illustrates input redirection. First, standard output from the `who` command is redirected to a file named `savewho`. Second, the `more` command displays the contents of `savewho`. Finally, standard input for the `wc` (word count) command is redirected to come from the `savewho` file:

```
$ who > savewho                                  Redirect output to savewho
$ more savewho                                   Display contents of savewho
pat          console       Oct  9 08:50
terry        tty01         Oct  9 11:57
kim          tty02         Oct  9 08:13
kelly        tty04         Oct  9 10:04
$ wc -l < savewho                                Redirect input from savewho
             4
$ _
```

In the preceding example, the `wc` command with the `-l` option counts the number of lines in the input file. Because input is redirected from `savewho`, this number equals the number of users logged in to the system when the `who` command was executed.

Figure 7-3 illustrates where `stdin`, `stdout`, and `stderr` are directed when input is redirected from a file.

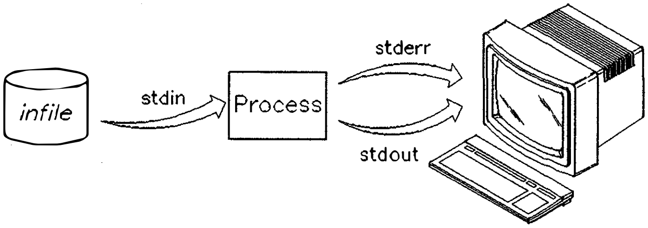


**Figure 7-3.
Standard Input, Output, and Error
When Input Is Redirected**

Redirecting Both Standard Input and Standard Output

You can redirect both the standard input and the standard output of a single command. However, do *not* use the same file name for standard input and standard output. When input and output operations use the same file, the original contents of the input file are lost.

Using the Default Standard Input and Standard Output

The following example uses the **sort** command to sort text typed at the keyboard. Typing CTRL-D ends standard input. The standard output displays on the terminal screen as follows:

```
$ sort
muffy
happy
bumpy
CTRL-D                              End of standard input.
bumpy
happy
muffy                               End of standard output.
$ _
```

Redirecting Standard Input

In the following example, input is redirected:

```
$ more socks                    Display contents of socks.
polka dot
argyle
plaid
$ sort < socks                  Redirect input from socks
                                and sort the contents.
argyle
plaid
polka dot
$ _
```

In the preceding example, the **sort** command uses a file named **socks** as input. As with the first example, the standard output displays on the terminal screen.

Using Both Standard Input and Standard Output Redirection

The next example combines both input and output redirection:

```
$ sort < socks > sortsocks   Use both input and output redirection.
$ more sortsocks             Display contents of sortsocks.
argyle
plaid
polka dot
$ _
```

In this example, the **sort** command reads input from the **socks** file and writes output to the **sortsocks** file; thus, standard output (unlike the first two examples) does not display on your screen.

Figure 7-4 illustrates where **stdin**, **stdout**, and **stderr** are directed when both output and input are redirected from and to files.

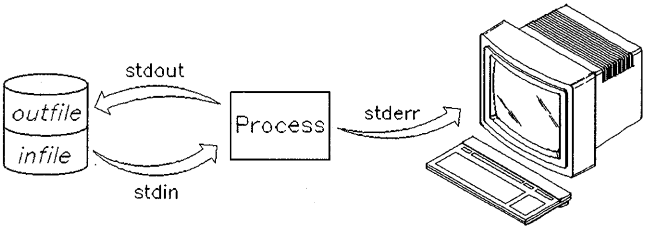


Figure 7-4. Redirecting Both Input and Output

Piping Command Output and Input

The shell lets you connect two or more processes, so the standard output of one process is used as the standard input to another process. The connection that joins the processes is a **pipe**. To pipe the output of one process into another, you separate the commands with a vertical bar (|).

In previous sections, you were shown how standard output and standard input are redirected to and from files. Using a pipe, output from one command is sent directly to another command as input. A pipe can link any two programs, provided the first program writes its output to `stdout` and the second program reads its input from `stdin`. The general syntax for a pipe is as follows:

command1 | *command2*

where *command1* is the command whose standard output is redirected or piped to another command, and *command2* is the command whose standard input reads the previous command's output. You can combine two or more commands into a single pipeline. Each successive command has its output piped as input into the next command on the command line:

command1 | *command2* | ... | *commandN*

You can use pipes whenever you want one command to use the output of another command. In the following example, output from the `who` command is again stored in the file `savewho`. Then, the `savewho` file is used as input to the `wc` command:

```
$  who > savewho    Redirect output of who to file savewho.
$ wc -l < savewho   File savewho is input to wc command.
    4               Sample result.
```

With a pipeline, these two commands become one:

```
$  who | wc -l
    4
```

As this example illustrates, using pipes eliminates the need for temporary intermediate files. Instead, the standard output from the first command is sent directly to the second command as its standard input.

Using the tee Command with Pipes

The tee command lets you divert a copy of the data passing between commands to a file without changing how the pipeline functions. The example below uses the who command to determine who is on the system. In the example, which is further illustrated in Figure 7-5, the output from who is piped into the tee command, which saves a copy of the output in the file savewho, and passes the unchanged output to the wc command:

```
$ who | tee savewho | wc -l
      4
$ more savewho
pat          console      Oct  9 08:50
terry        tty01        Oct  9 11:57
kim          tty02        Oct  9 08:13
kelly        tty04        Oct  9 10:04
$ _
```

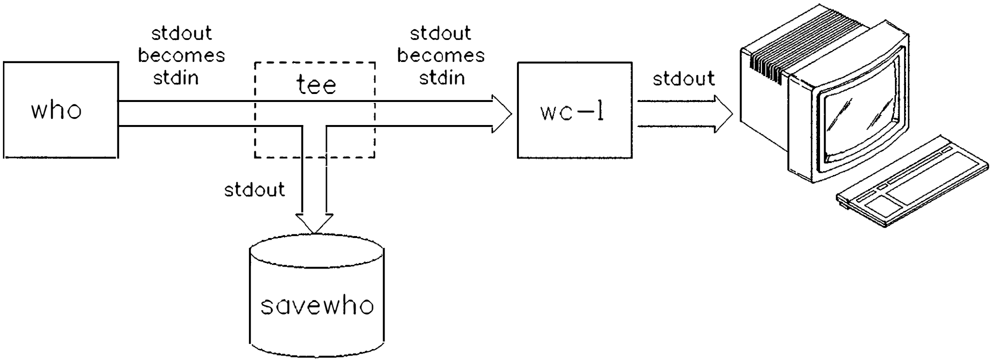


Figure 7-5. Standard Input and Output with Pipes and tee Command

Chapter Command Summary

Table 7-1.

To Do This ...	**Type This ...**
Find out what processes are running and their PID's	`ps -ef`
Stop a process (whose i.d. is *PIDnum*)	`kill` *PIDnum*
Stop an unresponsive process (whose i.d. is *PIDnum*)	`kill -9` *PIDnum*
Redirect (save) standard output to a file	*command* > *outfile*
Append standard output on a file	*command* >> *outfile*
Redirect input from a file to a command	*command* < *infile*
Redirect both standard input and output to a file	*command* < *infile* > *outfile*
Connect ("pipe" between) two process	*command1* | *command2*
Simultaneously file command output and send to another command ("tee")	*command1* | `tee` *file* | *command2*

8

Using Text Editors

Chapter Contents

Overview of Editors

If Your are Running HP VUE:

Your HP VUE editor is Text Editor. You can activate Text Editor through the Tool Box or by clicking on the editor button (an icon which looks like a pad and a pencil) on the HP VUE Front Panel. The pull-down menus with the Text Editor make it easy to use. For more information, see *HP VUE User's Guide*.

Your can also use the `vi` editor that comes with HP-UX in an HP VUE terminal window. The `vi` editor is the default for the Mailer, command line editing and for the action for editing your `.vue/vuewmrc` file.

If You are Using Shell Prompts:

Your HP-UX system includes the `vi` screen editor as the default editor. You can also run the optional Emacs editor, the optional WordPerfect, and other text tools. Use this chapter to find out about running `vi`, and to get an introduction to Emacs.

Using vi: Commands and Text Entry

To use `vi`, you will probably also want to refer to the *Ultimate Guide to the vi and ex Text Editors*.

The `vi` "Visual Interactive" editor has two basic modes for manipulating text:

- Command mode
- Text entry mode

When you enter `vi`, you will be in command mode until you enter one of the text entry codes, such as `i` or `a`, which are explained in this section. In text entry mode, you can backspace and type over text you have just entered (by pressing CTRL-h or Back Space). But, if you want to move around otherwise in your text and execute other text-manipulation commands, you will have to press ESC to return to command mode.

If You Make Mistakes in vi

Use the following procedures to correct mistakes:

- If you type an error while entering text, press Back space to back up over the error, and then re-type the correct text.
- If you type several errors and cannot recover, exit `vi` without saving the file, and start over. To do this, press ESC. Then type:

```
:q! Return
```

When you enter commands in `vi`, letter case (caps or small letters) *does* matter. For example, lowercase `i` and uppercase `I` represent two different commands. Therefore, if the cursor doesn't move as it should, make sure the Caps key isn't locked on, or see your system administrator.

Entering Text

Start `vi` by entering the command `vi` *filename* at the prompt. If a file called *filename* exists, you will see the first screen of that file. If the file does not exist, it is created, and you will see a blank screen.

Selecting Editing Functions

The `vi` editor has several functional modes. When you enter `vi` you are in **command mode** and it is in this mode that you select all editing functions. After entering a command and entering text, you return to command mode by pressing ESC. In this mode, your selection of editing commands determines what you can do to the text.

Press ESC to ensure that `vi` is in command mode. Then you can execute any of the following commands (among others):

`i` (the insert command) Puts `vi` in text mode and enters whatever you type preceding the cursor. Everything after the cursor will be moved to the right.

`a` (the append command) Puts `vi` in text mode and allows you to enter text after the current cursor position. The cursor moves to the right, and then text is inserted as with `i`.

`x`(the delete command) Deletes the character that is highlighted by the cursor. This command does not put your document in text mode.

Each command in command mode allows you to perform only that function. For example, if you place your file in text mode by typing `r` ("replace a single character" command), then you may only replace one character. You are then placed in command mode, and you can return to text mode by typing `i` or `a` before inserting text.

Positioning the Cursor

The most commonly-used method to move the cursor is to use the `h`, `j`, `k`, and `l` keys. You can also use the arrow keys. These keys move the cursor as follows (press ESC first for command mode):

Table 8-1.

To Do This ...	Type This Command ...
Move the cursor right.	`l` or ▶
Move the cursor left.	`h` or ◀
Move the cursor up.	`k` or ▲
Move the cursor down.	`j` or ▼

Saving Your Work and Quitting

You can save your work with or without quitting `vi`. Press ESC to ensure that `vi` is in command mode:

Table 8-2.

To Do This ...	Type This Command ...
Save without quitting `vi`	`:w`
Save and quit `vi`	`:wq`
Quit `vi` without saving changes	`:q!`
Save under another file name	`:w` *filename*
Save in an existing file and overwrite that file	`:w!` *filename*

For the procedure for printing your files, see “Viewing and Printing Files” in Chapter 4.

Making Your vi Environment Permanent

To avoid setting options or defining abbreviations or macros each time you enter **vi**, place all options and definitions you normally use into an **.exrc** file. Your system gives you a default **.exrc** file in your home directory which you can modify. Your changes to the **.exrc** file makes your customized **vi** environment permanent—until you decide to change the file again.

To change the default **.exrc** file, follow these steps:

1. Type `cd` at the HP-UX prompt to ensure that you're in your home directory; then use **vi** to create a **.exrc** file:

   ```
   $ cd
   $ vi .exrc
   ```

2. Type the options, word abbreviations, and macros you want to make permanent (don't precede the commands with a colon).
3. Type `:wq` to save the text and exit `vi`.

After creating the **.exrc** file, you can access it whenever you want to change your **vi** environment. Any of the editor options discussed in the previous section can be placed in this file for **vi** to read automatically each time you enter `vi`.

Summary of vi Essentials

You should now know how to enter **vi**, enter text, move the cursor, make corrections, and exit **vi** ... with or without saving the text. The table below summarizes the basic commands.

Table 8-3. vi Essentials

To Do This ...	Type This Command ...
Enter **vi** and create, or use existing, *sample_file*.	vi *sample_file* Return
Insert text before the cursor.	**i**
Exit text-entry mode and return to command mode.	ESC
Write the file and then quit **vi**.	**:wq** Return
Move the cursor right.	**l** or ▶
Move the cursor left.	**h** or ◀
Move the cursor up.	**k** or ▲
Move the cursor down.	**j** or ▼
Append text after the cursor.	**a**
Delete one character.	**x**

Table 8-3. vi Essentials (continued)

To Do This ...	Type This Command ...
Exit `vi` without saving changes.	`:q!` Return
Write (save) the current file.	`:w`
Write the current file to *filename*	`:w` *filename*
Overwrite the contents of *filename* with the current file.	`:w!` *filename*
Write lines *x* through *y* of the current file to *filename.*	`:`*x,y* `w` *filename where x,y represents specific line numbers or place markers.*
Run an HP-UX command while in `vi`	`:!`*command_name*
Print the current file	`:!lp %`
Print a file (see "Viewing and Printing Files" in Chapter 4)	`:!lp` *filename*
Write the current file and quit (exit) `vi`.	`:wq`

For More Information

You have now gone through most of the essentials for doing basic tasks in `vi`. For more information on the many details of this versatile editor, please see *The Ultimate Guide to the vi and ex Text Editors.*

Installing and Using Emacs: Overview

Emacs is an unsupported editor running on HP-UX which provides a number of versatile functions. It can do the following:

- Handle several documents simultaneously.
- Operate in editing modes specific to writing outlines. and for formatting certain programming languages.
- Use custom keyboard macros.
- Spell-check your document.
- Provide facilities for manipulating files and directories on your system.
- Provide a mail handler which lets you compose, send, and receive electronic mail messages without leaving the application.

To install Emacs ...

If you have the Gnu Emacs source code tape from your HP System Engineer (or see the address in the next section), it can be unloaded by entering:

```
tcio -i devicefile | tar xvf -
```

Where *devicefile*, for example, could be `/dev/update.src` on the Series 300 or 400.

Unloading the source tape requires approximately 10 MB of free file space, while compiling GNU Emacs requires another 10 MB. Most of this space is recovered when the installation process is complete. It does not matter what directory the tape is unloaded into, as long as it is on a file system which has enough space. You will need further details for installation, and these are in the documentation accompanying the tape or FTP/UUCP files.

For More Information on Using Emacs

Although Emacs is unsupported by HP, you can get copies at low cost from several sources. GNU Emacs, and related software, is available from:

Free Software Foundation, Inc.
675 Massachusetts Avenue
Cambridge, MA 02139
USA

+1 617-876-3296

Contacting this address will get you information on how to get Emacs source code and documents via UUCP, FTP, or US mail.

You can also get information on using Emacs from *GNU Emacs: UNIX*$^{\text{TM}}$ *Text Editing and Programming*.

9

Sending and Receiving Mail

Chapter Contents

- Getting Started with the Elm Mailer
- Reading Your Mail.
- Sending Mail to Users on Your System.
- Sending Mail to Users on Other Systems.
- Deleting Mail Messages.
- Saving a Mail Message to a File.
- Mailing a Directory and Contents
- Customizing Elm.

Getting Started with the Elm Mailer

If you are on a multi-user system, you can send mail messages to other users on your system using the `elm` mailer. If your system is configured to a network, such as a local area network (LAN), you can send mail messages to users on other systems.

If You are Running HP VUE:

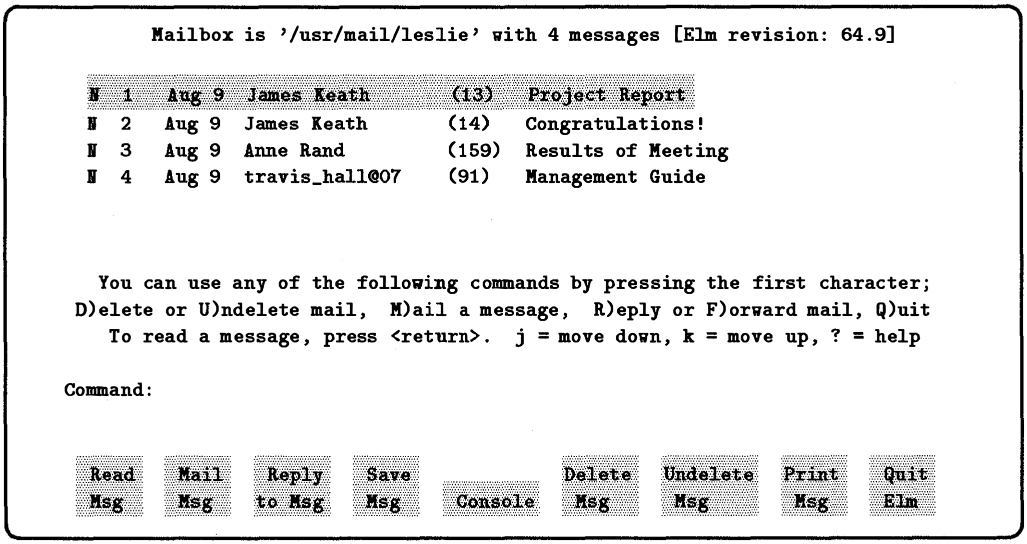
To start a window displaying the `elm` mailer, click on the postal letter icon at the right side of the Front Panel.

If You are Using Shell Prompts:

If you are working from a terminal window, enter at the system prompt:

```
$ elm
```

You will see a display similar to the following:



```
        Mailbox is '/usr/mail/leslie' with 4 messages [Elm revision: 64.9]

     N  1   Aug 9  James Keath      (13)   Project Report
     N  2   Aug 9  James Keath      (14)   Congratulations!
     N  3   Aug 9  Anne Rand        (159)  Results of Meeting
     N  4   Aug 9  travis_hall@07   (91)   Management Guide


    You can use any of the following commands by pressing the first character;
  D)elete or U)ndelete mail,  M)ail a message,  R)eply or F)orward mail, Q)uit
     To read a message, press <return>.  j = move down, k = move up, ? = help

 Command:
```

Figure 9-1.

You can enter an **elm** command in either of two ways:

- Type the first letter (uppercase or lowercase) of the command.
- Press the softkey that corresponds to the command. (You can execute the command menu choices, `Read Msg`, `Mail Msg`, and the like, that appear at the bottom of the screen by pressing softkeys (f1) through (f8) at the top of your keyboard.)

The examples in this guide use the first method: typing the first letter of the command.

Leaving Elm

When you are in **elm**, you can leave it by typing the response to **Command**. If you have mail, **elm** will respond in the following way:

```
      Command: q         Keep mail in incoming mailbox ? (y/n) y
```

You can either respond **y** to the prompt or press Return which also enters a **y** for your response.

Any messages in the incoming mailbox remain there, and the shell prompt returns. If you answer **n** to the prompt, messages are stored in an alternate mailbox, called **mbox**, in your home directory.

Setting Your Elm Environment: the .elm Directory

Like many programs in HP-UX, **elm** will automatically refer to a **~/.elm/elmrc** file for how to configure your version of **elm**. This will happen each time you start **elm**. If your home directory does not have the **.elm** directory with the **elmrc** file, you should create them:

```
$ mkdir .elm
$ cd .elm
$ vi elmrc
```

The elmrc file can contain different kinds of "customization" information. For example, it will tell elm how to sort your message lists: in alphabetical order, by the order in which they have come in, or in some other order. It can tell elm which type of editor you would like to use (if other than the one determined in your .profile). The section, "Customizing Elm," will give you more details on how elm will set up your elmrc file for you.

While you're in the .elm directory, you should also create a directory to store the names of users you commonly send mail to. These lists can then be identified and referred to by elm as one-word "aliases." The use of aliases makes group mailings much simpler.

For example, the following alias entry will enable you to mail to five different people with one "name" (some include an address for other systems):

```
call =  anne, pat, michael, don@ulpcu3, paul@ulpcu3,
```

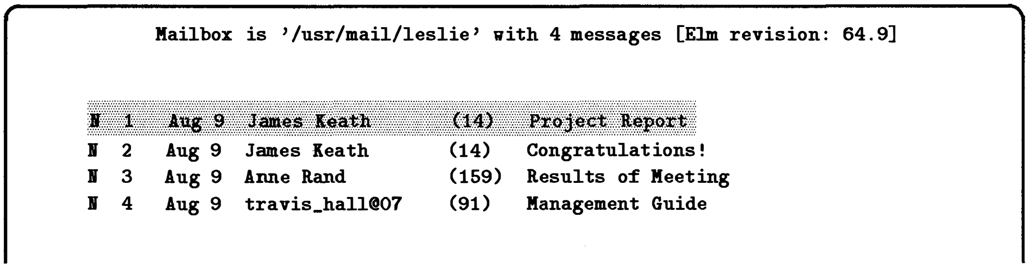
Reading Your Mail

To read your mail, start the `elm` mailer. If you *do* have mail, `elm` displays a list of mail messages. You can read the *current* message or pick a specific message to read.

To determine whether you have any mail, type:

```
$ elm
```

The `elm` screen appears. If you have messages, `elm` lists the message information in a display similar to the following:



```
          Mailbox is '/usr/mail/leslie' with 4 messages [Elm revision: 64.9]


   N  1   Aug 9  James Keath       (14)   Project Report
   N  2   Aug 9  James Keath       (14)   Congratulations!
   N  3   Aug 9  Anne Rand         (159)  Results of Meeting
   N  4   Aug 9  travis_hall@07    (91)   Management Guide
```

Figure 9-2.

To read the *current* message (that is, the message highlighted in inverse video) press Return. (On some systems the current message may be indicated by a > to the left of the message.)

Caution Problems with system behavior may result if you attempt to use one of the other HP-UX mailers, such as `mailx`, while you are also working in `elm`.

The following example shows the output from reading message 1.

```
Message 1/4 from James Keath

Subject:  Project Report
To: leslie
Date: Mon, 26 Nov 90 16:26:45 MDT
Cc:

The project report is in the mail.
Hope you enjoy it!

Best regards,

Jim
```

To return to the `elm` main screen, press Return.

Moving Through the Header Information

You can select a message to read as follows:

- Type `j` to advance to the next message, `k` to move to the previous one, just as you would move the cursor in `vi`. Press Return to read the message which you select.
- To jump to a specific message in the header, type the number of the message and press Return.

Displaying Message Headers

Only ten message headers appear on the screen at one time. If you have more than ten messages you can display them as follows:

- To see the next page of message headers, press `+`.
- To see the previous page, press `-`.
- To move to the first message in the list, press `=`.
- To move to the last message in the list, press `*`.

Sending Mail to Users on Your System

One of the easiest ways to learn how to send a mail message is to send one to yourself. If you're not already in `elm`, enter:

```
$ elm
```

To mail a message, type the response to `Command:`

```
Command: m
```

`elm` responds with a prompt requesting the *mail address* of the recipient.

```
Send the message to: leslie
```

Enter your own user name.

`elm` then responds with a *subject line* prompt:

```
Subject of message:
```

Type the subject line for the message.

For the message we're about to type, you might enter a subject line like:

```
Subject of Message: Important Message to Myself
```

After entering the subject line, press Return. `elm` responds with a prompt for the carbon copies:

```
Copies To:
```

Since you're sending a message to yourself, you don't want to send any additional copies so press Return.

On most systems, `elm` brings up the `vi` editor. (Your system can also be configured to bring up a different editor).

If you're in `vi`, press `i` to enter insert mode, and then begin typing the message. At the end of each line, press Return.

```
This is a mail message sent to myself.
Does it work?
We'll soon see.

Goodbye,
Leslie
```

To exit the `vi` editor and save your message, press esc and then :

```
:wq
```

For more information on vi, see the *Ultimate Guide to the vi and ex Text Editors*.

After you exit the editor, you'll see the following message on the screen.

```
Please choose one of the following options by the first character:
E)dit message, edit H)eaders, S)end it, or F)orget it. s
```

To mail your message, type:

```
s
```

After you've sent the mail message, the elm main screen reappears and displays the message "Mail Sent." It might take a few minutes for the system to deliver the message.

Sending a Message to Multiple Recipients

To send a message to multiple users, specify each user's name next to the elm prompt for a recipient.

```
Send the message to:  mike leslie tj
```

Sending Mail to Users on Other Systems

If your system is connected to other systems over a LAN (local area network), UUCP or other networking facility that supports `elm`, you can send mail to users on other systems.

Node Names

Every system connected over a network has a unique **node name** (also known as a **host name**). When sending mail to each other, systems must know each others' node names. Figure 9-3 shows an example LAN with four systems connected. The node names are `research`, `develop`, `market`, and `sell`.

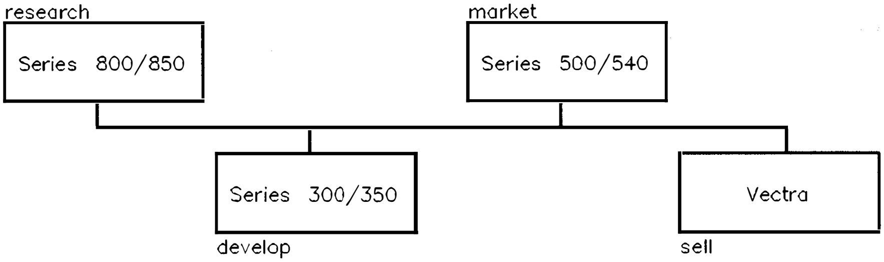


Figure 9-3. Sample LAN and Node Names

To determine your system's node name, use the `hostname` command.

```
$hostname
market        Your hostname is displayed.
```

Mail Syntax when Mailing to Other Systems

The general syntax used when mailing to a user on another system is either:

node!user   *UUCP format*

or

user@node   *ARPA/Berkeley format*

In the above example, *node* is the node name of the system the person is on, and *user* is the person's unique user-name. Which syntax you use depends on whether your system uses UUCP or ARPA/Berkeley network services. Your system administrator will be able to tell you which syntax to use on your system.

Some Sample Mail Addresses

Suppose you want to send mail to user `john`, who uses the system named `sell`. Then, in response to the `elm` prompt, you would use one of the following mail addresses:

```
sell!john
```

or

```
john@sell
```

To send mail to *arnie* on your system, *john* on the `sell` system, and *leopold* on the `research` system, use one of the following addresses:

```
arnie  sell!john  research!leopold
```

or

```
arnie  john@sell  leopold@research
```

Deleting Mail Messages

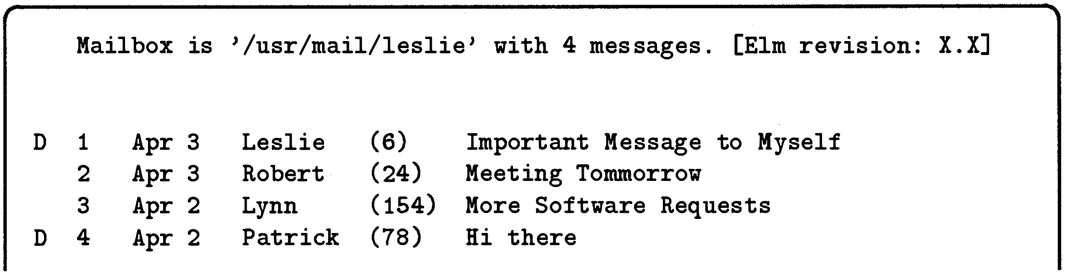
After you've read your mail messages, you may want to delete them. You can mark messages for deletion using the `d` command. When you execute either the `$` command, or if you quit `elm`, you will delete the marked messages.

Marking Messages to be Deleted

To delete a mail message, press `d` while the message is the current message. A `D` appears to the left of the message to show that it is marked for deletion.

You can mark additional messages for deletion by making each message the current message and then pressing `D`. *Remember, to move to the next message, press* `j`*, to move to the previous message, press* `k`*. You can also use the arrow keys,* ▼, ▲.

The following screen shows two messages marked for deletion.



```
   Mailbox is '/usr/mail/leslie' with 4 messages. [Elm revision: X.X]


D  1   Apr 3   Leslie   (6)    Important Message to Myself
   2   Apr 3   Robert   (24)   Meeting Tommorrow
   3   Apr 2   Lynn     (154)  More Software Requests
D  4   Apr 2   Patrick  (78)   Hi there
```

Delete the marked messages at the **command:** prompt in one of two ways:

- To delete the marked messages *and* quit **elm**, enter **q**. **elm** will ask you to confirm this action.
- To delete the marked messages *without* quitting elm, type **$**.

The next section tells you how you can store your messages in a different mailbox or save them to a file.

Saving a Mail Message to a File

When you `quit elm` you have the option of keeping your messages in the incoming mailbox (where your new messages arrive), or storing them in another mailbox (the default is *homedirectory*/`mbox`). While in `elm` you can also save messages to a designated file.

To save the current message to a file, at the `elm` command prompt, type:

```
Command: s
```

The following prompt appears;

```
Command: Save Message
File message in: =/username
```

If you press Return, the message is saved in a file named with the sender's *username* in the `Mail` directory in your home directory. The default mail option is set so that the equal sign (=) is shorthand for *homedirectory*/`Mail`. If the `Mail` directory doesn't already exist, you need to create it.

If you want to save the message in another file, enter the name of the file. For example:

```
Command: Save Message
File message in:=/oldnews
```

Supposing that Leslie is the user, the current message is saved in the file `oldnews` in the `/user/leslie/Mail` directory. If the file already exists, the message will be appended to the contents of the file. If there is no existing `oldnews` file, one will be created.

After you save a message in a file, the message is marked with a `D` for deletion.

Mailing a Directory and Contents

The **shar** utility bundles named files and directories into a single distribution package suitable for mailing or moving. The files can contain any data, including executable programs (which are not ordinarily mailable. The resulting package, written to standard output, is a shell script file that can be edited, for example, to add a message at the beginning.

For example, to archive all files under your home directory, type:

```
cd; shar .
```

To unpack a **shar** package, use the **sh** command with the package name as an argument as follows:

```
sh package
```

When unpacking, the files and directories in package are written to the path names recorded in the archive, as an image of the original directory structure.

shar protects the contained files from mail processing, if necessary, by inserting an @ character at the beginning of each line. If the file contains unusual data, the data is transformed into uuencode format, and a uudecode script is included in package so that the package can still be unpacked correctly by *sh*(1).

The **-b** option allows you to pack files from many directories and unpack them into one directory. The original path-names are ignored. Note that, if the **-b** option is specified, **shar** will not archive sub-directories automatically.

Customizing Elm

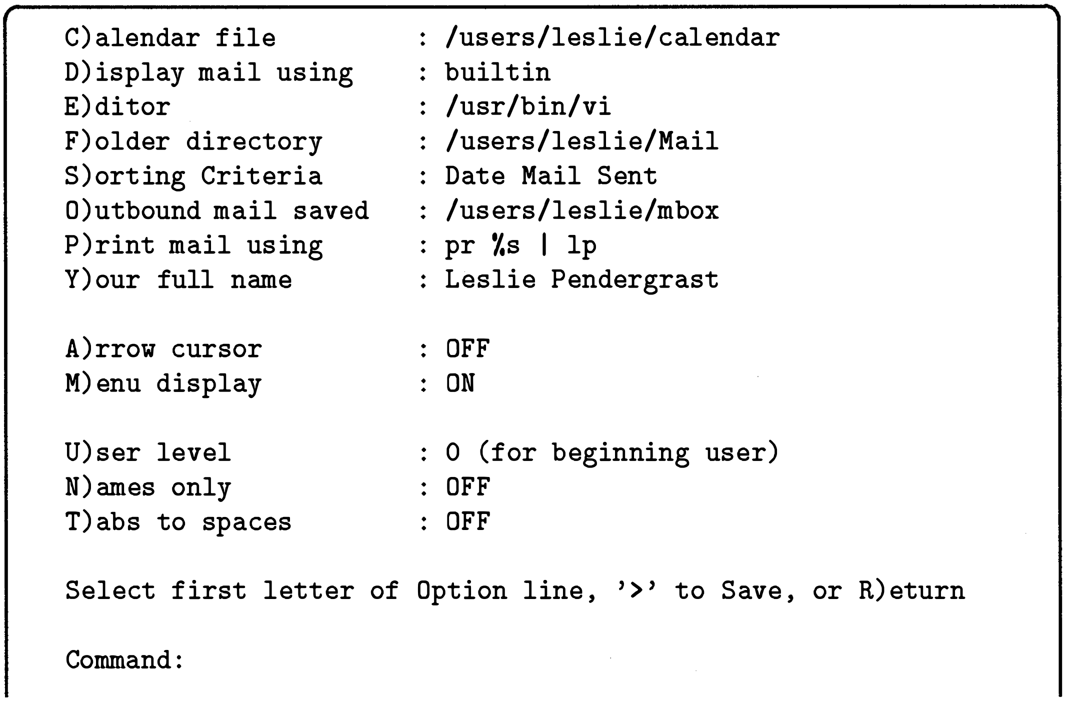
The elm mailer has different options you can set to make it more convenient for you to use. Among features you can change are the menus that appear on the screen, the printer your mail is sent to, and the order in which your mail is listed in your mailbox. These are entered automatically in the .elm/elmrc file which you set up previously in your home directory.

Bringing Up the Option Menu

To bring up the option menu, press o at the elm command prompt:

```
Command: o
```

You'll see a menu similar to the following:



```
C)alendar file          : /users/leslie/calendar
D)isplay mail using     : builtin
E)ditor                 : /usr/bin/vi
F)older directory       : /users/leslie/Mail
S)orting Criteria       : Date Mail Sent
O)utbound mail saved    : /users/leslie/mbox
P)rint mail using       : pr %s | lp
Y)our full name         : Leslie Pendergrast

A)rrow cursor           : OFF
M)enu display           : ON

U)ser level             : 0 (for beginning user)
N)ames only             : OFF
T)abs to spaces         : OFF

Select first letter of Option line, '>' to Save, or R)eturn

Command:
```

Figure 9-4. The Options Menu

This guide does not describe all of the options in the option menu. Rather, it gives you an example of changing an option; in this instance how to change the order in which your mail is listed.

Changing the Order of Your Mail Messages

To change the order in which your mail messages are listed in your mailbox, press `s` (the first letter of `S)orting criteria`) in the alias menu.

```
Command: s
```

You'll see a message indicating how messages are currently sorted. For example:

```
This sort will order most-recently-sent to least-recently-sent
```

To see different choices for sorting your messages, press (space bar). When you see the method you want, press (Return).

For instance, when you see:

```
This sort will order by sender name
```

Press (Return), then press > to save the change.

The change is entered in the `elmrc` file in the `.elm` directory that you have set up previously in your home directory.

To return to your mailbox, press (Return) again.

The messages in your mailbox will now appear in alphabetical order by sender name.

To get information about a specific option in the option menu, type `?` and then type the first letter of the option.

For More Information

For more information on using `shar` and its options, see the *sh*(1) entry in *HP-UX Reference*. You may also be interested in the similar Section One tools, `cpio` and `tar`.

For more information on using and customizing **elm** and other mail systems in HP-UX, using the **options** command and the **elmrc** file see *Mail Systems: User's Guide*.

For specific information on **elm** commands, see the following entries in Section 1 of the *HP-UX Reference*:

elm
readmail
newmail
elmalias
shar

Chapter Command Summary: Elm Commands

Use the following commands from the Options Menu. Other commands are available from the Alias Menu.

Table 9-1.

To do this ...	Use this elm command ...
Delete the messages marked for deletion without quitting elm. ("Resync")	$
Get help on elm commands.	?
Allows you to send a command to the shell without leaving elm.	!
Set up mail aliases.	a
Change the mailbox.	c
Mark messages for deletion.	d
Forward the current message to another user.	f
Send a group reply to all recipients of the original message.	g
Move the message pointer to the next message (below).	j
Move the message pointer to the previous message (above).	k
Send mail to a specified user or users.	m
Allows you to alter the setting of different mail parameters, including the sorting method for messages, the destination of printed messages, the type of menus displayed, and so on.	o

Table 9-1. (continued)

To do this ...	Use this elm command ...
Print messages. (You can change the destination of printed messages using the o command listed above.)	p
Quit elm with the option of changing the contents of the mailbox.	q
Reply to the author of the current message.	r
Save a message to a file.	s
Exit elm without making any changes.	x
Prepare files or a directory structure for mailing	shar *filename*, or, in the desired directory: shar .

To see a summary of all of the commands you can use from elm, type "?" at the elm command prompt.

You can abbreviate every mail command (except help) merely by specifying the first letter. For example, you can abbreviate the delete command with a single d. To abbreviate help, use a question mark (?).

10

Searching and Sorting

Chapter Contents

- Searching for files using `find`.
- Searching for text patterns using `grep`.
- Sorting files using `sort`.

Overview

In addition to the `:/` command for use within `vi`, there are many valuable search and file-manipulation tools in HP-UX and in HP VUE.

If You are Running HP VUE:

See the *HP VUE User's Guide* for doing search and text-related tasks in Text Editor and using the File Manager.

If You are Using Shell Prompts:

Using this chapter will give you all the basic procedures you will need. For detailed information on any of commands discussed in this chapter, see the *HP-UX Reference* entries for `find`, `grep`, and `sort`, or refer to the appropriate man page on-line.

Searching for Files using find

You can use the **find** command to search through a directory and its subdirectories for files meeting certain criteria. You can then execute a command on the files you've found. To display the output of **find** on the screen, you must use the **-print** option.

Finding Files that Match a Pattern

Although the syntax of **find** can be complex, it may help you in using HP-UX in a more productive way, simply by applying the following examples.

Suppose you want to display all files in the current directory and its subdirectories that begin with d. Enter:

```
$ find . -name 'd*' -print
```

The dot (.) causes **find** to search the current directory and its subdirectories. The **-name** option followed by a filename or a filename pattern (in this case **d***) tells **find** to search for all filenames that match that pattern. In this example, **find** will look for all file names beginning with **d**.

Note that:

- **d*** is enclosed by single quotes '**d***'. If you use a file name pattern in the **find** command, you must quote it so that the shell will interpret it correctly. The **-print** option displays the output on the screen.
- The order of **-name 'd*'** and **-print** is important. If **-print** were first, **find** would print *all* names.

Finding Files that are Newer than a Certain File

Suppose you want to display all files modified after a certain file. To display all files newer than **myfile** in the **/users/leslie** directory and its subdirectories, enter:

```
$ find /users/leslie -newer myfile -print
```

This example can be read as follows: find in directory **/users/leslie** and its subdirectories all files modified after **myfile** and display the output on the screen. (To find out the date and time a file was last modifi.ed, use the ll command.)

Running Commands on Files

You can execute commands on files located with the find command. Let's say you want to remove all files with a .tmp extension in the current directory and its subdirectories. Enter:

```
$ find . -name '*.tmp' -print -exec rm {} \;
```

This example finds and displays on the screen all files in the current directory and its subdirectories that end in .tmp, then deletes these files. The -exec option causes the following command (rm) to be executed. The brackets { } represent the files found with the find command. The semi-colon that ends the command string is escaped with a backslash (\;).

Searching for Text Patterns Using grep

You can use the grep command to search for a text pattern within a file or to display the names of files that contain a specified text pattern. This command is useful when you want to search for information in files or directories.

Using Regular Expressions in Searches

In using grep and other search commands, you will need to know something about the kinds of text-pattern arguments that HP-UX expects. Regular expressions are a simple pattern-matching language used by most HP-UX text-processing tools for locating desired text patterns in a file. They can be used to locate a misspelled word, to find all five-letter words in a file that begin with T or t, to locate lines in a file that contain a certain pattern of characters (or a given word) followed by an arbitrary string of text that is followed, in turn, by another specific pattern of text characters, or to find almost any other imaginable combination.

Editors (such as vi, ex, ed, and sed), text processors (such as awk and grep) use regular expressions to search text files for any character patterns that match the possible character sequences defined by the regular expression.

For example, suppose a file contains the word cjt, which happens to be a typographical error that should have been cat. Searching the file for a character sequence that matches the regular expression cjt quickly locates the misspelled word, and a simple substitution of ca for cj solves the problem.

Regular expressions provide an easy method for describing any type of character sequence or pattern so that it can be correctly identified by the search command or text editor.

Regular expressions are used by the following commands, among others:

- vi and its related editors edit, ex.
- sed, the streaming editor, which is used to edit (non-interactively) files according to a script of commands.
- grep and fgrep.
- more.
- awk, a language for scanning and processing text patterns.

Commands such as `grep` search entire the files in an entire directory for a regular expression.

Constructing Regular Expressions

The simplest form of regular expression is a series of common typing characters that restrict matching to an identical character in an identical sequence in the file being searched. Thus, `cjt` in a regular expression matches `cjt` occurring anywhere in the specified region in the file. However, there are other times when text can be classified into general patterns that do not have identical contents. For example, the following series of lines results from execution of an `ls -l` command:

```
drwxrwxrwx   2 hank     projA        1024 Dec 29  1987 proj_mail
drwxr-x---   2 hank     projA        1024 Oct 25  1987 proj_status
-rwxr-----   2 hank     projA        1024 Jan 24  1988 do_today
drwxr-x---   2 hank     projA        1024 Oct 20  1987 master_files
drwxrwxrwx   2 hank     projA        1024 Dec 20  1987 prod_input
drwx------   2 hank     projA          64 Nov 15  1987 personal
```

Suppose you needed to modify *only the lines that describe directories*, that is, the lines which have a `d` in the first column. Suppose also that you wanted to restrict the listing to *directories that included write permission for users outside of the group named* `projA`. Then the task would become somewhat more complex, but it can be handled by an expression described below.

To identify a character at the beginning of a line, it must be preceded by a circumflex character (^) like this:

```
^d
```

This expression tells the search to look for the letter `d` at the beginning of each line (indicated by the circumflex character) in the text being searched. This expression looks at only the first visible character on the line.

To find the directories that have write permission enabled for users outside of the group named `projA`, the letter `w` must be present in column 9. Since we do not know or care what other permissions are set for the directory, we can look at column 1 to find the directory (must match `d` as before) and at column 9 for a `w`. We can use the period character to represent arbitrary text for other characters in columns 2 through 8:

```
^d.......w
```

Using Beginning- and End-of-Line Anchors in Regular Expressions

To locate the word `The` when it is the first (or the last) word on a line, you might use the two characters which are reserved for use in regular expressions to represent the beginning and the end of a line, the circumflex (`^`) and the dollar sign (`$`), respectively. When constructing a regular expression, the `^` or `$` is typed as a single-character expression just as any normal character, except that it must be the first or last character, respectively, in the expression. If they appear elsewhere in the regular expression, they are interpreted literally as ordinary typing characters. (Thus, the expression `$The` causes a search routine to search for a four-character sequence consisting of the four visible characters `$The` followed by any arbitrary combination of characters and/or end-of-line.)

For example, to find the three letters `The` at the beginning of a line, the correct regular expression would be `^The`. To locate the same word at the end of a line, you would use the expression `The$`.

Excluding Characters from a Set

You can also specify that any character is to be accepted as a match *except* those specified. You can do this by starting the series with a circumflex and placing the series between square brackets. For example:

```
[^aslm]
```

in a given position tells the matching routines to accept any character in the position represented by this single-character expression except `a`, `s`, `l`, and `m`.

For More Information

To learn more details about using regular expressions, see *The Ultimate Guide to the vi and ex Text Editors* and *Shells: User's Guide.*

Overview of the grep Command

The grep command ("global regular expression print") looks at each line of one or more files for a text string that matches a specified pattern. When it finds a matching text string, it displays the line in which the matching string is found.

Searching a File for a Text String

Suppose you have a mailing list called mailist with the contents shown below:

```
Smith, Joe       2345 Pine St.     Santa Clara, CA
Walsen, Stacy    493  Winkle Ave.  San Jose, CA
Diaz, Robert     6789 Pine St.     Santa Clara, CA
Wang, Michael    1832  Jackson St. Santa Clara, CA
```

If you want to extract the addresses of all the people on Pine St. Enter:

```
$ grep Pine mailist
```

The grep command lists all lines in mailist that contain the string Pine. The output is:

```
Smith, Joe       2345 Pine St.     Santa Clara, CA
Diaz, Robert     6789 Pine St.     Santa Clara, CA
```

Searching Multiple Files

The grep command can be useful in other ways. Sometimes, you want to find information, but you don't know or can't remember in which file it's located.

Suppose you have three mailing lists, and can't remember which contains Stacey Walsen's address, enter:

```
$ grep 'Walsen, Stacey' mailist mailist2 mailist3
mailist: Walsen, Stacy    493  Winkle Ave.  San Jose, CA
```

The grep command displays the line containing Stacey's address and the file in which it was found. Note that because it contains a space, the string must be surrounded by single quotes ('Walsen, Stacey'). If grep had found other instances of the specified text string, it would list each instance that it found.

If you wanted to search the entire current directory for this information, you could simply enter:

```
$ grep 'Walsen, Stacey' *
```

Ordering Files Using sort

You can use the sort command to order the contents of files, sorting alphabetically, numerically, or by different fields.

Overview

The sort command displays sorted file contents on a line-by-line basis. It compares the first characters in each line; if they are the same, it compares the next two characters, and so on through the end of each line.

The sort command also recognizes different sortable fields. Fields are separated from each other by spaces or tabs. If there is more than one leading space before a field, sort will count each space as a sortable character.

Displaying Sorted File Contents in Alphabetical Order

Here is an example file called list, containing the names and telephones numbers of the members of a tennis club and the number of tickets they have sold for a benefit match. Each line in the file has four fields: the first name, the last name, the number of tickets sold, and the phone number. The fields are separated by spaces.

If you want to practice using the sort command, type in the list file using the cat command as shown in the following list:

```
$ cat > list
Nancy Smith     4    467-2345
Jeff Bettleman  8    438-7689
Jeff Plimpton   13   729-8965
Joyce Smith     6    245-1342
```

To display the sorted contents of the file, enter:

```
$ sort list
Jeff Bettleman  8    438-7689
Jeff Plimpton   13   729-8965
Joyce Smith     6    245-1342
Nancy Smith     4    467-2345
```

Note that the list is sorted alphabetically by *first* names. The sort command without any options will sort starting with the first field in each line (in this example the first name).

The sort command places the two "Jeffs" in the correct order, because after comparing the first names and finding them identical, it compares the second fields (last names).

Sorting Files by Different Fields

The next example shows how you can use sort to order the list by last names.

```
$ sort +1 list
Jeff Bettleman  8    438-7689
Jeff Plimpton   13   729-8965
Nancy Smith     4    467-2345
Joyce Smith     6    245-1342
```

The +1 option causes sort to skip the first field of each line and start sorting at the second field (the last names). However, note that the first names Nancy and Joyce are not in alphabetical order.

The next example shows how to sort by last name and then by first name.

```
$ sort +1 -2 +0 list
Jeff Bettleman  8    438-7689
Jeff Plimpton   13   729-8965
Joyce Smith     6    245-1342
Nancy Smith     4    467-2345
```

The +1 causes sort to skip the first field (first name) and sort by the second field (last name). The -2 causes the sort command to stop after comparing the second fields. If any of the second fields are identical (in this case the Smiths), the +0 instructs sort to make a second pass and sort from the beginning of the line (putting the first names in correct order).

Note that the sort command will only make a second pass at comparing lines if its first pass shows all of the examined fields in the lines to be identical. If -2 were not specified in the above example, sort would show the third fields (telephone numbers) to differ and would never make a second pass to compare the first names.

Sorting in Numerical Order

Suppose you want to sort the file by the number of tickets sold. Enter:

```
$ sort -rnb +2 -3 list
Jeff Plimpton    13  729-8965
Jeff Bettleman   8   438-7689
Joyce Smith      6   245-1342
Nancy Smith      4   467-2345
```

There are several things to keep in mind when sorting in numerical order. First, `sort` will consider multiple leading blanks in its comparison. Since there are a variable numbers of blanks preceding the third field (ticket number), your results will be incorrect unless you use the `-b` (ignore leading blanks) option.

Also, the numbers will not be in the right order unless you use the `-n` (numeric) and `-r` (reverse) options. The `sort` command by default sorts from least to greatest numbers so you need the `-r` option to list the greatest number first. Also, by default `sort` works by comparing characters in a field in the order they appear. So without the `-n` option, 8 would come before 13, because 8 is larger than 1.

Chapter Command Summary

Table 10-1. General Commands

To do this ...	Type this ...
Sort the contents of a file and display the results to standard output	`sort` *options filename*
Find all filenames in the current directory and subdirectories beginning with "x"	`find . -name 'x*' -print`
Find the word "Jim" in any file in the current directory	`grep Jim *`
Find the words "Joe Smith" in any file in the current directory	`grep 'Joe Smith' *`

Table 10-2. Common Options for the Sort Command

`sort` Option	Does this ...
`-b`	(disregard leading blanks) Blanks, which can be either spaces or tabs, delimit sortable fields. Without the `-b` option, `sort` considers all blanks preceding a field to be part of that field, and will use them in sort comparisons.
`-d`	(dictionary) disregards all characters that are not alphanumeric or blanks. In particular, this option disregards control characters or punctuation.
`-f`	(fold) doesn't consider any difference between upper and lower case.
`-n`	(numerical) sorts in numerical order. Plus and minus signs are considered as plus and minus signs, and dots (.) are considered as decimal points.
`-r`	(reverse) reverses sort order (i.e., z-a).
`-u`	(unique) removes duplicate lines from a sorted file.

11

Using Your Shell Environment

Chapter Contents

- Shell Features: Determining and Changing Your Shell.
- Editing the Command Line.
- Recalling Previous Commands.
- Setting the Login Environment.
- Using Login Scripts to Set the System Environment.
- Setting and Referencing Variables.
- Finding Commands with Search Paths.
- Setting Terminal Characteristics.

Shell Features: Determining and Changing Your Shell

HP-UX gives you your choice of several different shell-types which you can run. This chapter discusses the Bourne, Posix, Korn, and Key Shells. Details on the C shell can be found in the *Shells: User's Guide*.

Each of these shells has different characteristics, and you can increase the speed and efficiency with which you interact with HP-UX if you learn to use some of the built-in features of the shell of your choice.

With the Posix, Korn, and Key Shells, you can edit your command line and recall, and re-enter, previous commands. Your shell environment can be "customized" using shell variables and login scripts. One of the variables, PATH, enables the shell to find the correct directories for commands and files.

Using simple commands, you can determine which shell you are running, temporarily change your shell, or permanently change your shell. Table 11-1 compares the features of the Bourne, Posix, Korn, Key, and C Shells. Refer to Table 11-2 for a listing of both the file name for each shell and the default system prompt.

Note When you first log in as `root`, in HP VUE or otherwise, you are in a Bourne Shell (`/bin/sh`). The default shell that SAM presents on the User Account screen is also the Bourne Shell. You can, of course change these shell selections, as described in this chapter.

Below are listed some of the features which may help you make a decision on which shell would be best for the kind of work you are doing:

Table 11-1. Comparison of Shell Features

Features	Description	Bourne	Posix Korn Key	C
Command history	A feature allowing commands to be stored in a buffer, then modified and reused.	No	Yes	Yes
Line editing	The ability to modify the current or previous command lines with a text editor.	No	Yes	No
File name completion	The ability to automatically finish typing file names in command lines.	No	Yes	Yes
`alias` command	A feature allowing users to rename commands, automatically include command options, or abbreviate long command lines.	No	Yes	Yes
Restricted shells	A security feature providing a controlled environment with limited capabilities.	Yes	Yes *(Not Posix Shell)*	No
Job control	Tools for tracking and accessing processes that run in the background. *See Shells: User's Guide*	No	Yes	Yes

Determining Your Login Shell

As you found in "Logging In the First Time" in Chapter 2, the command `echo $SHELL` displays the file name of the shell you entered when you logged in.

```
$ echo $SHELL
/bin/posix/sh
$ _
```

The echo command displays the contents or value of a variable named SHELL. The SHELL variable contains the name of the file that contains the shell program that you are running. In this example, it is /bin/posix/sh, the file that contains the code for the Posix Shell.

Table 11-2 lists both the file name that displays for each shell and the default system prompt. (The root prompt for each is #.)

Table 11-2. Shell File Names and Default Prompts

Shell	File Name	Prompt
Bourne	/bin/sh	$
Posix	/bin/posix/sh	$
Korn	/bin/ksh	$
Key	/usr/bin/keysh	$
C	/bin/csh	%
Restricted Bourne	/bin/rsh	$
Restricted Korn	/bin/rksh	$

Temporarily Changing Your Shell

Unless you are in a restricted shell, you can temporarily change your shell by using this command:

shell_name

where *shell_name* is the name of the shell (for example, **sh**, or **ksh**). Temporarily changing your shell lets you experiment in other shells. By typing the name of the shell you want to run, you *invoke* (enter) that shell, and the correct prompt is displayed. After experimenting in the new shell, return to your original shell by typing either **exit** or CTRL-D.

The following example begins in the Bourne Shell, enters the Korn Shell, and returns to the Bourne Shell:

```
$ ksh                                Enter Korn Shell.
$ ps                                 Execute the ps command.
   PID TTY       TIME COMMAND
  6009 tty01     0:00 ksh            Notice that both the Korn Shell and
  5784 tty01     0:00 sh             Bourne Shell processes are running.
  6010 tty01     0:00 ps
$ exit                               Exit Korn Shell.
$ _                                  Bourne Shell returns.
```

Permanently Changing Your Shell

To permanently change your *login shell* (the default shell you get when you log in), use the **chsh** (**ch***ange* **sh***ell*) command:

```
chsh username full_shell_name
```

where *username* is your user name and *shell_path_name* is the full path name (e.g., **/bin/posix/sh**) of the shell you want as your default. Table 11-2 contains the full path names for each of the shells. After you use the **chsh** command, you must log out and log in again for the change to take effect. For example, if **terry** changes the default login shell to the Korn Shell, the command reads:

```
$ chsh terry /bin/ksh
$ _
```

Editing the Command Line

In the Posix, Korn, and Key Shells, you can correct errors and make changes in a command line before you enter it. You can make these changes with the *line-editing commands* or edit keys.

The built-in line editor in these shells allows you to move the cursor to the point in the line where you wish to make a change. There you can enter the correction, and execute the command. If you type in a line containing an error and execute it, it is also possible to recall that command, edit it and execute it again. (See "Recalling Previous Commands," later in this chapter.)

Using a Line-Editing Command Set

In "Using vi: Commands and Text Entry" in Chapter 8 in this manual, you learned how to use **vi** screen editor with text files. The **vi** editor is also used to edit command lines.

To enter the **vi** line-editor mode while in the Posix, Korn, or Key Shells, press (ESC) to change from the usual "typing mode" into "edit mode." to move the cursor to a new location or to delete characters. You then change back to "typing mode" by entering the **vi** commands **i** or **a** to insert or append text. In Key Shell, you can also use the arrow and editing keys on your terminal.

Note that you can use the line-editing features *only* in the Posix, Korn, or Key Shell. If you are in another shell, change to the Posix or Korn Shell before proceeding further.

```
$ /bin/ksh
```

Table 11-3 contains a partial list of the editing commands available.

Table 11-3. vi Line-editing Commands in the Posix or Korn Shell

Desired action	vi command
Move back one character	h
Move forward one character	l
Move back one word	b
Move forward one word	w
Move to the beginning of the line	^
Move to the end of the line	$
Delete the character under the cursor	x

The editor command set you will use is governed by the setting of the EDITOR variable. Some possibilities are vi or emacs. Setting the EDITOR variable also depends on the VISUAL variable being defined. If you decide to use the vi editor on a temporary basis, set it in this way:

```
$ set -o vi        “set minus letter-o vi”
$ _
```

To turn off the vi editing mode, type:

```
set +o vi
```

To set the EDITOR variable automatically each time you log in, see *Shells: User's Guide.*

Executing vi Line-Editing Commands

To execute the vi editor commands, press ESC (to enter the command mode), then enter vi commands for the desired actions. To return to “typing mode,” press i (to insert text at the current position of the cursor) or a (to append text at the point immediately to the right of the cursor).

An Example of Line Editing with the vi Command Set

Activate the **vi** command set (if it is not already activated at login by your login script):

```
$ set -o vi
$ _
```

Type this line *without* pressing Return:

```
$ ll /dve | grep '^d' | more_
```

The second element should have been **/dev**. Correct the error by following these steps:

1. Press ESC. The cursor moves back one space (beneath the **e** in **more**). The line editor is now in "command mode."

   ```
   $ ll /dve | grep '^d' | more
   ```

2. Press h repeatedly to move the cursor to the beginning of the line.

   ```
   $ ll /dve | grep '^d' | more
   ```

3. Press L five times. The cursor moves beneath the **v** in **/dve**.

   ```
   $ ll /dve | grep '^d' | more
   ```

4. Press X. The character **v** disappears, and the rest of the line shifts one space to the left to fill in. The cursor is now under the **e** in **/de**.

   ```
   $ ll /de | grep '^d' | more
   ```

5. Press A. The cursor moves one space to the right. The line editor is now ready to "append" text to the line.

   ```
   $ ll /de_| grep '^d' | more
   ```

6. Press V. The character **v** is inserted after **/de**, completing the correction.

   ```
   $ ll /dev_| grep '^d' | more
   ```

7. Press Return to execute the command line.

For More Information

The `vi` line-editing command set works in a way that is very similar to the full `vi` editor. See "Using vi: Commands and Text Entry" in Chapter 8 in this manual for an introduction to `vi`.

You can also use Emacs as a line editor. Emacs is optional and is available through your HP user's group or by mail order. See Chapter 8 for the availability of Emacs information an installation tapes.

Recalling Previous Commands

The Posix, Korn, and Key Shells store the commands you execute in a **command history**. You can retrieve these commands, modify them, and re-execute them.

The Posix, Korn, and Key Shells have a built-in "memory" of the command lines you type in. These commands are stored in a special area called the **command history**. For information on the C Shell implementation of command history, see *Shells: User's Guide.*

The Posix or Korn Shell's Command History

First make sure you are in the Posix or Korn Shell. If you are not, type the following:

To get into the Posix Shell:

```
$ /bin/posix/sh
```

To get into the Korn Shell:

```
$ /bin/ksh
$ _
```

The cursor will remain the same for both, although the shell has changed.

Execute some commands, as a test. Then, to re-execute a previous command:

- Make sure you have set `vi` as the command line editor (enter `set -o vi` on the command line for the login session, or make the the appropriate entries in your `.profile` to set and export the `EDITOR` variable.)
- Press ESC.
- Then press k repeatedly to scroll "up" to the previous command that you want.
- Or press j to scroll back "down" through the command history list.
- Once you have found the command you want, you can edit it just as if it were the current command.
- You can then execute whatever is on the command line by pressing Return.

The Korn or Posix Shell "remembers" the last 128 command lines you typed in and can display all or any of them. For example, type in some commands:

```
$ date
Wed Mar 29 10:57:28 MST 1989
$ pwd
/users/terry
$ hostname
hpfcma
$ _
```

Now type in this command:

```
$ history -3

121     date
122     pwd
123     hostname
124     history -3
$ _
```

Notice that the Korn or Posix Shell displays the last three commands (`date`, `pwd`, and `hostname`) *and* the `history -3` command. You can increase the amount of the command history shown by using a larger negative number to follow `history`. For example, this will display the last 100 commands if there are 100 commands in the history:

```
$ history -100
```

If there are fewer than 100 commands in the history, the full contents of the history will be displayed.

The Key Shell will also display command history, with the added option of allowing you to use the terminal arrow and editing keys (instead of `vi`) to scroll through the command history and edit commands. As with Korn or Posix Shell, once you have displayed the command line you want, you can execute it by pressing Return.

The Posix Shell command history mechanism is the same as that of the Korn Shell.

For More Information . . .

For more details on the command history in the Korn or Posix Shells, see the relevant tutorials in the *Shells: User's Guide*. For more information on the Key Shell, see Chapter 6, in this manual.

Briefer presentations are available in the *ksh*, *keysh*, and *csh* entries in the *HP-UX Reference*.

Setting the Login Environment

When you log in, your shell automatically defines a working **environment** for you that is uniquely identified from every other environment on the system. Your environment defines such characteristics as who you are, where you are working, and what processes you are running. Your shell maintains this environment until you log out. The characteristics of your environment are defined by values assigned to environment variables.

Your shell environment is analogous to an office environment. In the office, physical characteristics like lighting and temperature are similar for everyone. But many factors in your office environment are unique to you, such as your routine tasks and your individual workspace. Thus, your work environment is different from that of your co-workers—just as your shell environment is different from theirs.

The login Program

When you log in, HP-UX runs a program named `login`. This program starts your session using data stored in the `/etc/passwd` file, which contains one line for each system user. This file includes your user name, password (in encrypted form), home directory, and the shell to run when you log in. If `/etc/passwd` doesn't specify a shell, the Bourne Shell (`/bin/sh`) is selected.

The `login` program executes when you type your user name at the prompt (`login:`). Its tasks include the following:

- Display the `Password:` prompt (if you have a password).
- Verify your user name and password in the `/etc/passwd` file.
- Assign default or user-defined values to the shell environment.
- Start executing the shell process.

Environment Variables

The shell environment defines how HP-UX interacts with you. The environment's characteristics are defined by **environment variables**, which consist of a name and a value. For example, the directory in which you begin each session is your **home directory**; its environment variable is the variable named `HOME`, and its value is assigned during the login process. Throughout this section, the value of `HOME` is equal to `/users/terry`.

The following table contains some environment variables set during the login process. They are available to all three shells. Note that most of these will already be set in your default `.profile` script.

Table 11-4. Environment Variables Set During the Login Process

Variable	Description	Typical Default Value
`HOME`	Defines the user's home directory; the default directory for the `cd` command (for example, `/users/terry`).	Assigned during login
`LOGNAME`	Contains the user name (for example, `terry`).	*username*
`MAIL`	Determines where the system looks for mail. Set based on the user name (for example, `/usr/mail/terry`).	`/usr/mail/`*username*

`PATH`	Sets the directories through which the system searches to find and execute commands.	`/bin/posix:`
		`/bin:/usr/bin:`
		`/usr/contrib/bin:`
		`/usr/local/bin`

`SHELL`	Determines which shell to run. Set to the last field in the `/etc/passwd` file entry for the user logging in. If this field is not defined, the default value is used.	`/bin/sh`
`TERM`	Specifies the kind of terminal for which output is prepared.	`hp`
`TZ`	Provides the current time zone and difference from Greenwich Mean Time. Set to Mountain Standard Time by default; your system administrator should change the value if you are in another time zone. Set by the script `/etc/profile`.	`MST7MDT`
`EDITOR`	Determines the default editor.	`vi`
`DISPLAY`	Specifies window display host. Use on a remote system to display windows locally.	`DISPLAY=`*local*`:0.0`

Using Login Scripts to Set the System Environment

During the login process, HP-UX prompts you for your user name and password (if applicable) before displaying a shell prompt. HP-UX also notes which shell you've selected to run, starts your shell process, and sets up your environment referring to *login scripts*. A login script is a file that lets you customize your environment.

A login script contains commands that let you define your system environment. When you log in, default values are assigned to environment variables. Login scripts provide an automatic way to change the value of these variables every time you begin a session. If login scripts exist, they are executed by your shell before you get a shell prompt.

Two types of login scripts are used:

- A system script for all users of a particular shell on your system or HP-UX cluster.
- Local login scripts in your own home directory.

Typically, a system administrator maintains the system login scripts. (If there is no one in your group responsible for these tasks, refer to *Managing Clusters of HP 9000 Computers* and to *System Administration Tasks*). These scripts set up a default environment for everyone on that system. The Bourne, Posix, and Korn Shells use a system login script named `/etc/profile`.

Once your account is set up, you maintain the local login scripts in your home directory. The local scripts allow you to set up an environment specific to your needs. The Bourne Shell looks for one script: `.profile`. The Posix and Korn Shells use two login scripts: `.profile` and the one referred to by the `ENV` variable (by convention, it is called `.kshrc`). Default versions of the login scripts are placed in your home directory when your account is set up. Default versions are also in the `/etc` directory. For reference, the default `.profile` script for the Bourne, Posix, and Korn Shells is `/etc/d.profile`. (This is not functional unless it is moved to your home directory, if, for example, your own customized `$HOME/.profile` was accidentally erased.)

Why Use Login Scripts?

Login scripts provide a convenient way to set up the shell environment to suit individual needs. For example, the script can change the value of the search path used to find commands, change the shell prompt, set the terminal type, or simply cause the shell to greet you with a friendly message of your choosing.

Customizing your login script is not required, and the login script your system administrator provides should set up the most critical shell parameters. As you experiment with your shell's capabilities, your desire to be creative and customize the login script will probably increase. You may want to refer to the next chapter, "Customizing Login Scripts," for examples of how to customizing your own login scripts.

A Summary of Login Scripts

Table 11-5 summarizes the login scripts for each shell. All the scripts run when you first log in. For the Korn or Posix Shell, **$ENV** is run for each subshell and this variable may refer to other scripts.

A **subshell** is an entirely new shell that your current shell creates to run a program you have requested. For example, typing **ksh** at any shell prompt creates a Korn Shell subshell.

Table 11-5. System and Local Login Scripts

Shell	Program Name	System Login Script	Local Login Script
Bourne	`/bin/sh`	`/etc/profile`	`$HOME/.profile`
Posix	`/bin/posix/sh`	`/etc/profile`	`$HOME/.profile` `$ENV`
Korn	`/bin/ksh`	`/etc/profile`	`$HOME/.profile` `$ENV`
Key	`/usr/bin/keysh`	`/etc/profile`	`$HOME/.profile` `$ENV`

For information on the C Shell, see *Shells: User's Guide*.

Setting and Referencing Variables

Your shell uses both environment variables and shell variables to define your environment. Your login shell uses **environment variables** and passes them to all processes and subshells that you create. **Shell variables** are known only to your current shell and are not passed to subshells.

The Bourne, Posix, and Korn Shells set variables using an assignment statement and an optional `export` command. In all shells, you refer to the *value* of a variable by placing a dollar sign (`$`) in front of the variable name.

To use variables, you will find out how to:

- **set** (assign a value to) a variable.
- **reference** (refer to) the variable's value.

Assigning Values to Variables

In the Bourne, Posix, Korn, and Key Shells, variables are assigned (or set). They can also be created, if necessary. Both tasks are done with an assignment statement:

name=value

The *name* is the variable name and *value* is the value assigned to the variable. No spaces are allowed between *name* and `=` or between `=` and *value*.

In the following example, which works with Bourne, Posix, or Korn Shells, the shell prompt (`PS1`) is reset so that it reads: "`Ready ==>` ." If `PS1` is a shell variable, the subshell (created by typing `sh`) does not know the new value. If you export `PS1`, the value of `PS1` passes to the subshell:

```
$ PS1="Ready ==> "        Set shell variable PS1.
Ready ==> sh              Type in subshell name.
$ exit                    Subshell now has default prompt; exit returns
                          to original shell.
Ready ==> export PS1      Set environment variable with export.
Ready ==> sh              Enter subshell.
Ready ==> _               Subshell knows the new value of PS1.
```

Referencing the Values of Variables (Parameter Substitution)

All three shells use **parameter substitution** for referencing the value of variables. Parameter substitution means that the variable's value is substituted for the variable name. Parameter substitution occurs when a dollar sign (**$**) is placed in front of the variable name.

Earlier in this guide, you learned to determine your login shell with the command `echo $SHELL`:

```
$ echo SHELL     Because $ is omitted, the word SHELL is echoed.
SHELL
$ echo $SHELL    The $ is included, so the value of SHELL is echoed.
/bin/sh
$ _
```

The `echo $SHELL` command uses parameter substitution. The shell substitutes the value of the environment variable named `SHELL` into the `echo` command because the dollar sign (**$**) precedes the variable name.

For More Information ...

To learn more about parameter substitution, refer to *sh*, *ksh*, *keysh*, or *csh* in the *HP-UX Reference* (section 1) or to the *Shells: User's Guide.* For more information on environments in HP VUE, see the *HP VUE User's Guide.*

Finding Commands with Search Paths

When you type a command, HP-UX must be able to find the directory containing the command before it can run the command. The PATH environment variable contains a list of directories you want HP-UX to search when looking for commands. Your PATH should contain all the directories necessary to locate all the commands that you use. If necessary, PATH can be customized, so that non-standard directories are searched.

PATH Variable Format

The PATH variable is read from your .profile or /etc/profile login script. If you look at PATH in one of these files, you will see that it contains a list of directories to search, separated by colons. There should be no spaces surrounding the colons.

You can also use the echo command to determine the current value of PATH, as follows:

```
$ echo $PATH
/bin/posix:/bin:/usr/bin
```

This line means that when you type a command, the shell first searches for the command in the /bin/posix directory, in the /bin directory, and then in the /usr/bin directory. If the command isn't found in any of these directories, the shell displays this message:

```
command_name: Command not found.
```

To determine the current value of your PATH variable, use the echo command as shown above. For example, suppose you run echo and get a response as follows:

```
$ echo $PATH
/bin:/usr/bin:/usr/contrib/bin:/usr/local/bin
```

In this example, the shell searches through /bin, /usr/bin, /usr/contrib/bin, and /usr/local/bin—*in that order*—to find commands. The shell will execute the first instance of the command that it finds along this path.

Changing PATH

If the shell can't find a command that you know exists, you have two options:

1. You can type the full path name of the command. For example, if you wish to execute a command called `prog`, and it resides in the directory `/users/leslie/bin`, type this:

   ```
   /users/leslie/bin/prog
   ```

2. However, you can also change the value of the `PATH` variable to add the command path, `/usr/contrib/bin`. This may offer a better long-term solution if you use the command frequently.

Table 11-6 shows the path names of the most frequently used directories. You might want to add some (or all) of these directories to your `PATH`.

Table 11-6. Possible Directories to Include in PATH

Directory	What It Contains
`/bin`	Frequently used HP-UX commands.
`/etc`	Commands the system administrator uses.
`/usr/bin`	Additional HP-UX commands.
`/bin/posix`	Posix Shell
`/usr/contrib/bin`	Contributed programs not supported by Hewlett-Packard.
`/usr/local/bin`	Programs and commands written locally (at your location).
`$HOME/bin`	A directory you might create for your shell scripts and programs.

Caution Because of the potential security risk, you would not usually put your current directory (represented as . or the equivalent) as the first element in PATH. Leave the current directory out of your `PATH`, or include it only as the *last* element.

Remember that directories in PATH are searched in the order in which they appear (left to right). In general, put the most frequently used directories first in the path—unless two commands in the search path have the same name (for example, /bin/rm and $HOME/bin/rm). In this example, if you want the shell to find your version of rm first, put $HOME/bin before /bin in PATH.

The following example shows how to alter PATH to include $HOME/bin before any other directories, and to include the current directory as the last directory in the search path (this example assumes you're using the Bourne, Posix, Korn, or Key Shell):

```
$ echo $PATH
/bin/posix:/bin:/usr/bin:/usr/contrib/bin:/usr/local/bin

$ PATH=$HOME/bin:$PATH:.    Including . as the last element makes
$ echo $PATH                the current directory the last one searched.

/users/terry/bin:/bin/posix:/bin:/usr/bin:/usr/contrib/bin:/usr/local/bin:.
$ _
```

Setting PATH as an Environment Variable

Normally, you set PATH as a environment variable, so it is set to the appropriate value when you log in. In the Bourne, Posix, and Korn Shells, you can change PATH in the .profile script and export it. You can find out more about these scripts in *Shells: User's Guide.*

Setting Terminal Characteristics

For most effective use of your terminal, HP-UX must know the type of terminal or graphics display you're using. If no terminal type is provided, the default value is TERM=hp. The tset command sets terminal characteristics.

The default local login script prompts you to enter your terminal type as follows:

```
TERM = (hp)
```

Pressing Return sets the TERM environment variable to hp, the default value. This value works with Hewlett-Packard terminals, but it may not let you take full advantage of your terminal or graphics display features. Entering a different value sets the TERM environment variable to that value.

Selecting a Value for the TERM Variable

HP-UX supports many terminal types. The /usr/lib/terminfo database tells HP-UX how to communicate with each terminal type. When you assign a value to TERM, the value must equal a value in the terminfo database.

For example, the files listed under /usr/lib/terminfo/2 show all acceptable TERM values that begin with 2 (this is an example only and shows only a partial listing):

```
$ ls /usr/lib/terminfo/2
2382        2397a       2621a       2623p       2626-x40    2640a
2392        2500        2621k45     2624        2626A       2640b
2392A       2621        2621nl      2624a       2626P       2644
2392a       2621-48     2621nt      2624p       2626a       2645
2393        2621-ba     2621p       2625        2626p       2647
2393A       2621-fl     2621wl      2626        2627        2647F
⋮
```

Table 11-7 outlines the most common terminal and graphics display settings for Hewlett-Packard equipment. When more than one choice is listed, all choices are equivalent.

Table 11-7. Settings for the TERM Environment Variable

If You Are Using a ..	Set TERM to ..
terminal	the terminal's model number; for example 2622, hp2622, 262x, or 2392
Vectra	2392
medium resolution graphics display (512x600 pixels)	3001 or hp3001
high resolution graphics display (1024x768 pixels)	300h or hp300h
HP 98550 display station (1280x1024 pixels)	98550, hp98550, 98550a, or hp98550a
HP 98720 or HP 98721 display station (1280x1024 pixels)	98720, hp98720, 98720a, hp98720a, 98721, hp98721, 98721a, or hp98721a

Setting TERM with the tset Command

The tset command (with the -s option) sets the value of TERM and initializes your terminal characteristics. If you always log in using the same terminal type, you may change your local login script, .profile, to eliminate the TERM prompt. If you look in your .profile, you'll find a line similar to:

```
eval ‘ tset -s -Q -m ’:?hp’ ‘
```

This command displays the TERM prompt. To customize the command, replace ?hp with your terminal type.

For example, the following command initializes your terminal as a high-resolution graphics display (300h), but the TERM prompt itself does not display:

```
eval ‘ tset -s -Q -m ’:300h’ ‘
```

If you use more than one type of terminal (such as one at work and one at home), you can modify your tset command to include multiple terminal types. See *tset*(1) in the *HP-UX Reference* for more information.

Chapter Command Summary

Table 11-8. Commands

To Do This ...	Type This ...
Determine what shell you're in	echo $SHELL
Temporarily change to Bourne Shell	/bin/sh
Temporarily change to Posix Shell	/bin/posix/sh
Temporarily change to Korn Shell	ksh
Temporarily change to C Shell	csh
Permanently change to another shell	chsh *username shell_path_name* (then log out and log in again)
Set command-line editor	set -o *editor_name*
Edit your command line (once editor is set)	Press ESC; use **vi** commands to move cursor and enter text
Recall a previous command line	In **vi** mode, press ESC; press **k** (backwards) or **j** (forward) to move through command history file
Execute a previous command line	Press Return when desired command line is displayed
Set a variable value	*VARIABLE_NAME*= *variable_value*
Enter a subshell	sh
Display PATH setting	echo $PATH
Set terminal parameters	tset *options term_type*

12

System Housekeeping

Chapter Contents

- Managing Disk File Space Usage
- Doing Basic Tasks with the System Administration Manager (SAM).
- Running a Command at a Specified Time with Crontab
- Getting Information on Printers
- Backing Up Your System and Software.
- Restoring Individual Files
- Restoring Your File System.
- Enabling and Disabling HP VUE
- Updating from a Network Server
- Shutting Down Your System.

Managing Disk File Space Usage

One of the most important things you will want to monitor on your workstation is the amount of disk space your files consume. Several tools will help you do this.

The following are some of the file space management tools that can be accessed from a shell prompt.

Displaying Disk Usage: du

The du command gives the number of 512-byte blocks allocated for all files and directories within the current directory or the directory you specify as an argument. By dividing by two the number that du returns, you will get approximately the number of kilobytes in the designated files and directories.

The following are among the more useful options to use with du:

-a Prints entries for each file encountered in the directory hierarchies in addition to the normal output. The default is to give the total for directories only.

-s Prints only the grand total of disk usage for each of the specified file operands.

To display disk usage for files on the root volume ("/") only, with no usage statistics collected for any other mounted file systems:

```
du -x /
```

For More Information

To get more information about your system status, you may also want to see the *HP-UX Reference* for how to use the following tools *df*(1), *bdf*(1), *quot*(1m), and *checklist*(4).

Compressing Files to Save Disk Space: compress and uncompress

In cases where you may need to conserve disk space, you can temporarily reduce the size of little-used files quite significantly by using the **compress** utility. Running the compress command on specific files, or on a group of files by using wildcards, causes each file to be replaced by a compressed file with the extension `.Z`, while keeping the same ownership, modes, access and modification times.

Compressed files can be easily restored to their original form by using `compress -d`, `uncompress`, or `zcat`.

The amount of compression obtained depends on the size of the file(s), the types of characters, and the distribution of common substrings. Typically, text such as source code or English is reduced by 50-60 percent.

Some of the messages you may get from the use of `compress`, `uncompress`, or the equivalents are:

`file: not in compressed format`
The file specified to `uncompress` has not been compressed.

`file: already has .Z suffix -- no change`
The file is assumed to be already compressed. Rename the file and try again.

`file: filename too long to tack on .Z`
The output file name, which is the source file name with a .Z extension, is too long for the file system on which the source file resides. Make the source file name shorter and try again.

`file already exists; do you wish to overwrite (y or n)?`
Respond with `y` if you want the output file to be replaced; n if not.

`-- not a regular file: unchanged`
When the input file is not a regular file (for example, a directory), it is left unaltered.

`-- has` *xx* `other links: unchanged`
The input file has links; it is left unchanged. See *ln*(1) for more information.

12

Examples Using compress

Here are some examples using compress with options. The following command using the -v ("verbose") option compresses the file *zenith* and gives information about the results:

```
compress -v zenith
```

You will see information like the following, (indicating that the compressed file is 23.55% smaller than the original):

```
zenith: Compression: 23.55% -- replaced with zenith.Z
```

To undo the compression, type:

```
uncompress zenith.Z
```

This restores file zenith.Z to its original uncompressed form and name.

Recovering Disk Space

Removing filesets for HP VUE and the X Window System is not recommended unless you are experienced, as there are many cross-dependencies among these and other filesets.

But, if you are not using Instant Ignition and you want to recover disk space, you can easily remove those filesets.

Removing Instant Ignition File Sets

All of the Instant Ignition files copied to your system can be found in /etc/filesets/IGNITION. You can use rmfn to remove these files by typing:

```
rmfn IGNITION (Return)
rmfn IGNITION-HELP (Return)
```

Recovering Disk Space with SAM

You can also use SAM to automatically monitor and remove core files, files that exceed certain limits in size, or files that haven't been modified in a specified length of time. For the general SAM screens to use for this, please see Table 12-1.

Doing Basic Tasks with the System Administration Manager (SAM).

SAM helps you do a wide variety of system administration tasks interactively, working either from a graphical HP VUE screen, or from a character terminal display. SAM helps you find system-wide information easily, and it provides an interface to allow you to more easily enter information for a wide variety of standard HP-UX system administration tools. Moreover, SAM functions in such a way as to minimize the likelihood of making mistakes which would result in lost time and data.

When you invoke SAM, the first screen gives you a selection of functional areas which you can use, one at a time. Selecting one of these areas eventually leads to a display of information, such as a listing of current users or the devices you currently have available for backup and recovery. You can easily stop at this point and just use SAM to get information, without changing anything. You can also select an action which helps you in reconfiguring the system, such as `Add a User`. The SAM `Routine Tasks` subarea will help in trimming and organizing system logfiles, while `Process Management` enables quick access to monitoring tools which you have configured on your system.

The following sections show you a few of the typical tasks you can do in SAM.

Accessing SAM Tasks

A typical task with SAM consists of, at most, the following steps:

1. Highlighting a functional area in the opening screen ("Control Box"), such as `Users and Groups`.
2. Pressing `Open` causes a screen with information on subareas in the selected area to be displayed.
3. Highlighting one of the subareas, such as `Users`.
4. Choosing an object, such as one of the users in the list.
5. Choosing an action, such as `Modify` from the `Actions` "pulldown" menu on the upper border of the screen.
6. Entering parameters, such as `Change password`.

7. Performing the task by pressing OK.

You can always back up to a previous level, or to the Control Box, by selecting Previous Level. Exit SAM by selecting Exit.

This section describes SAM on a graphical display. SAM also displays in a similar way on character terminals. The same functions are accessed using the arrow, (Tab), and (Return) keys.

The following table summarizes the SAM screens you use, along with additional sources of information, for some of the more common system administration tasks.

Note

- Click on an item to "select" (illuminate) it. Then open that screen by clicking on Open. You may also "select and open" with one action by double-clicking on the item, such as Process Management, just as in other HP VUE applications.
- If you are using a character terminal, use (Tab) and the arrow keys to select an item, and press (Return) to open it. Use the softkeys for their indicated functions.
- Help is available at the Control Box or on individual tasks by selecting Help.

The following is a summary of some common tasks you can do in SAM. For information about other SAM functions, see *System Administration Tasks*.

Table 12-1. A Selection of Basic SAM Tasks

Task	Control Box Selection	Next Screen Selection	Additional Action or Information
Managing or getting information about Users or Groups	Choose `Users and Groups`	`Users` or `Groups` →	Use information or do actions which are available in the `Actions` pulldown menu.
Managing peripheral devices.	Choose `Peripheral Devices`	Select the appropriate subarea.	Actions are available in the `Actions` pulldown menu.
Managing processes	Choose `Process Management`	`Process Control`	Actions, such as killing a process, are available in the `Actions` pulldown menu.
Controlling excess file space usage	Select and open `Routine Tasks`	`Disk Space Recovery`	Select from the options available in the next screen. For File Removal, modify search parameters as needed.
Getting information on performance	Choose `Process Management`	Select and open `Performance Monitors`	Select and open performance tools such as `Top` or `Monitor`, if available on your system. The options depend on what is installed on your system.

Table 12-1. A Selection of Basic SAM Tasks (continued)

Task	Control Box Selection	Next Screen Selection	Additional Action or Information
Performing tasks automatically	Choose Process Management	Job Scheduling via Cron	Add or modify tasks, for specific times. See "Running a Command at a Specified Time with crontab" for information on using `cron` from the command line.
Backing up files and software	Choose Backup and Recovery	Backup Devices or Automated Backups .	Either select Automated Backups directly, or first select Backup Devices and view or modify the configuration of your backup devices first. If you look first at the Backup Devices screen, select Automated Backups from the List menu to set backup times and frequencies.

Table 12-1. A Selection of Basic SAM Tasks (continued)

Task	Control Box Selection	Next Screen Selection	Additional Action or Information
Installing a storage device on HP-UX	Choose `Disks and File Systems`	Choose a category from the list of disks and file systems	Highlight a selection from the list of attached devices. Actions, such as adding or removing a device, are available in the `Actions` pulldown menu. See *System Administration Tasks* for details.
System administration to a remote system	Choose `Remote Administration`	Select from the `Actions` menu to add or remove systems or run SAM remotely.	See *System Administration Tasks*.
Installing/configuring a printer	Choose `Printers and Plotters`	`Printers/Plotters`	Configuration actions are available for selected items from the `Actions` pulldown menu. See "Getting Information on Printers", in this chapter. Also see the manuals *System Administration Tasks* and *Installing Peripherals*.

12

Running a Command at a Specified Time with crontab

If you are logged in as `root`, or if you have specific permission as determined by an "allow" list, you can run commands automatically at remote times and at regular intervals. This is done by the `cron` utility.

The `cron` utility is useful if you want to run a resource-intensive command at a time when few demands are being placed on the system. You can also use it to do routine jobs, such as system backup (or see the SAM procedure for this in Table 12-1, in this chapter), or automatic erasure of unneeded temporary files. The `cron` utility can even be used to mail to mail yourself a reminder of a weekly meeting.

You determine `cron` behavior either by using SAM, or by using `crontab`, which is a utility which configures `cron` actions.

`cron` is started by `/etc/rc` each time you boot your system and, by default, runs continuously. Its actions are determined by the contents of the `/usr/spool/cron/crontab` file. You determine the commands run by `cron` and the times at which they are run by using `crontab` or SAM to make entries in the `usr/cron/crontab` file. Use either one of the following procedures:

1. Using SAM

To use SAM in order to access the `crontab` file and add or delete specific actions to perform at given times,

1. Enter `/usr/bin/sam` as `root`.
2. Select `Process Management` from the control box.
3. Select `Job Scheduling via Cron`.
4. Use the `Actions` options to add, remove or modify Cron entries. The screens will give you the alternatives of filling in times, days, dates, or months for performing a specific command.

Select `Help` to get suggestions as to the actions which are possible on a given screen.

If you want to remove or check the contents of your `crontab` file from the command line, without entering SAM, see the next section.

2. Using Crontab Directly

As root, you can use Crontab. If you are logged in as user, permission to use crontab is determined by two files in /usr/lib/cron, called cron.allow and cron.deny. These contain lists of user login names. If a request is made by a user whose login does not appear in cron.allow, (or cron.allow doesn't exist) the command checks cron.deny for the presence of the login. Users with login names in cron.deny cannot access this command. If neither file exists, only someone with root privileges is able to use crontab.

A crontab file consists of lines of six fields each. The fields are separated by spaces or tabs. The first five are integer patterns that specify the time(s) of execution.

To create a crontab command file, which will overwrite any previous one, enter crontab and type asterisks (*) or the appropriate time information, followed by the commands you want to have executed. Here is an example:

- 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).

Each of these patterns can be either an asterisk (meaning "all legal values"), or a list of elements separated by commas. An element is either a number, or two numbers separated by a hyphen (meaning an inclusive range).

For example, enter crontab followed by your specifications, and then terminate the input with CTRL-D:

```
crontab

30  8  *  *  4  echo "Staff meeting today at 1:00PM."
0   0  *  *  *  rm * .tmp 2> errfile
CTRL-D
```

This file means the following:

- Line 1: On Thursday at 8:30AM, cron sends you a reminder of your 1:00 staff meeting. The first field (30) indicates 30 minutes past the hour. The second field (8) indicates the hour. The third and fourth fields contain asterisks (*), which indicate all legal values. The fifth field (4) indicates Thursday.

The `cron` command sends output and error messages to your mailbox, unless you specify that they be redirected to a file. The result of line 1 is the following mail message, every Thursday morning at 8:30AM:

```
Staff meeting today at 1:00PM.

********************************
Cron:  The previous message is the standard output and
standard error of one of your cron commands.
```

- Line 2: At midnight, everyday, `cron` erases all files in your home directory which have a `.tmp` extension. Also, any error message is redirected to `errfile`, in your home directory.

You can get a listing of the contents of your `crontab` command file by entering the following:

```
crontab -l
```

You can remove your crontab file by entering the following:

```
crontab -r
```

Getting Information on Printers

Using SAM:

You can use SAM to set up and install a printer on HP-UX. See the *Owner's Guide* for your system for the procedure for doing this.

To use SAM in order to get information on the current status of your printer(s), do the following:

1. As `root`, enter `/usr/bin/sam`.
2. Select `Printers and Plotters` from the Control Box.
3. Select `Printer/Plotters` on the next screen.
4. You will see a display listing the status of all the currently-connected printers. Highlight the one which you are interested in. You will see a display like the following:

```
Name                 Status            Fence        Type       Location
                                       Priority
lp1                  enabled, idle     0          local        /dev/null
lp2                  enabled, idle     0            remote     /dev/null
test                 disabled          0            local      /dev/null


     . . .
```

5. To change the status of a highlighted device, select an action from the `Actions` menu.

In case of printer problems, choose `Show Common Problems...` from the `Actions` menu, for useful troubleshooting information.

For more information on printer configuration, see *Installing and Updating HP-UX 9.0* and *System Administration Tasks*. To get information on current printer status, also see *lpstat*(1).

Using HP VUE:

You can also get a similar listing of printer information by clicking (once) on the Printer icon on your HP VUE Front Panel.

Backing Up Your System and Software

The most important part of your system is the data you have accumulated. It is also especially important to protect your system in general from corruption if your HP-UX has been supplied as a pre-installed "Instant Ignition" disk. You can protect your data and system from loss, using the procedures given in this chapter:

- Make sure you create and maintain a backup HP-UX kernel (SYSBCKUP) on your disk from which you can boot in an emergency. A SYSBCKUP is automatically created by SAM whenever you reconfigure and reboot a new kernel from your system console.
- To build your recovery system, you can use the following devices:
  - □ Cartridge tape drives
  - □ DDS Format ("DAT") drives
  - □ Magneto-optical disk drives
  - □ Other hard disk drives
- Back up your file system.
- Perform automated backups using `crontab` via SAM or directly. (See "Running a Command at a Specified Time with crontab")
- Restore your file system if needed.

If you received your workstation with Instant Ignition, it is important to create your first recovery tape and to archive your existing file system *as soon as possible*.

Creating a Recovery System

A "recovery system" is a special tape containing a subset of the HP-UX operating system. In the event of an operating system failure that prevents you from booting or logging into HP-UX, you can boot from the recovery system tape and use the tools on the tape to repair the file system on your disk. A recovery system is created by using commands rather than by using SAM.

You can also restore your system from a system CD ROM that you can purchase from your HP sales representative.

You should make a recovery tape using `mkrs` at the following times:

- Immediately after you set up your new workstation.

- Each time you update your operating system or make a change in your disk swap configuration.

To do this, you will need a tape drive (cartridge or DDS-format) and one or more tapes.

Using mkrs to Create a Recovery System:

The `mkrs` command constructs a recovery system on removable media (or a formatted hard disk drive). If a system is unbootable due to a corrupt root disk, then you can boot your system from the recovery tape. Once booted on the recovery system, you can then use the tools it provides to repair the corrupt root disk.

Options with mkrs

Note The -s option is necessary for building Series 700 DDS-format tape recovery systems.

If enough free disk space is available in `/usr/tmp` (typically 10-20Mb), the `-q` option can be used to make mkrs create an image of the recovery system in this directory before copying it to the recovery media. This option generally saves a great deal of time due to reduced seeking on non-random-access recovery media (cartridge tape and DDS-format). Note: for DDS-format tape recovery systems, the `-q` option is assumed.

When creating DDS-format recovery system for a small memory system (8Mb or less), the `-s` option should be used to specify that a smaller set of files be placed on the recovery system. Note that the DDS-format device must be SCSI.

Source Device Files

By default, `mkrs`, by default, uses the following device file:

```
/dev/update.src

/dev/rct/c0

/dev/rct
```

If none of the above defaults exist on the system, one of these device files must be created or the `-f` option must be used to specify the device file to be used (the recovery device file can be either a block or a character device file).

Root Device Files

`mkrs`, by default, uses the following device files for the root device:

```
/dev/dsk/0s0
/dev/root
/dev/hd
```

If none of the above defaults exist on the system, one of these device files must be created or the `-r` option must be used to specify the device file to be used. The root device file must be a block device file.

If You Have a Problem

An error message results if:

- None of the default device files for the recovery device exist and the -f option is not used to specify a recovery device file.
- None of the default device files for the root device exist and the -r option is not used to specify a root device file.
- The machine type cannot be determined and the -m option is not used to specify the machine type.

Backing Up Your File Systems

Preparation:

The following procedure sets up a scheduled backup:

1. If your system is more than six months old, you have non-HP supported software, or you have never done a backup before, see *System Administration Tasks* or *Installing and Updating HP-UX 9.0* before proceeding. Otherwise, the interactive guidance provided by SAM will suffice to get you through the process.
2. Log in as `root`.
3. Type `/usr/bin/sam`.

4. Choose Backup and Recovery from the Control Box.
5. Choose Backup Devices (to determine what backup devices are connected) or Automated Backups.
   a. If you opened Backup Devices, and no devices are shown, make sure the device is connected and the tape is inserted.

Note If you have to connect a device during this process, choose Refresh from the Options menu.

   b. Choose the the entry for the desired device.
   c. Choose Add an Automated Backup from the Actions menu.

1. If you go directly to Automated Backups, you can select your local or remote backup device from Actions → Add an Automated Backup (Local or Remote) → Specify Backup Device. A form will display any existing backup devices.
2. Before initiating the backup, verify that your tape is *not* write-protected by checking that the write-protect device is in "writable" position.
3. Load a tape into the tape drive. Depending on what tape drive you are using, you may see activity lights flashing while the tape is loading. You can proceed when one light remains on, indicating that the drive is ready to accept data.
4. From the Add an Automated Backup screen, you can select the options which will bring up additional forms for specifying the following required items:
   a. Select Backup Device (if you have not already specified it).
   b. Select Backup Scope: what filesets to include or exclude. The default is to backup the entire system.
   c. Select Backup Time: time, day, date. You can also set whether you want a full or incremental backup, for each time specification.
5. Additional Parameters (optional) allows you to specify the following:
   a. If you want your backups to cross NFS mount points.

b. If you want the tape (cartridge or autochanger) rewound.
c. If you want an index log to be created for each backup. (Results can be mailed to a specified user).

6. After the minimum required forms are filled out, SAM will then use your specified tape device to complete the backup according to your specifications.

Restoring Individual Files

To restore specified files from a local device using SAM, first you will need to have the following information and materials:

- A list of files you need.
- The media on which the data resides.
- The location on your system to restore the files (original location or relative to some other location).
- The device and device file for restoring the data.

Note

- To restore data to disks physically connected to another computer, enter the `Remote Administration` functional area of SAM.
- When restoring files that are NFS mounted to your system, `frecover` can only restore those files having "other user" write permission. The `frecover` command normally operates in user-mode when crossing NFS mount points; not superuser-mode. To ensure that `frecover` can restore the files exported from the NFS server, login as superuser on the NFS file server and use the `root=` option to the `/usr/etc/exportfs` command to export the correct permissions. Refer to *exportfs*(1M) in the *HP-UX Reference* and the *Installing and Administering NFS Services* manual.

To restore individual files:

1. Ensure that you have superuser capabilities.
2. Run SAM; type:

```
/usr/bin/sam
```

3. Choose `Backup and Recovery`
4. Choose `Backup Devices`.
5. Highlight the device in the list from which the data is to be restored.

6. Choose `Recover Files or Directories` from the `Actions` menu.

7. Turn on the `Selected Files` checkbox. Do either of the following:

   - Fill in the filename containing a list of files to restore. The filenames should be full pathnames. This file is *not* a graph file. This file is used to create a graph file. You can use the on-line index file created by a previous backup, but it must be edited to containing only the full pathnames of the files to be restored.
   - Or enter each file name in the "Included" and "Excluded" boxes and click on `Add`. If you make a mistake, highlight the entry with the error and use `Modify` or `Remove` to correct the mistake.

     You can use both the file and the included/excluded method simultaneously to specify files to be restored.

   When you have completed determining the selected files to be recovered, click on `OK`.

8. To do the following:

   - Overwrite new files.
   - Maintain original ownership.
   - Recover files using full path name, or
   - Place files in a non-root directory.

   Activate `Set Additional Parameters`.

   Turn on the appropriate checkbox.

   To restore files relative to a particular directory, fill in the directory.

   Activate the `OK` control button to set the additional parameters.

9. Activate the `OK` control button to start the restore process.

   If confirmation messages appear, read the message(s) and activate the `OK` control button to proceed in each case. SAM displays a window containing the output of the executed `frecover` command.

Restoring Your File System

If your file system becomes corrupted and not usable, you can restore your system with your recovery system tape and your archive backup tape(s).

To restore your system:

1. Turn off the system, except for the tape drive.
2. Load the recovery system tape which you have previously made into the tape drive and wait for the indicator lights (if your drive has them) to stop flashing. One light will remain on when the tape drive is ready.
3. Turn on the power for your workstation and hold down the keyboard space bar until you see the `SEARCHING FOR A SYSTEM` message on your display.
4. Wait for a list of available operating systems to appear, with a message in the upper right part of the screen similar to the following:

   ```
   :HP35450A -A REMV,1401,0
      1R SYSRECOVER
   ```
5. Type the digit-character combination to the left of the system you want to boot, for example, `1R` to designate `SYSRECOVER`. The workstation will boot HP-UX from the tape. Booting will take several minutes.
6. Eventually, a menu similar to the following appears:

   ```
   1) Remove the root password

   2) Work in a shell to perform recovery manually

   3) Perform an automatic recovery

   4) Exit recovery system and reboot root file system

   5) Help
   ```

   Choose option 3. This replaces key files so that the disk can be used as a root disk again. The replaced files are saved in a directory under `/tmp`.
7. Choose option 4 to reboot the file system from the disk.
8. After your system has successfully rebooted, unload the recovery tape and load the first archive (file system backup) tape. Wait for the tape drive lights to stop flashing.
9. Type the following command exactly as it is shown here:

```
cd / ; tcio -i /dev/update.src | cpio -icdumx  <Return>
```

You will be prompted for the remainder of your backup tapes, as they are needed.

10. Load your latest incremental backup tape into the tape drive. This tape contains all the files that changed or were added since the archive tape was made. Wait for the lights to stop flashing. Then type the same command line to read in the changed files.

For more information on backup and recovery, see the *HP-UX System Administration Tasks*, and the references for *mkrs*(1m), *cpio*(1), and *tcio*(1) in the `man` pages or in HP VUE Help.

Enabling and Disabling HP VUE

Some systems are set up to automatically start Login Manager (the portion of HP VUE responsible for starting the X server), during the system `init` process. These systems automatically display the HP VUE Login Screen when they are rebooted. Other systems may be initially configured to run a console.

You must have root permission to configure a system to start or stop HP VUE.

In case your system has the HP VUE filesets but does not have VUE activated, the following procedure will ensure that VUE becomes your default login. There are two ways to start HP VUE:

1. You can configure the system to automatically start HP VUE when the system is rebooted. This is done by changing the default run level for the system.
2. You can manually change the system run level from the console. This has no effect on what happens when you reboot your system.

Once a system is running Login Manager, HP VUE can be halted by one of the following:

- Changing the default run level (by editing `/etc/inittab`).
- Manually changing the run level from the console (using `telinit`).

Enabling HP VUE

If your system has one of these special configurations, you may need to edit certain Login Manager files before starting HP VUE:

Requirements for Running HP VUE

- You must have a bitmapped console.
- The system must have at lease 8 MB of memory.

Otherwise, if any of the following apply, see *HP VUE User's Guide* for procedures for editing Login Manager files:

- The system is part of an HP-UX cluster.
- The system is an X terminal or a host for X terminals.
- The system has more than one display.

Starting HP VUE Automatically at Boot Time

When HP VUE is started automatically, the HP VUE Login Manager is run when the system is booted.

To start HP VUE automatically, the system default run level must match a run level assigned to HP VUE. The default run level and run level assignments are made in `/etc/inittab`.

You can modify the default init state in `/etc/inittab` using one of the two methods shown below:

- If the IGNITION fileset is installed on your system, you may be able to use the Instant Ignition configuration tool, in the script `/etc/newconfig/Ignition/configure.sh`.
- You can manually edit `/etc/inittab` in a text editor.

Using configure.sh to Edit inittab

Requirements for Using 'configure.sh'

You can use `configure.sh` to edit `/etc/inittab` if your `/etc/inittab` file has not been extensively customized.

If your system is not suitable for running the `configure.sh` script, you must edit `/etc/inittab` using a text editor.

Procedure for Running configure.sh

1. Log in as `root`.
2. Run the script by executing:

   ```
   /etc/newconfig/Ignition/configure.sh
   ```

   For an HP-UX cluster, you must run the script on every cnode that will be running HP VUE.
3. Respond appropriately to the prompts.

The script changes the system default run level to 4. This is the preferred HP VUE state, as both HP VUE and a console are both invoked.

Editing /etc/inittab Manually

1. Log in as `root`.
2. Make a backup copy of `/etc/inittab`.
3. Edit `/etc/inittab` (in an HP-UX cluster, `/etc/inittab` is a context-dependent file).

   The default run level (`initdefault`) must match a run level specified for respawning HP VUE (`vue`). The `/etc/inittab` file shipped with HP-UX 9.0 systems respawns HP VUE at run levels `3` and `4`; you should avoid changing this line.

Example

The following portion of `/etc/inittab` illustrates how to start HP VUE automatically when the system is booted. The default run level is set to `4`, which is a run level assigned to HP VUE.

```
init:4:initdefault:
    .
    .
    .
cons:012456:respawn:/etc/getty -h console console
vue :34:respawn:/etc/vuerc
```

Starting HP VUE Manually

If Login Manager is not started automatically when the system is booted, it can be started later from the console.

1. Log in to your system console as root.
2. Examine (for example, with the `page` command) the contents of `/etc/inittab`. Look for the line that respawns HP VUE, and make note of the run level(s) used. For example, the following lines specify that HP VUE runs at run levels 3 and 4.

   ```
   vue :34:respawn:/etc/vuerc
   ```

3. Execute:

   ```
   /etc/telinit n; exit
   ```

where *n* is a run level assigned to HP VUE.

Example

If /etc/inittab contains these lines:

```
init:2:initdefault:
    .
    .
    .
cons:012456:respawn:/etc/getty -h console console
vue :34:respawn:/etc/vuerc
```

then HP VUE does not start when the system is booted. However, you can start HP VUE by executing:

```
/etc/telinit 3; exit
```

or

```
/etc/telinit 4; exit
```

Disabling HP VUE

There are two ways to stop HP VUE:

1. Change the system default run level so that HP VUE does not start when the system is rebooted. This is done by changing the default run level.
2. Change system run levels manually.

Configuring the System for Console (Non-VUE) Login at Boot

The following procedure prevents HP VUE from starting automatically when the system is booted. However, HP VUE can be started later by changing run levels.

1. If you are in an HP VUE session, log out.
2. Use the Options menu on the login screen to enter No Windows mode.
3. Log in as `root`.

4. Make a backup copy of `/etc/inittab`.
5. Edit `/etc/inittab`. Change the default run level (`initdefault`) to a run level that runs a console but does not run HP VUE.

Example

The following lines show a portion of `/etc/inittab` in which the default run level is set to 2, which runs a console. Since the file specifies that HP VUE runs only at run levels 3 and 4, HP VUE does will not start automatically when the system is booted.

```
init:2:initdefault:
    .
    .
    .
cons:012456:respawn:/etc/getty -h console console
vue :34:respawn:/etc/vuerc
```

Stopping HP VUE Manually

1. If you are in an HP VUE session, log out.
2. Use the Options menu on the login screen to enter No Windows Mode.

   Alternatively, you can log into a failsafe session.
3. Log in as `root`.
4. Switch to a run level that does not run HP VUE by executing:

   ```
   /etc/telinit n; exit
   ```

   where *n* is a run level for which `/etc/inittab` specifies a terminal or console and does not specify HP VUE.

Example

Suppose the contents of `/etc/inittab` is:

```
init:4:initdefault:
    .
    .
    .
cons:012456:respawn:/etc/getty -h console console
```

```
vue :34:respawn:/etc/vuerc
```

The following command would stop HP VUE:

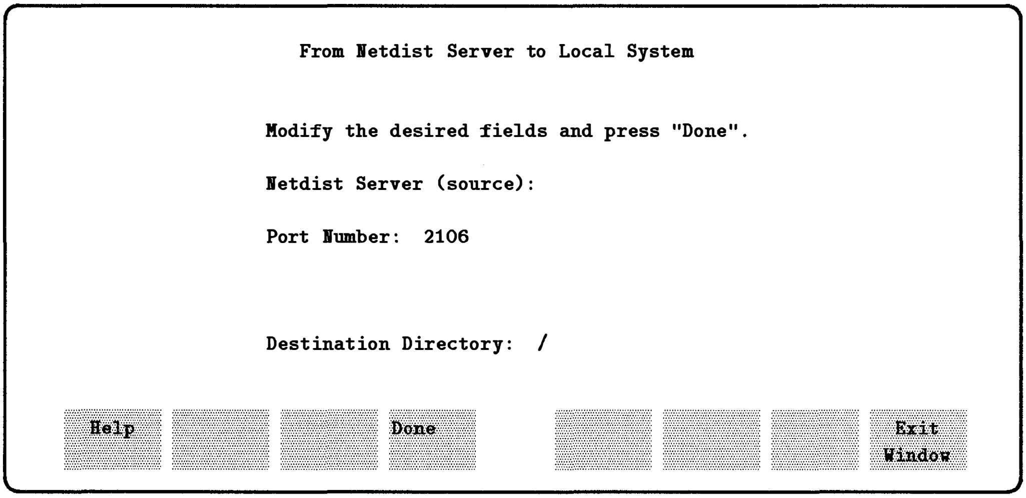
```
/etc/telinit 2; exit
```

Updating from a Network Server

You can update software on your system by using the `/etc/update` utility.

To update from a netdist server:

1. Perform any necessary planning steps. (See the appropriate chapters of *Installing and Updating HP-UX 9.0*).
2. As `root`, start the `/etc/update` program.
3. Select `Change Source or Destination` from the main menu.
4. Select `From Netdist Server to Local System`. The following menu appears:



```
From Netdist Server to Local System

Modify the desired fields and press "Done".

Netdist Server (source):

Port Number:  2106


Destination Directory:  /
```

5. In the `Netdist Server source` field enter the system name or the internet protocol address of the `netdist` server that will be the source of the update. To find out the system name, enter `hostname` on the `netdist` server, or look in the `/etc/hosts` file for the internet address. (There is no default.)
   a. The `Port Number` field indicates the network port number where the `netdist` server's `netdistd` command is responding to requests

for updates. The default is 2106, or the number associated with the "netdist" entry in the /etc/services file on your system.

b. The Destination Directory field indicates the name of the directory on the local system where the files should be loaded. The default is root (/).

6. When you're finished making changes, press Done to return to the Main Menu.

For More Information

See *System Administration Tasks* and *Installing and Updating HP-UX 9.0* for detailed procedures for updating from a netdist server.

Shutting Down Your System

If you need to cycle power on a system using a local disk, you will have to execute the `shutdown` command first. You can do this either from the command line, with SAM or by using the HP VUE Toolbox.

Using the shutdown Command to Stop Your System

Caution Do not turn off power to your system without first shutting down the operating system software according to the following procedure. Turning off the power for your computer without first doing the shutdown procedure may result in damage to data on your disk. Always execute the shut-down process to completion first.

1. As `root`, enter the command `shutdown -h`. This will give you and any other users on your system a one-minute "grace period" to save files and terminate processes before the system goes down to the halted state.
2. You will see a message:

   ```
   Waiting a grace period of 60 seconds for users to logout.  Do not turn
   off the power or press reset during this time.
   ```

   (You can specify this message and you can determine the "grace period" that `shutdown` allows. See *shutdown*(1M)) and *System Administration Tasks* for using various options.
3. At the end of the period, you will see another warning and the following request for confirmation:

   ```
   Do you want to continue?  ...
   ```

4. Respond with `y`. You will see another message confirming shutdown. Finally, you will see the following message:

   ```
   Halted, you may now cycle power.
   ```

5. At this time the system no longer responds to keyboard input and you may turn off the power. Turning the system back on again will initiate the boot process.

If you want to shutdown and reboot automatically enter the following:

```
shutdown -r.
```

Using SAM to Shut Down Your System

Caution Do not turn off power to your system without first shutting down the operating system software according to the following procedure. Turning off the power for your computer without first doing the shutdown procedure may result in damage to data on your disk. Always execute the shutdown process to completion first.

You can login as `root` and shut down your system, using SAM.

1. Enter `sam` as `root`.
2. Select `Routine Tasks` from the Control Box.
3. Select `System Shutdown`.
4. You will be given a choice of the following:
   a. `Halt the System`. All currently executing processes except those essential to the system are terminated. Then the system is halted.
   b. `Reboot (Restart) the System`. The system is shut down and rebooted automatically.
   c. `Go to Single User State`. The system is put in single-user mode for administrative purposes such as backup or file system consistency checks.

Using HP VUE to Shut Down Your System

Caution Do not turn off power to your system without first shutting down the operating system software according to the following procedure. Turning off the power for your computer without first doing the shutdown procedure may result in damage to data on your disk. Always execute the shut-down process to completion first.

1. First make sure you are logged in as `root`. If you are not, click on the logout button, as described previously. Then, log in as `root`.
2. Click on up arrow *above* the Toolbox icon at the right of the Front Panel.
3. Click on the `General` icon on the subpanel which appears.
4. Double-click the `System_Admin` subdirectory.
5. Double-click the `Halt_System` action in the `System_Admin` subdirectory. (Note that `Halt_System` has online help which can be accessed using the `Help` button on the `Halt_System` screen).
6. To observe the `shutdown` messages, you may need to bring the window to the front by clicking on it. Wait for the following message to appear on your screen:

   ```
    . . .

   Halted, you may now cycle power.
   ```

7. You can now safely turn off the power for your computer.

Chapter Command Summary

You should be familiar with the appropriate section in HP-UX Reference for more information and options before using system administration commands.

Table 12-2. Some Basic System Administration Commands

To Do This ...	Type This ...
Invoke SAM	`/usr/bin/sam`
Display disk usage	`du`
Compress files	`compress`
Uncompress files	`uncompress`
Remove a fileset	`rmfn`
Run a scheduled command	`crontab`
Create a recovery system	`mkrs`
Stop your system	`shutdown`

13

Networking with HP-UX

Chapter Contents

13

HP-UX Network Services

Your HP-UX system can use a variety of networking services to enable you to transfer copies of files to other computer systems. These services can also enable you to log onto remote machines on the network and run commands and processes remotely.

Specifically, this chapter gives you information on using HP-UX and ARPA/Berkeley Services to do the following tasks:

- Copying Files to and from a Remote Computer: `ftp`.
- Copying Files Remotely: `rcp`.
- Logging onto Another Computer on the Network: `rlogin`.
- Running a Command Remotely: `rmsh`.
- Sending Files with Kermit.

For information on using HP-VUE on remote systems, NFS-mounting remote file systems, and exporting file systems to remote systems, please see "Networking and Distributed Computing with HP VUE", in this chapter, *HP VUE User's Guide*, or *Using Network Services*.

Copying Files Using ftp

The ftp file transfer program allows you to copy files between your local system and remote systems and among remote HP-UX, UNIX, and non-UNIX network hosts that support ARPA services. The ftp program not only allows you to perform remote file copying , but also facilitates file management operations such as changing, listing, creating, and deleting directories on a remote system for which you have a valid login or account.

Using ftp you can copy a local file to a remote file or vice versa. You can also append a local file to the end of a remote file. The file to which you are copying can have either the same or a different directory path and/or name as the one on the originating system.

Preparing to Use ftp

- Make sure that your /etc/hosts file contains entries for the remote hosts with which you will communicate.
- Have the system administrators for the remote hosts arrange to give you a password and an account, or a login guest account, so that you can log in on the remote hosts.

Transferring Files with ftp

1. To invoke `ftp` and connect to a remote host in one step, type the following:

   ```
   ftp remote_hostname
   ```

   This connects you to the remote host. `ftp` then confirms the connection and prompts you for a remote login name:

   ```
   Name (remote_hostname):
   ```

   If you intend to log in with the same remote login name as you local login name, just press (Return).

2. Enter the password associated with your remote login name and `ftp` will confirm this action with a message and a confirmation that you are logged in:

   ```
   Password (remote_hostname):

   Password required for remote_login_name
   User remote_login_name logged in.
   ```

3. If you are going to transfer binary (as opposed to "readable" text) files, type `bin` at the prompt, before proceeding.

- Use `get` to transfer files from a *remote host* to your *local directory*.
  - □ At the `ftp>` prompt, type:

    ```
    get remote_filename
    ```

    The *remote_filename* is the name of a file in the remote working directory. In that case, `ftp`, copies the file to the local working directory and gives it the same file name. If the file is in another directory on the remote host, *remote_filename* is the absolute or relative path for that file. The `ftp` program copies the file to a file name with the same path on your local system. (For example, `get /user/doc/`*filename*). If there is no matching path, `ftp` gives you a message, "`No such file or directory`". If the destination file already exists, `ftp` overwrites its contents with the contents of the remote file.

    When copying successfully, `ftp` gives you messages confirming the copy and the length of time it required.

- Use `put` to transfer files from your *local directory* to a *remote host*.
  - □ At the `ftp>` prompt, type:

    `put` *local_filename remote_filename*

    - In this case, *local_filename* is copied to the remote file name in the specified remote directory.
      - □ *local_filename* can be the name of the local file in your current local working directory. `ftp` will copy the file into a file of the same name in *remote_file*.
      - □ *remote_filename* can be an absolute or relative path to a file name on the remote host. If not specified otherwise, it will be in the current working directory on the remote host.

Note If you have an `ftp` connection to a remote system, you cannot `put` or `get` a file back onto the same system. If you attempt this inadvertently, `ftp` gives you a message, "`No such file or directory`."

13

General File-Manipulation Commands for ftp:

Some of the set of `ftp` file-manipulation commands, such as `cd`, `mkdir`, `pwd`, and `rmdir` function in the same way as the corresponding HP-UX commands. Others, such as `append`, `delete`, and `lcd` have functions which are unique to `ftp`.

Table 13-1. File-Manipulation Commands for `ftp`

To Do This ...	Type This ...
Display the name of the current remote working directory	`pwd`
Display the name of the current *local* working directory	`!pwd`
Invoke a shell on the local host	`!`
Copy a local file onto the end of a remote file	`append`
Change the working directory on the remote host to *remote_directory*	`cd` *remote_directory*
Change the working directory on the local host to *local_directory*	`lcd` *local_directory*
List the contents of the current remote directory	`ls`
Create a remote directory	`mkdir` *remote_directory*
Delete a remote directory	`rmdir` *remote_directory*

If you need information on any of the ftp commands, just type help (or ?) at the ftp prompt.

Exiting ftp

To close the connection with the remote host and exit ftp, type:

```
bye
```

Copying Files Remotely Using rcp

You can copy files between HP-UX or other UNIX hosts on the network using `rcp`. Also, using appropriate options, you can copy directories between systems using `rcp`, if the configuration files that the service uses are set up properly.||

Preparing to Use rcp

Using `rcp` allows you to copy files and directories to and from a remote host and to copy among remote systems as well. To use `rcp`, you'll need the following prerequisites:

- An account (login) on the remote host.
- A `.rhosts` file in the remote host home directory containing the names of your local host system and your local login name.
- A `.rhosts` file on your local system, as well. This contains the names of all the systems you will copy from. It will ensure that you will be able to use `rcp` when you use `rlogin` on the remote system.
- A `/etc/host` file on your local system which lists hosts with which you can communicate using ARPA/Berkeley Services. For each host, the file has a line containing information about the remote host in the following form:

  internet_address official_name alias

You will find that the `/etc/hosts` file is useful for looking up names and addresses on the network. To facilitate such a lookup, use the `grep` tool described in *grep*(1), in Chapter 10, and in *System Administration Tasks*.

For example, an entry in the `.rhosts` file on the remote system might be:

```
hpabc leslie
```

where `hpabc` is the name of your local system and `leslie` is your local login name.

Note

It is important to protect your remote .rhosts file and home directory to prevent unauthorized users from gaining rcp access to your remote account. Only you should be able to write to the .rhosts file.

- Make sure it is owned by you.
- Use chmod to set the permission of .rhosts to 400 (-r-------). See Chapter 14 for details.
- Do the same to protect the rest of your remote home directory with at least 711 (-rwx--x--x) permission.

Copying a Local File to a Remote Host

To copy from your system to a remote system, use the following syntax:

rcp *local_filename remote_hostname*:*remote_filename*

Note that, if *local_file* is not in your current directory, you will need to supply the relative path (to get from your current directory) or the absolute path (from /), in addition to the local file name. You will need to specify the complete (absolute) path for the *remote_filename* on *remote_hostname* only if you want it to go into a directory other than the remote home directory.

For example, to copy myfile from your current directory to a remote system called xyz:

```
rcp myfile xyz:/users/leslie/otherdir
```

In this case, myfile will be copied as myfile into the remote subdirectory, otherdir. If you had only supplied the remote host name, rcp would have copied myfile into the remote home directory, also as myfile.

You can also include a filename in the destination. For example, to copy to a system named xyz:

```
rcp myfile xyz:/users/leslie/otherfile
```

In this case, you have copied myfile as otherfile, in the remote directory leslie.

Copying a File on a Remote Host to Your Local Directory

Now, to reverse the process, here is how you would copy a file *from* a remote host into your local directory. Use the following syntax:

rcp *remote_hostname*:*remote_filename* *local_filename*

For example, to copy `myfile` from your account in a remote system `xyz` into your current directory:

```
rcp xyz:/users/leslie/myfile .
```

13

The dot (.) is shorthand for "current directory". In this case, `myfile` will be copied as `myfile` from the remote directory into your current directory. You do not have to supply the destination filename if you don't want to copy it to a new name.

If you want to copy `myfile` into another directory in your home system, use a path name, absolute or relative, as shown:

```
rcp xyz:/users/leslie/myfile otherdir/
```

Or, if you want to copy the file to another file name in another directory:

```
rcp xyz:/users/leslie/myfile otherdir/otherfile
```

Run the `ls` command to confirm what you have done.

Copying a Local Directory and its Contents to a Remote System

To copy a local directory with all its files and subdirectories to a remote host, use `rcp` with the `-r` (recursive) option using the following syntax:

rcp -r *local_dirname* *remote_hostname*:*remote_dirname*

As before, if *local_filename* is not in your current directory, you will need to supply the relative path (to get from your current directory) or the absolute path (from /), in addition to the local file name. Also, the *remote_file* will require an absolute path (from /).

For example, you may want to copy a directory of miscellaneous files called `work` to a remote computer host called `abc`. To copy the entire `work` directory to a directory which is already there called `products`, type the following:

```
rcp -r work abc:/users/leslie/products
```

You have created a directory named **work**, with all its contents, in `abc:/users/leslie/products`.

Again, this assumes that you are doing this while you are in the local directory which contains **work**. Otherwise, you would have to give a relative or absolute path to that directory, along with its name, such as **/users/leslie/work**.

Copying a Remote Directory and its Contents to a Local System

To copy a remote directory with all its files and subdirectories to a local directory, use `rcp` with the `-r` (recursive) option in the following syntax:

```
rcp -r remote_hostname:remote_dir local_dir
```

For example, you may want to copy a remote directory called **work** to your current directory. To copy the entire **work** directory, type the following:

```
rcp -r abc:/users/leslie/work .
```

The dot (`.`) signifies the current directory, where the **work** directory will be copied. Run the `lsf` command to confirm that this has been done.

Logging In on Another Computer Using rlogin

If you have an account on a **remote host**, then you can use `rlogin` to log in on a remote host by supplying your remote login name and password. You can then work on that system just as you would on your home system.

If the remote host is configured to allow it, you can also log in on a remote host automatically, without having to supply your login name and password.

Logging In on a Remote Host

- At the shell prompt, use the form:

  `rlogin` *remote_hostname*

  The *remote_hostname* is the name of an appropriately configured remote system. As before, this system is named in your `/etc/hosts` file and in your `.rhosts` file. The remote host prompts you for your remote password.

- Enter your remote password. The remote host logs you in with the login message and the remote host prompt.

If for some reason you should make an error in entering your password, the remote host will give you the error message, `Login incorrect`, and will prompt you for your login, and your password:

```
Login incorrect
login:
```

Getting the Same Working Environment on the Remote Host

To get the remote host environment to behave in the same way as your home environment, you can set the `.profile` or `.login` values to be the same by copying your local `.profile` or `.login` files to your home directory on the remote system. As with your home system, the values in your `.profile` or `.login` will take precedence over the values in the remote system's `/etc/profile` or `/etc/csh.login` file.

Logging Out and Exiting the Remote Host

You can log out of the remote host just as you would from your home system, by typing:

```
exit
```

Typing CTRL-D also logs you out on most system.

At this point you are logged out of the remote host, disconnected, and returned to HP-UX on your local system, which displays a message and your local prompt:

```
Connection closed.
$
```

Temporarily Returning to Your Local System

If you wish to execute a command on your local system while you are in `rlogin`, type the `rlogin` escape character (normally a ~) followed by `!` and the command. (The "~" will be invisible until you type the "!" after it.) After the command has executed, `rlogin` returns you to the remote host. For example:

```
~! pwd
/users/leslie
[Returning to remote]
```

Press Return, or enter a command to redisplay the remote host prompt.

Running a Command Remotely Using remsh

Caution Do not use **remsh** to run an interactive command, such as **vi** or **more**. With some interactive commands, **remsh** hangs. To run interactive commands, log into the remote host with **rlogin**.

The **remsh** command enables you to execute a command on a remote host if the remote host is configured in either of two ways:

- You must have an account on the remote host with the same login name as your local login name.
- The name of your local host must be in the remote host's /etc/hosts.equiv file.

or:

- You must have an account on the remote host.
- The name of your local host and your local login name must be in a .rhosts file in your home directory on the remote host.

Caution A $HOME/.rhosts file creates a significant security risk. To prevent unauthorized users from gaining **remsh** access to your remote account and host, only you should be able to create a .rhosts file in your remote home directory and write entries to the file. To protect the file:

- Ensure that your remote .rhosts file is owned by you ("user").
- Use the HP-UX chmod command to protect your remote .rhosts file with 0400 (-r--------) permission.
- Use the HP-UX chmod command to protect your remote home directory so that no one else can read it or write to it. For example, you should probably protect your remote home directory with 0700 (-rwx------) permission.

For example, if your local host's name were `play.bc.com` and your local login name were jim, you would create on the remote host a `$HOME/.rhosts` file with the following entry:

```
play.bc.com jim
```

Executing Commands on a Remote Host as Yourself

At your HP-UX prompt, enter:

```
remsh   remote_hostname   command
```

where *remote_hostname* is the name or alias of a remote host and *command* is a command to execute on the remote host.

Sample entry:

```
remsh play cp form form.bkp
```

`remsh` executes the command on the remote host, and then your local host redisplays its prompt. You can then run the following to confirm that the file `form` has been successfully copied:

```
remsh play ls
```

For More Information

See *Using ARPA Services* or *hosts.equiv*(4) in *HP-UX Reference* for more details on running and configuring for `remsh`..

Displaying Remote Graphical Programs Locally

If you are running HP VUE or the X Window System, you can run a program using windows on a remote machine and display the results locally. This is done by setting the `DISPLAY` environment variable on the remote system. `DISPLAY` sets the host, display number, and screen number to which a system sends bitmapped output for clients.

For example, if the remote machine is called *remote*, your local system is *local*, and the remote program is called **`xwijit`**, enter the following on your system:

`xhost +`*remote* — *This enables your system to recognize the remote host.*

`rlogin` *remote* — *Log in on a remote machine on which you have an account.*

`DISPLAY=`*local*`:0.0` — *On the remote machine, set the DISPLAY variable to display on your local system.*

`export DISPLAY` — *Export the variable*

`xwijit` — *Run the program*

Using a Remote File System: NFS

The HP Network File System (NFS) services allow many systems to share the same files. NFS is independent of the operating system and can provide data-sharing among heterogeneous systems. Explicit file transfers across the network are unnecessary. Since access techniques are transparent, remote file access remains similar to local file access.

For More Information

For more information about setting up NFS-mounted file systems, see *Installing and Administering NFS Services*, *System Administration Tasks*, and appropriate entries in *HP-UX Reference*

For basic information on using HP VUE with network services, see "Networking and Distributed Computing with HP VUE", in this chapter, and *HP VUE User's Guide* for more details.

Transfering Files Among Different Systems: Kermit

Kermit is a protocol and a file transfer program for moving files, using RS-232-C connections, between many machines of different operating systems and architectures, such as Apple^{TM}, DEC^{TM}, IBM^{TM} PC, and UNIX. Kermit can also transfer files among workstations having different versions of HP-UX.

To transfer a file, `kermit` must be running on both systems involved in the transfer. There are different versions of `kermit`, and most work in similar ways. But you should also be prepared to encounter versions which function differently from the HP-UX version. Kermit can be run interactively or within a shell command line.

An Example of Using Kermit Interactively

The following example shows how to transfer a file from a mainframe computer to a PC, using the PC Kermit interactively. To alter the direction, you would reverse the role of the two Kermits (send from the PC and receive on the mainframe).

1. Start up Kermit on the PC local system by typing **`kermit`**.

   A display such as the following one appears:

   ```
   Kermit yyy . . .        Displays information about Kermit.

          Kermit-PC>              The Kermit prompt appears on PC.
   ```

2. Set line (if necessary). For example:

   ```
   set line ttyq1
   ```

3. Set speed (if necessary). For example:

   ```
   set speed 2400
   ```

   Kermit will confirm `line` and `speed` with messages.

4. Now connect with the other Kermit.

```
    Kermit-PC> connect          Connected to mainframe.

    CTRL-C                    Returns to PC.

    ...                         See escape character information.

    ...                         See more messages.
```

5. Log onto the mainframe using an account previously set up for you.

```
    $ login logname password    Log in on mainframe.

    $ kermit                    Start Kermit on mainframe.

    Kermit-MF>                  See Kermit prompt for mainframe.
```

6. Send the file named *report* to the PC:

```
        Kermit-MF> send report     Send file from mainframe to PC.
```

7. Move back to the original system:

```
                CTRL-C              Escape back to the PC.
```

8. Get the PC system ready to receive files:

```
          Kermit_PC> receive          Tell PC to receive files.

          ...                         See messages during transfer.

          Transfer Complete           Indicates end of transfer.
```

9. Go back to the remote system so you can shut down the Kermit:

```
                                Go back to Kermit on mainframe.

    Kermit-PC> connect
```

```
...                                         See messages again.
```

10. Exit from Kermit and logout from the mainframe.

```
Kermit-MF> exit                Exit from mainframe Kermit.

$ logout                       Logout from mainframe.
```

11. Move back to the original PC and exit from Kermit.

```
CTRL-C                         Escape back to PC again.

Kermit-PC> exit                Exit from PC Kermit.
```

For More Information

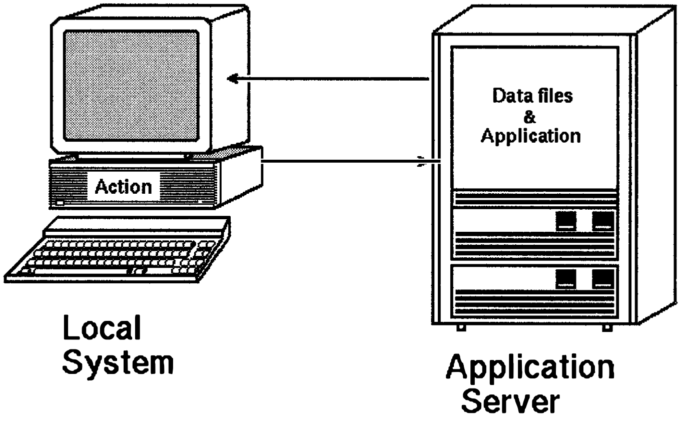
For more information on setting lines and running Kermit, with action and setting options, see the references to *kermit*(1), *cu*(1), *tar*(1), and *uucp*(1) in *HP-UX Reference* or in you online `man` pages.

Networking and Distributed Computing with HP VUE

For detailed information on running HP VUE in a networked environment, see the *HP VUE User's Guide.*

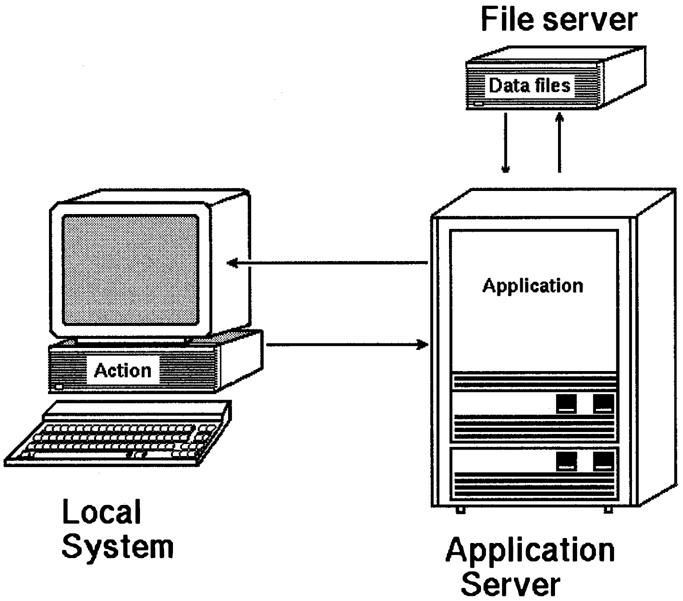
HP VUE is designed to work in a networked environment. This is especially the case with HP VUE actions, which are designed to provide "network transparency" for the user.

- An action defined locally can run an application on a remote host, or application server.



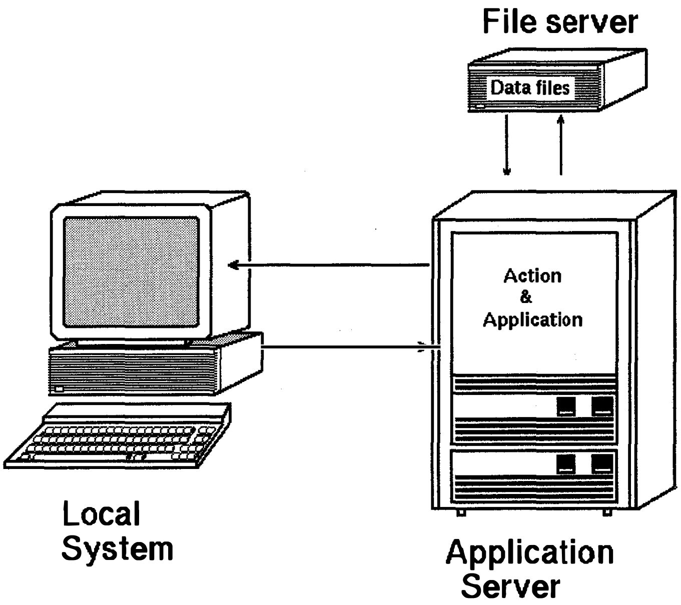
A local action can run a remote application.

- The data can be located on a remote file server. For example, an action defined locally can use an application on one remote system, which in turn uses data from another remote system.



A remote application can access data from a file server.

- An action can be defined remotely and used just as though it were defined on your system. Thus, you can use a host in your network as an action server.



An action can be defined on a remote system.

In X-terminal configurations, an additional host runs the X server.

For detailed information on networking with HP VUE, please see *HP VUE User's Guide*.

NFS with a Remote File System

The File Manager and actions use NFS to provide easy access to remote data files. Usually, the remote system files are mounted under `/nfs/`*hostname*.

Using NFS on the Data Host

The data host is the system whose files are made available or exported to remote systems. To make local directories or files available to remote systems, do the following steps:

1. Log in as root.

2. Enter **sam**. (If the system is running HP VUE, you can run SAM using the action in the `System_Admin` subdirectory of the General Toolbox.)

3. In SAM, select:

   ```
   Networking/Communications →
      Networked File Systems (NFS) →
         Local File Systems Exported
   ```

4. Choose Add from the Actions menu.

13

5. Use the SAM online help to provide permission for the local system to mount the file system on the data host. The local host *must* have permission to mount `/tmp`. It also must be able to mount directories containing the data files it needs to access.

Using NFS on the Local System

To access remote directories or files, do the following steps:

1. Log in as root. Enter **sam**.

2. Double-click on `Sam` in the `System_Admin` subdirectory of the General Toolbox to run Sam.

3. In SAM, select:

   -
     ```
     Networking/Communications →
        Networked File Systems (NFS) →
           Remote File Systems Mounted
     ```

4. Select `Add Remote Directory` from the Actions menu.

5. Follow SAM's on-line help to mount the data host's file system under `/nfs/`*hostname*.

Chapter Command Summary

Table 13-2. Networking Commands

To Do This ...	Type This ...
Invoke `ftp` and connect to *remote_host*	`ftp` *remote_host*
Copy files from *remote_host* to current directory, in `ftp`	`get` *remote_file*
Copy files from your local current directory to the current directory on *remote_host* in `ftp`	`put` *local_file*
List the contents of the current remote directory	`ls`
Exit `ftp`	`bye`
Also see Table 13-1 for File-Manipulation Commands on `ftp`	
Copy *local_file* to a remote host, using `rcp`, with full pathnames.	`rcp` *local_file remote_hostname*: *remote_file*
Copy a file from a remote host to your local directory, using `rcp`, with full pathnames.	`rcp` *remote_hostname*: *remote_file local_file*
Copy a directory structure from your local system to a remote host	`rcp -r` *local_dir remote_hostname*:*remote_dir*
Copy a directory structure from a *remote* system to your local system	`rcp -r` *remote_hostname*:*remote_dir local_dir*
Log in on a remote system	`rlogin` *remote_hostname*
Exit `rlogin`	`exit`
Run a command on a remote host	`remsh` *remote_hostname command*
List the contents of a remote home directory	`remsh` *hostname* `ls`

13

14

Making Your System Secure

Chapter Contents

- Security Strategies
- Securing Your Terminal
- Choosing a Secure Password
- Protecting Your Files and Directories
- Changing Who Has Access to Files and Directories
- Controlling Default Access Permissions

Security Strategies

HP-UX provides many security features to protect files from unauthorized access. However, you need to follow good security practices to maintain security on your system. The degree to which you need to enforce security measures depends on where you work, the security policy in your workplace, and the type of information with which you work.

If You are Running HP VUE:

See the *HP VUE User's Guide* for the details of securing HP VUE sessions and remotely-hosted processes.

If You are Using Shell Prompts:

This chapter summarizes the security strategies you should follow to help keep your system secure:

- Become familiar with the security policies of your workplace.
- Keep your terminal secure.
- Choose a secure password, and protect your password after you've chosen it.
- Be aware of who has permission to access your files and directories, and be able to control such access.

Note Security requires ongoing attention, and it may be impossible to have a 100% secure system under all circumstances. Therefore, this chapter gives you some guidelines for securing your system. However, even these cannot guarantee you a completely secure system.

Securing Your Terminal

When you are working with sensitive material, take care to position your terminal so the screen is not visible to others. Never leave your terminal unattended. Log off (`exit`) when you leave your terminal, or, if you're using VUE or the X Window System, lock your screen.

Guidelines for Securing Your Terminal

When working with sensitive material, take these security precautions:

- Position your terminal so the screen points away from open windows and doors.
- Never leave your terminal in a non-secure state:
  - □ Exercise care when logging in. Make sure no unauthorized person is observing you while you're entering your password.
  - □ Log off if you will be away from your terminal for a time.
  - □ Clear your display, even if you leave your terminal for a brief period. Type `clear` at the command line prompt. (Note that the `clear` command clears only the current screen; one can still scroll up and see previous screens.)

Note Check the security policies in your workplace. You may be required to log off whenever you leave your terminal, even if only for a brief period.

Working in an Audited Environment

HP-UX provides the capability to audit computer use, both on an individual and system-wide basis. Depending on how your system is configured, your actions may be recorded by an audit program. This subsystem monitors user actions at your terminal and records security-relevant information.

Choosing a Secure Password

When you choose a password, you want to ensure that no one can guess what you chose. If someone knows your password, that person may log in and access your files. This section offers suggestions on how to select and protect your password. These guidelines are of particular significance if you work with sensitive material.

What is a Secure Password?

When selecting a password in a secure environment, follow these guidelines:

- Choose a password that isn't publicly associated with you (your personal or professional life, your hobbies, and the like):
  - □ Don't use your name, your spouse's name, your children's names, or your pets' names.
  - □ Don't use the name of your street or your car.
  - □ Don't use phone numbers or special dates (anniversaries, birthdays, and the like).
  - □ Don't use your address, social security number, or license plate numbers.
- Choose a password that isn't listed in the dictionary (spelled either forwards or backwards). Password-cracking programs can use dictionary lists.

What *can* you use as a password? Here are a few suggestions:

- Make up a nonsense word.
- Make up an acronym.
- Misspell a word intentionally.
- String together syllables from a favorite song or poem.

Note HP-UX requires that your password be six to eight characters long. At least two of these characters must be letters (uppercase or lowercase); at least one character must be either a numeral (the digits 0 through 9) or a special character (such as -, _, or $). See Chapter 2 for some examples.

Protecting Your Password

When you have chosen your password, follow these guidelines to ensure that no one discovers it:

- Never write down your password.
- Don't tell others your password.
- Don't let others watch as you type your password.
- Don't store your password in the function keys of a terminal.
- Change your password occasionally (for example, once every three or four months). Refer to "Setting Your Password" in Chapter 2 if you need information on how to change your password.
- If you use more than one computer, use a different password for each system.

Protecting Your Files and Directories

Access permissions determine who can access your files and directories and the type of access allowed. You should always be aware of the permissions assigned. Check your files and directory permissions periodically to make sure appropriate permissions are assigned. If you find any unfamiliar files in your directories, report them to the system administrator or security officer.

Always carefully consider the permissions you allow on your files and directories. Give others access to them *only* when you have good reason to do so (if you are working on a group project, for example, your group may need access to certain files or directories.)

As you found in Chapter 4 and Chapter 5, the basic access permissions assigned to files and directories distinguish between three classes of users: owner, group, and other.

Each of the classes of users can access files or directories in any of three ways: read, write, and execute/search (`r`, `w`, and `x`). You can list the access permissions of your files and directories by using the `ll` command.

For information on access control lists (ACLs), which allow finer control of access to files, see *acl*(5) in *HP-UX Reference* and *HP-UX System Security.*

Access to Sensitive Files

Make sure that permissions assigned to sensitive files and directories are appropriate. Here are some general suggestions:

- Only you should be able to write to your home directory.
- Only you should be able to write to the files used to customize your home environment, for example, `.login` and `.profile`. (`.profile` is discussed in Chapter 11, in this manual, and in the *Shells: User's Guide.*)
- Only you (and the pseudo-group "mail," assigned to the mailer) should be able to write to your mailfile `/usr/mail/`*username*.

Changing Who Has Access to Files

If you want to change the basic access permissions assigned to a file, you can use the chmod ("change mode") command to control who has read, write and execute permission to your files. The chmod command sets a file's read, write, and execute permission for you, the file's group, and other users. Before using this command, you should always carefully consider what file permissions you give to others.

In general, give others access to your files *only* when you have good reason to do so (if you are working in a group project, for example, your group may need access to certain files).

See Chapter 5 for information on changing permissions for directories.

You can set permissions two ways: numerically or symbolically. The following sections will give you the details.

Using chmod to Set File Permissions Numerically

To illustrate the chmod command, you can set the permissions for **myfile** so that *only you* can read from and write to the file. A commonly-used syntax for chmod is as follows:

```
chmod number filename
```

Number is a three-digit number specifying what permissions you want to assign to the file. Each of the three digits sequentially sets permissions for each of the three classes: the owner, the group to which the owner belongs, and all other users. The *filename* is the name of the file you want to protect.

Also, each of the three digits can be thought of as made up of a sum of one or more of the following, taken separately for each of the three digits:

0	no permission (-)
1	execute (x) permission only
2	write (w) permission only
4	read (r) permission only

Then, by this scheme:

- 400 gives "read" (r--------) permission for you, the user, only.

- 440 gives "read" (`r--r-----`) permission for you and your group.
- 700 (4 + 2 + 1, 0, 0) gives "read, write, and execute" (`rwx------`) permission for you, but no permissions to anyone else.

Using chmod to Set File Permissions Symbolically

You can also specify permissions for `chmod` using the letters `u`, `g`, and `o`, as symbolic code for the owner ("user"), group, and others (the *class*). This "symbolic mode" is an easier scheme to remember than the numeric mode, since the symbols `r`, `w`, and `x` (the *mode*) are used directly as arguments in the command. Setting up the `chmod` syntax is a little different, making use of the `+`, `-`, and `=` signs. The syntax is:

$$\texttt{chmod}\ \textit{class}\begin{bmatrix}\pm\\=\end{bmatrix}\textit{mode}\texttt{,}\,[\ \dots\ ]\textit{filename}$$

For example, you can use the symbolic mode to create 644 (`rw-r--r--`) permissions by specifying the symbols `rw`, `r`, and `r` directly in the `chmod` command. "User" is represented by `u`, "group" by `g`, and "other" by `o`. To assign the permissions *absolutely*, use the `=` sign in the argument. Unspaced commas separate class-permissions:

```
chmod u+rw,g+r,o+r filename
```

To create 600 (`rw-------`) permissions and set "no permissions" for the classes `g` and `o`., use `=` with no symbol following:

```
chmod u=rw,g=,o= filename
```

Permissions are *added* with the `+` sign. Again, separate each class-permission with a comma and no space:

```
chmod u+rw,g+r,o+r filename
```

You can also *subtract* permissions from `u`, `g`, or `o`, using `-`, if you just want to restrict the level of permission from a previous "higher" level. For example, if you had set `rwxrw-rw-` and you wanted to change this to `rwx------` (`g` represents "group" and `o` represents "other"):

```
chmod g-rw,o-rw filename
```

However, unless you began with *no* permissions you may find that using `+` or `-` has added to, or subtracted from, some previously existing permissions for that file. Run the `ll` command to check this. If in doubt, set the permissions absolutely by using `=`.

For example, suppose you want to protect `myfile` so that neither you nor anyone else can modify it, but everyone can still read from it:

```
$  chmod 444 myfile
```

or

```
$ chmod u=r,g=r,o=r myfile
```

When permissions are being set the same, you can also combine the arguments as:

```
$  chmod ugo=r myfile
```

With only read permission on `myfile`, no one can write to it. Also, if you now try to remove `myfile`, the `rm` command asks you whether you really want to remove the file:

```
$ rm myfile
myfile: 444 mode? (y/n) n   If you do not want to remove it, enter n.
                            If you do want to remove it, enter y.
```

Later, if you want to permit yourself and members of your group to read from and write to `myfile`, use `chmod` as follows:

```
$ chmod 664 myfile
```

or

```
$ chmod ug=rw,o=r myfile
```

The `ll` command now should show:

```
-rw-rw-r--  1  leslie  users      154 Nov  4 10:18 myfile
```

For More Information ...

This section covered some of the most common uses of the `chmod` command for protecting files. To learn more about `chmod`, refer to the *chmod*(1) entry in the *HP-UX Reference*.

Table 14-1 summarizes the various chmod commands you can use to protect myfile.

Table 14-1. Examples of Uses of chmod

To Set Permissions so that ...	Type This ...	
	Numeric:	Symbolic:
Only you can read from myfile, and no one (including you) can write to it. Set permissions to -r--------.	$ chmod 400 myfile	$ chmod u=r,g=,o= myfile
Everyone can read from myfile, but no one can write to it. Set permissions to -r--r--r--.	$ chmod 444 myfile	$ chmod ugo=r myfile
Only you can write to myfile, but everyone can read it. Set permissions to -rw-r--r--.	$ chmod 644 myfile	$ chmod u=rw,go=r myfile
Only you and members of your group can write to myfile, but everyone can read it. Set permissions to -rw-rw-r--	$ chmod 664 myfile	$ chmod ug=rw,o=r myfile
Everyone can read from or write to myfile. Set permissions to -rw-rw-rw-.	$ chmod 666 myfile	$ chmod ugo=rw myfile
Only you can read from or write to myfile, but no one else can. Set permissions to -rw-------	$ chmod 600 myfile	$ chmod u=rw,go= myfile

Changing Who Has Access to Directories

In addition to changing permissions on files, the chmod command can also change permissions on directories. Using chmod, you can control who has access to your directories, and what kind of access they have. For example, you can protect a directory so that no one can list its files. Or you can control whether users can remove, rename, or alter files in a particular directory.

Table 14-2 defines some of the more common uses of chmod with directories. All the examples in this table assume that the directory projects exists under your current working directory.

Table 14-2. Setting Directory Protection for the projects Directory

To Set Permissions to ...	Type This ...	
	Numeric:	Symbolic:
Allow other users to list and access the files in projects, but not to create or remove files from it. Set permissions to drwxr-xr-x.	`$ chmod 755 projects`	`$ chmod u=rwx,go=rx projects`
Allow all users to list, create, remove, and access files in projects. Set permissions to drwxrwxrwx.	`$ chmod 777 projects`	`$ chmod ugo=rwx projects`
Allow only yourself to list, create, remove, and access files in projects. Set permissions to drwx------.	`$ chmod 700 projects`	`$ chmod u=rwx,go=- projects`

Note When determining who should be allowed to use your directories, be aware that *anyone who can write to a directory also can remove or rename a file in that directory*—even if that person cannot write to the file.

Controlling Default Access Permissions

In Chapter 4 and Chapter 5, you learned how to change the permissions on individual files and directories using the `chmod` command. You should also be aware of the default permissions assigned to all of your files and directories at the time you create them. You can list or change the default permission settings by using the `umask` command.

Default file permissions are assigned by the system whenever you create a new file or directory, and these are governed by your umask setting. Unless set up otherwise by you or your system administrator, your default `umask` setting will be 0, which means that new *files* you create will have read/write permission for everyone (666 or -rw-rw-rw-) and new *directories* you create will have read/write/search permission for everyone (777 or `drwxrwxrwx`).

You may want to change your `umask` setting to a non-zero value, to make the default access permissions to your newly-created files and directories more restrictive. You can do so using the `umask` command and either the numeric or, in the Korn Shell, symbolic arguments. For simplicity, the following shows you how to use `umask` with numeric arguments.

The number given as a parameter to the umask command works in an opposite manner to the number given to the `chmod` command. The "mask" serves to remove permissions as opposed to granting them. That is, the digits in the `umask` number are "subtracted" from 666 for files and 777 for directories when you are creating their initial permissions. For example, suppose you enter:

```
$umask 022
```

Now when you create new files their default permissions will be 644 (`-rw-r--r--`). When you create new directories their default permissions will be 755 (`drwxr-xr-x`). If the `umask` value were instead set to 077, your default file permissions would be 600 (`-rw-------`) and your default directory permissions would be 700 (`drwx------`).

To find out what **umask** you currently have, type:

```
umask
```

Note If you are using the Posix, Korn, or Key Shell, you can also enter the **umask** permissions symbolically. In the case above, you can type: **umask u=rwx,g=rx,o=rx**, with results as shown above for files and directory permissions. Check which shell-type you have by typing **echo $SHELL**, from you login shell, before attempting this.

Here are some examples of common settings for umask:

`umask 077` Assigns permissions so that only you have read/write access for files (read/write/search for directories) you own. All others have no access permission to your files and directories.

`umask 022` Assigns permissions so that only you have read/write access to files (read/write/search access to directories) you own. All others have read access only to your files (read/search access to your directories).

`umask 002` Assigns permissions so that only you and members of your group have read/write access to files (read/write/search access to directories) you own. All others have read access only to your files (read/search access to your directories).

Caution You should not set a **umask** value, such as 2xx or 7xx, which restricts your access permissions to your own files. A number of HP-UX utilities, such as **vi**, assume that you can always access newly-created files. Such files might include the temporary files which **vi** creates. These utilities may malfunction when used under such a restrictive **umask** setting.

If you set **umask** at a shell prompt, it will apply to shells and subshells in the current login session only. It won't apply to future login sessions. To apply a **umask** setting automatically at login, add the **umask** command to your `.profile` (Bourne, Posix, and Korn Shell users) or `.login` file (C Shell users).

For More Information ...

To get more information about the **umask** command refer to the *umask*(1) entry in the *HP-UX Reference*).

To learn more about the `.profile` and `.login` files see the *Shells: User's Guide*.

Chapter Command Summary

Table 14-3. Commands

To Do This . . .	**Type This . . .**
Display file permissions	`ll`
Numerically change file or directory permissions	`chmod` *number name*
Symbolically add or subtract file or directory permissions	`chmod` *class±permissions name*
Symbolically change (absolutely) file or directory permissions	`chmod` *class=permissions name*
Find out permissions mask setting	`umask`
Numerically change permissions mask setting	`umask` *mask_number*

A

HP-UX Quick Reference

The following table summarizes the most useful HP-UX commands. To make it easier to refer to selected commands, copy or tear out these pages and place them near your display. If you need further information on a specific command, click on ? or use the **man** command, followed by the command name.

HP-UX Quick Reference

How to use this Reference:

1. Type the commands as they are shown in the second column below.
2. Include paths with file names, if you are working with different directories.
3. Follow each command with (Return).

To Do This ...	Type This ...
Working with Directories:	
Show current working directory	`pwd`
Determine what shell you're in	`echo $SHELL`
Change directory	`cd` *directory_path*
Change to home directory	`cd`
Create a directory	`mkdir` *directory_name*
Remove an (empty) directory	`rmdir` *directory_name*
Display permissions for a directory	`ll -d` *directory_name*
Change permissions for file or directory	`chmod` *permission_nos name*
Symbolically change file or directory permissions	`chmod` *class=permissions name*
Working with Files:	
Create or reset password	`passwd` *login*
Read mail	`elm`
List files and directories in current directory	`ls`
List all files or directories, including invisible ("dot") files	`ls -a`
List files, and mark directory names with "/"	`lsf`

To Do This ...	Type This ...
Display permissions for a file	`ll` *file_name*
Create or edit a file	`vi` *file_name*
Display file contents	`more` *file_name* (`q` to quit)
Display the first 10 lines of a file	`head` *file_name*
Display the last 10 lines of a file	`tail` *file_name*
Copy a file	`cp` *file_name file_copy*
Move a file to a new file name	`mv` *old_file new_file*
Append *file1* onto the end of *file2*	`cat` *file1* `>>` *file2*
Remove *file*	`rm` *file*
Remove a directory *dir_name* and all its files	`rm -rf` *dir_name*
Check the spelling in a file	`spell` *file_name*
Print a file	`lp` *file_name*
Finding and Organizing:	
Find file(s) beginning with *x* in current- and sub-directories	`find . -name '`*x**`' -print`
Find all occurrences of *word* in all files in current directory	`grep` *word* `*`
Sort *listfile* of two-word names by last name	`sort +1` *listfile*
Display date and time	`date`
List all aliases	`alias`
Find HP-UX command information	`man` *command_name*
Determine PATH setting	`echo $PATH`

A

To Do This ...	Type This ...
System Operations:	
Clear screen	`clear`
Set command-line editor	`set -o` *editor_name*
Edit your command line (in Korn/Key/Posix Shell set for `vi`)	ESC (use `vi` commands)
Recall previous command line (with `vi` editor)	ESC `k` (back) or `j` (forward)
Execute previous command line	Return (when line is displayed.)
Set terminal type (select *term_type* from `/usr/lib/terminfo`)	`TERM=`*term_type*
List current process status and *PID's*	`ps -ef`
Kill (terminate) a process	`kill` *PID*
Create or change a password	`passwd`
Redirect input from a file to a command	*command* `<` *infile*
Connect two processes with a "pipe"	*command1* `|` *command2*

Glossary

absolute path name

The name of a file which lists all the directories leading to it, starting with root ("/") and ending with the file base name itself. If the path name indicates a *directory*, leave the trailing slash. For example, `/users/jth/`.

access permissions

File name characteristics (including *read, write*, and *execute*) which determine whether a process can perform a requested operation on the file (such as opening a file for writing). Access permissions can be changed by a *chmod*(1) command.

active window

The window in which keyboard input appears. Only one window can be active at a time. The active window is said to have the keyboard focus.

application

A program used to perform a particular task, usually interactively, such as computer-aided design, text editing, or accounting.

argument

The part of a command line which identifies what (file, directory, etc.) is to be acted upon.

ARPA/Berkeley host name

A system name assigned to each system that supports ARPA services.

ASCII

American Symbolic Code for Information Interchange

background process

A program, usually low priority, run non-interactively by the shell without terminal I/O, while other processing occupies the terminal. The "&" at the end of a command line causes that command to be run as a background process.

backup

A copy of all or part of the file system.

bit

BInary digiT

bitmap

An array of data bits used for graphic images. For HP VUE, a two-color image (foreground and background).
Also see **pixmap**.

boot

To start or activate a system.

boot ROM

A read-only memory which is incorporated into a terminal for the purpose starting the operating system, testing the terminal, and producing a standard display.

Bourne Shell

A command interpreter, invoked as `/bin/sh`. The Bourne Shell is the default shell in HP-UX.

BSD

Berkeley Software Distribution.

bus address

A number which makes up part of the address HP-UX uses to locate a particular device. The **bus address** is determined by a switch setting on a peripheral device which allows the computer to distinguish between two devices connected to the same interface.

button

A graphic element in a display that functionally represents an actual push button. It is usually accessible by mouse pointer and is used to start an action.

C

A standardized and highly-portable computer language. Also the name of the NLS default language/environment (formerly n-computer). Also the name of one of the HP-UX command interpreters, the C Shell (`csh`).

CDF

context-dependent file.

CD-ROM

Compact Disc Read-Only Memory.

CD ROM file system

A read-only memory file system on compact disk. You can read data from a CD ROM file system, but you cannot write to one.

character

An element used for the organization, control, or representation of text. Characters include graphic characters and control characters.

click

To press *and release* a mouse button rapidly.

cluster

A group of workstations connected via a LAN. One computer, the **cluster server**, performs as a file-system server for the **cluster clients**.
For more information on cluster concepts, see Managing Clusters of HP9000 Computers: Sharing the HP-UX Filing System.

cluster client

A cluster node that does not have a local HP-UX file system. Its file system resides on the cluster server. However for HP-UX 8.0, cluster clients can have locally mounted disks for local data storage.
A **client** can also refer to any process run by a server.

cluster node

Any workstation networked into an HP-UX cluster. (Also called "cnode".)

cluster server

The cluster node which acts as a file system server and operating system server for all the cluster nodes in an HP-UX cluster. Also called **cluster root server**.

cnode

Abbreviation for **cluster node**.

command interpreter

A program which reads lines of text from standard input (typed at the keyboard or read from a file), and interprets them as requests to execute other programs. An HP-UX command interpreter is called a "shell".

CPU

Central Processing Unit. Refers only to the instruction-processing module inside the computer. *See also* **SPU**.

cron

A process which executes commands at specified dates and times.

CRT

Cathode ray tube. Same as "display".

C Shell

An HP-UX command interpreter, invoked as `csh`.

current session

The HP-UX or HP VUE session to which you are logged in at a particular time.

current working directory

The directory in which relative path name searches begin. It is also called the "current directory" or "working directory", and is identified by entering the command `pwd`.

Desktop

The HP VUE workspace backdrop which allows you to place any file or directory icon directly for easy access.

device file

A file used for the computer to communicate with a device such as a tape drive or a printer.

DDS

Digital Data Storage. HP-supported "DAT" format for data storage.

dialog box

A subwindow of an application used to request information, or to display status or error conditions.

DIO

Device input/output.

directory

A table of identifiers and references (such as file names) that refer to corresponding files and items of data. Used in a typical HP-UX organizational structure to provide an organizational and logical identity for a given group of files and directories. In HP VUE, a directory is sometimes called a "folder."

double click

Pressing and releasing a mouse button twice in rapid succession. For HP VUE, "double click" an icon.

drag

Pressing and holding down a mouse button while moving the mouse pointer.

drop

Releasing an icon that has been "dragged" to a new position by release the mouse button.

drop zone

A special space in a window display which responds to an icon which has been dropped there as an object to be acted upon. For example, the "Trash

Can", on your Front Panel, is a **drop zone** which responds to file icons that you "drag and drop" there.

environment

The set of defined shell variables (some of which are PATH, TERM, SHELL, HOME) that define the conditions under which your commands run. These conditions can include your terminal characteristics, home directory, and default search path.

file access permissions

File name characteristics (including *read, write*, and *execute*) which determine whether a process can perform a requested operation on the file (such as opening a file for writing). Access permissions can be changed by a *chmod*(1) command.

File Manager

The HP VUE component that allows you to manipulate your files and directories, or to set the format and behavior of HP VUE.

fileset

Describes a logically-defined, named set of files on an update or installation tape.

file system

The organization of files on a given storage device, possibly including hierarchical directories.

filter

A command, such as `cat`, `grep`, or `sort`, that reads data from the standard input, performs a transformation on the data, and writes it to the standard output.

foreground process

The process occupying the currently active terminal I/O, which may be a window. The shell will not return a prompt until a foreground process has finished executing.

Front Panel

The window area of a default HP VUE screen which contains some accessories, such as the clock, and the control buttons for activating various functions of the workspace manager program, such as print, terminal, style manager, and trash.

group

An association of users who are all permitted to access the same set of files. The members of a group are defined in the files `/etc/passwd`, `/etc/group`, and `/etc/logingroup` (if it exists) via a numerical group ID. Users with identical group IDs are members of the same group.

group access list

The group access list is a set of supplementary group IDs, associated with a process, used in determining resource accessibility.

GUI

Graphical User Interface.

hardware installation

Includes the connection of hardware (disk drives, printers, monitors, terminals) and the physical placement of hardware in enclosures.

heterogeneous cluster

A cluster containing both Series 800 and Series 300 cluster nodes.
Also may be used to refer to a mix of other, non-HP hardware systems in the same cluster.

$HOME

The value of the environment variable representing the **home directory**.

home directory

The directory name given by the value of the shell variable HOME. This is the directory where the user starts after logging in, typically `/users/`*login*, where *login* is your login name.

home session

An HP VUE user-configured default session. Unless you specify otherwise, the home session is restored each time you log in.

homogeneous cluster

An HP-UX cluster containing only Series 300 or only Series 800 cluster nodes.

host name

Refers to a string which uniquely identifies a system in a network. There are generally different **host name** domains associated with different networks.

HP-HIL

Hewlett-Packard Human Interface Link.

HP-IB

Hewlett-Packard Interface Bus (IEEE 488 standard).

HP-UX cluster

A group of workstations connected via a LAN. One computer, the **cluster server**, performs as a file-system server for the **cluster client**. **Cluster** can also be used by itself if the context is clear. "Diskless cluster" should not be used.

HP VUE

HP Visual User Environment.

icon

A miniaturized graphic representation of a graphic object in the workspace (typically an application window). Objects can be "iconified" (turned into icons) to clear a cluttered workspace, and restored to their original appearance, as needed. Processes executing in such an object continue to execute when the object is iconified.

iconify

The act of turning a window into an icon.

IEEE

Institute of Electrical and Electronics Engineers, and the interface format which bears its name.

ITE

The Internal Terminal Emulator program, which allows a bit-mapped display to function as a standard computer terminal.

kernel

The part of the HP-UX operating system that is responsible for managing the computer's resources.

key binding

In HP VUE, association of a special key press with a Workspace Manager function. For example, pressing the special keys (Shift) + (Esc) displays the window menu of the active window.

keysh

The command for invoking a Key Shell.

Key Shell

An HP-UX shell which, as an extension of the Korn Shell, uses hierarchical softkey menus and context-sensitive help to aid users in building command lines. Invoked as `usr/bin/keysh`.

Korn Shell

An HP-UX shell, featuring command history recall and line-editing. Invoked as `/bin/ksh`.

LAN

Local Area Network.

LANG

An NLS environment variable that is used to inform a computer process of the user's requirements for "native language," "local customs," and "coded character set."

Local Area Network

The systems and/or clusters which share data, hardware, and software resources via Networking Services software.

localization

The process of providing software with the ability to support the "native language," "local customs," and "coded character set" of the user.

locally-mounted file system

A file system mounted on a disk attached to a cluster client and shared by other nodes in the cluster.

login, logout

Your login name, the name by which you are known to the workstation. This may be any group of characters, so long as it meets system rules.

Login Manager

The program that controls the initial startup of HP VUE, accepts the user's login and password, and then starts the session manager.

maximize

The function of enlarging a window to its largest size. OSF/Motif window managers, such as the HP VUE Workspace Manager, provide this function. *Also see* **minimize** and **restore.**

menu bar

An area at the top of a window that contains the pull down menus for that application.

minimize

The function of reducing a window to an icon. OSF/Motif window managers—such as the HP VUE Workspace Manager—provide this function. *Also see* **restore.**

mnemonic

A single character used as a shortcut for a command. Usually, a command's mnemonic is its initial letter. There is often a visual cue that a mnemonic exists (such as underlining or illuminating the mnemonic character).

MOTD

Message Of The Day.

mount

To add an auxiliary (removable) file system to an active existing file system.

mount directory

The directory in an existing file system that is the root directory of a mounted auxiliary file system.

multiuser state

The condition of the HP-UX operating system in which the cluster nodes (and console) allow communication between the system and all its users.

Native Language Support (NLS)

A feature of HP-UX that provides the user with internationalized software and the application programmer with tools to develop this software.

NFS

Network File Services.

NFS file system

A file system accessible over a network via the NFS Services product.

NLSPATH

An NLS environment variable used to indicate the search path for message catalogs.

node name

A unique string used to identify each node in a cluster.

object

A passive entity that contains or receives information.

In C programming, an **object** is a location in storage, sometimes called a variable.

operating system

The contents of `/hp-ux`, including the kernel, commands, input-output control, system accounting, storage assignment, and other services. *Also see* **kernel.**

OSF

Open Software Foundation.

owner

The owner of a file is usually the creator of that file. However, the ownership of a file can be changed by the superuser or the current owner with the *chown(1)* command or the *chown(2)* system call.

PAM

Personal Application Manager.

parent process

In a shell environment, an existing process which has caused a new process (a **child process** to be created.

parent process ID

A parent process identification. *Also see* **PID**.

password

An encrypted sequence of characters used by HP-UX to identify an authorized user and to permit authorized login on a system.

path name

(Two words, except as an italicized argument, *pathname*.)
A sequence of directory names, separated by slashes, which specify the location of any file or directory.

PID

Process identity (number).

pixmap

For HP VUE, a bitmapped image with more than two colors.

pop-up menu

A menu that remains invisible until a user action or some condition in the program causes the menu to display. Typically the user either presses the "menu button" on the mouse (a common default is button 3), or presses the "menu key" on the keyboard (a common default is F10).

POSIX

POrtable Systems Interface (for UN*X). UN*X standard from IEEE.

Posix Shell

POSIX-compliant version of the Korn Shell.

ppid

Parent process ID.

process

An invocation of a program. Generally, **process** refers to a program running in memory, while **program** is the code stored on disk.

process ID

A unique identification number assigned to all processes by the operating system. *Also see* PID.

pty

Pseudo-terminal.

RAM

random-access memory.

rc

Run commands.

regular expression

A string of characters that selects text.

relative path name

The name of a file, listing all the directories leading to that file in relation to the current working directory.

resource

A component of the X Window System resource data base. Resources values control many of the settings used by X Window System applications, and can be changed by users.

restore

The function of returning a window to its normal size from a *maximized* or *minimized* state. OSF/Motif window managers, such as the HP VUE Workspace Manager, provide this function.
See **minimize** and **maximize**.

ROM

read-only memory.

root directory

The highest level directory of the hierarchical file system, from which all other files branch. In HP-UX, the slash (/) character refers to the "root directory." The root directory is the only directory in the file system that is its own "parent directory."

root file system

The file system mounted on the cluster server.

root server

The node in a cluster to which the storage device containing the root file system of the cluster is physically attached. *Also* **cluster server**.

run-level

The system state determined at boot which defines, among other things, multi- or single-user status.

script

A file that contains commands that a shell can interpret and run.

scroll bar

In graphical interfaces, a graphical device used to scroll data displayed in a window. A scroll bar consists of a slider, scroll area, and scroll arrows.

SCSI

Small Computer System Interface.

server

A computer program that provides file access, login access, file transfer, printing and other services across a network. Sometimes, but not always, a server consists of a dedicated computer.

session

Generally describes the time between beginning to use an application and quitting the application. More specifically, used to describe the time between logging in and logging out.

session manager

Starts the workspace manager and other programs that were running during a previous session.

shell

An HP-UX command interpreter (Bourne, Korn, Key, Posix or C), providing a working environment interface for the user. The shell takes command input from the keyboard and interprets it for the operating system.

shell function
A function within a Korn shell script.

shell script
A file that contains commands that a shell can interpret and run. Also "shell program."

short file names
Files with names consisting of 14 or fewer characters.

shut down
To take the system from multi-user state to a state in which no processes are running, using the `shutdown` command.

single-user state
The state of a computer where there is little or no process activity and no users logged in. The system is only accessible to the current system administrator (root). This mode is brought about by execution of *shutdown*(1). Also called **single-user mode.**

source code
The fundamental high-level information (program) written in the syntax of a specified computer language. Object (machine-language) code is derived from source code.

SPU
System Processing Unit. The computer "box" minus all peripherals.

standalone
A computer which is not part of a cluster.

standard error
The destination of error and special messages from a program, intended to be used for diagnostic messages. The standard error output is often called `stderr`, and is automatically opened by the shell for writing on file descriptor 2 for every command invoked. Standard error usually appears on the display unless it is directed otherwise.

standard input

The source of input data for a program. The standard input file is often called stdin, and is automatically opened by the shell for reading on file descriptor 0 for every command invoked.

standard output

The destination of output data from a program. The standard output file is often called `stdout`, and is automatically opened by the shell for writing on file descriptor 1 for every command invoked. Standard output appears on the display unless it is redirected otherwise.

striping

The distribution of a logical disk onto two or more physical disks.

style manager

The HP VUE application that provides the ability to customize various aspects of the screen format and behavior.

SU

Super User.

super-user

Loosely synonymous with **root-user**.

swap in

The process of reading the process's image from the swap space on the disk into the computer's main memory.

swap out

The process of writing the process's image from the computer's main memory onto the swap space on the disk.

swap space

Space on the disk used for temporarily storing the process image.

system call

Invocation of a kernel process by a user program.

system name

The eight-character (or less) string which uniquely identifies a system. Usually identical with the system's host name found in `/etc/hosts`. The Internet Protocol (IP) number is sometimes used instead of a system name to identify the system.

timeout

Deactivation of a device or system after a pre-set time.

title bar

The rectangular area at the top of a window that contains the title of the window (for example, "Terminal Window" or "File Manager"). In OSF/Motif window managers, such as the HP VUE Workspace Manager, the title bar can also be used with the mouse pointer to move ("drag") the window to a new location.

Toolbox

A special File Manager view for managing applications and other software.

tree structure

The HP-UX method of organizing files and directories into a hierarchical structure. This structure looks like an inverted tree with the "root" directory at the top, descending into multiple directory/file branches that end in clusters of files.

trusted computer system

A system that employs sufficient hardware and software integrity measures to allow its use for sensitive or classified information.

uid

User ID.

user

Any person who interacts directly with a computer system.

user interface

The medium through which users communicate with their workstations. The command-line prompt is one type of interface. The graphical objects of HP VUE are another type of interface.

user ID

An integer which identifies each system user.

utility

An executable file, which might contain executable object code or a list of commands to execute in a given order (such as a shell script).

volume number

Part of an address used for devices, used to specify a particular volume on a multi-volume disk drive.

VUE

Visual User Environment.

window

A frame-defined, rectangular area of the screen used by the HP VUE to contain a particular application or a command line. Client applications can have multiple windows.

window manager

The program that controls the size, placement, and operation of windows. The window manager provides window frames for moving and resizing windows, title bars for labeling and moving windows, and icons when windows are minimized. Some window managers support multiple workspaces, such as the HP VUE Workspace Manager.

window menu

The menu in the upper-left corner of a window frame (provided by the window manager). The window menu generally contains only window manager commands for manipulating the window. However, it may also contain application-specific commands.

working directory

This is the directory in which relative path name searches begin. It is also called the current directory, or the current working directory.

workspace

The entire area of the display screen when HP VUE starts up. To switch from one workspace to another, the user chooses the corresponding workspace button in the Front Panel.

workspace manager

The program that creates and manages alternate virtual screens, called workspaces. Some window managers also take on the job of being the workspace manager, such as the HP VUE Workspace Manager.

workstation

A graphics-oriented computer, generally high-speed and high-capacity, designed for use in limited space.

Index

WIN AN HP CALCULATOR!

Your comments help us determine how well we meet your needs. Returning this card with your name and address enters you in a quarterly drawing for an HP calculator*.

Using HP-UX
B2910-90001 E0892

What operating system and hardware do you use? (Type: `uname -rvm` and write the displayed information on the following line.)

________________ ______ ________________

How much have you used this manual?

______ Extensively ______ Often ______ Occasionally ______ Not at all

`----fold here---------------------------------------------------fold here----`

Complete this section only if you have used this manual. Use the column labeled "NC" if you have no comment or opinion regarding that topic.

	Agree					Disagree	NC
The manual is well organized.	○	○	○	○	○	○	○
It is easy to find information in the manual.	○	○	○	○	○	○	○
It tells me clearly what I need to know.	○	○	○	○	○	○	○
It is easy to perform step-by-step procedures.	○	○	○	○	○	○	○
Overall, the manual meets my expectations.	○	○	○	○	○	○	○

Please take the time to describe any problems you've had or any suggestions that would improve our product/manual. Use additional pages if needed. The more specific your comments, the more useful they are to us. Thank you.

Comments:

`----fold here---------------------------------------------------fold here----`

○ **Check here if you would like a reply.**

** Offer expires 08/1/94.*

↑ *Please tape here* **Please print/type the following information** *Please tape here* ↑

Name: ____________ Telephone: ____________

Company: ____________

Address: ____________

City: ____________ State: ______ Zip Code/Country: ____________

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 37 LOVELAND, COLORADO

POSTAGE WILL BE PAID BY ADDRESSEE

Learning Products HP-UX
Hewlett-Packard Company
3404 East Harmony Road
Fort Collins CO 80525-9988

HEWLETT
PACKARD

Reorder No. or
Manual Part No.
B2910-90001

**Manufacturing
Part No.
B2910-90001**

B2910-90001