Tru64 UNIX

System Administration

Part Number: AA-RH9FE-TE

September 2002

Product Version: Tru64 UNIX Version 5.1B or higher

This manual describes the tasks you must perform to maintain the Tru64 UNIX operating system running on a workstation or server. You use UNIX commands, shell scripts, and the SysMan Menu or SysMan Station user interfaces to perform the administration tasks described in this manual.

Hewlett-Packard Company
Palo Alto, California

Contents

2 Starting Up and Shutting Down the System

3 Customizing the System Environment

4 Configuring the Kernel

5 Administering Disks

6 Administering UNIX File Systems (UFS)

7 Administering User Accounts and Groups

8 Administering the Print Services

9 Administering the Archiving Services

10 Administering the System Accounting Services

11 Monitoring and Testing the System

14 Administering Crash Dumps

A Administration Utilities

Index

Examples

Figures

Tables

About This Manual

This manual describes the tasks you perform to administer the Tru64 UNIX operating system running on an AlphaServer.

Audience

This manual is intended only for system administrators. As a system administrator, you should have knowledge of the UNIX operating system concepts and commands and the supported hardware and software configurations. You must be trained in the operational aspects of UNIX system administration and be familiar with all the procedures necessary to maintain a UNIX system for high availability. This manual is not intended to train system administrators or to plan the installation of a UNIX system.

New and Changed Features

The following changes were made to this manual:

- *Chapter 5* is a subset of the previous version of this chapter. The removed material has been incorporated into the *Hardware Management* manual.
- *Chapter 8* has been updated to focus on printer configuration using TCP/IP. This chapter also describes the updated `lprsetup` procedure and describes new `lpc` commands.
- *Section 9.7.5* has been updated.
- *Section 11.3* has been updated with a discussion of environmental monitoring from the command line, as part of HP Insight Manager, and using sensor monitoring (which is available on a limited number of recent hardware platforms).
- *Chapter 13* has been updated for discussions on starting and stopping the Event Manager, configuration for preventing missed events, remote authentication, remote logging, and the increased message capacity of event connections.
- *Chapter 14* now includes a description of the graphical user interfaces for configuring crash dumps for your needs and to save a snapshot of system memory to a dump file.

If you are updating your system from an older version of the UNIX operating system, you may want to review all the changes that were implemented in the intervening releases. You can find this information in the HTML files

provided on the Software Documentation CD-ROM, especially *New and Changed Features from Previous Releases*. In addition, the following online resource isavailable:

- Previous versions of this manual are available on the World Wide Web at: **http://www.tru64unix.compaq.com/docs/**

  See the New and Changed features section of those versions to learn the evolution of this manual.

  Also, you can obtain technical updates for any information that is not included in the documentation provided with your media.

New features are added to many of the operating system's administrative commands and utilities. Command examples and procedures throughout the manual are verified to ensure that they are correct. In several cases, the related reference pages are revised completely. Some information is relocated to reference pages to eliminate redundancy and reduce the size of this manual.

Organization

This manual is organized as follows:

Chapter 1	Describes the methods and tools that you use to perform system administration tasks.
Chapter 2	Explains how to start up and shut down the operating system. It also explains how to recover from an unexpected shutdown.
Chapter 3	Describes how to customize operating system files and operating system components to tailor the operating system environment.
Chapter 4	Describes how to configure an operating system kernel dynamically and statically.
Chapter 5	Discusses system administration tasks related to the administration of disks, including disk partitioning, disk copying, and disk monitoring.
Chapter 6	Explains how to administer the UFS file system.
Chapter 7	Explains how to administer accounts for operating system users and groups of users.
Chapter 8	Explains how to administer the print services system and configure printers.
Chapter 9	Explains how to administer the archiving services of the operating system in order to back up and restore mass storage devices.

Chapter 10	Explains how to administer the resource accounting services of the operating system.
Chapter 11	Describes the monitoring and testing utilities.
Chapter 12	Explains how to set up and administer the basic event logging services of the operating system.
Chapter 13	Explains how to set up and administer EVM, the advanced event management and logging mechanism.
Chapter 14	Explains how to set up and administer crash dumps.
Appendix A	Lists the administration utilities.

Related Documentation

The following documents provide important information that supplements the information in certain chapters:

- The *Installation Guide* and *Installation Guide — Advanced Topics* describe how to install your operating system. Several important administrative tasks, such as installing software and installation cloning, are described in detail in these manuals.
- The *Hardware Management* manual is the companion manual to this manual. The *Hardware Management* manual describes the tasks you must perform to maintain system hardware that is controlled by the Tru64 UNIX operating system.
- For important information on storage configurations, including the configuration and maintenance of storage arrays, see your StorageWorks documentation. You use StorageWorks software applications, such as the StorageWorks Command Console (SWCC) in addition to the utilities provided by the operating system. See *Related Documentation* for resources on the Web.
- The *Network Administration: Services* and *Network Administration: Connections* manuals describe how to set up, configure, and troubleshoot your network.
- The Advanced Server for UNIX (ASU) *Concepts and Planning Guide* and *Installation and Administration Guide* provide information on administering Windows domain accounts and sharing printers with PC users. These documents are supplied with the ASU software on the *Associated Products CD-ROM, Volume 2*.
- The *Security Administration* manual provides information on security that affects account management and file system sharing.
- The *AdvFS Administration* and *Logical Storage Manager* manuals provide information on advanced file systems and storage management.

- The *System Configuration and Tuning* manual provides information on system performance tuning and advanced kernel configuration.
- The *Release Notes* provide important information such as restrictions on using certain operating system features.

Many procedures described in this manual concern the administration of system hardware and peripherals such as storage devices. See the owner's manual for any hardware device, particularly if you need information on using software that is supplied with, or required to manage the device.

Use the console commands for your processor; they are documented in the owner's manual. The *Release Notes* provide information on device-specific restrictions. The following online resources are available:

- You can find hardware documentation at the Alpha Systems Technology web site, located at the following URL: **http://www.compaq.com/alphaserver/technology/index.html**
- You can find software and drivers, including Alpha firmware downloads at the following URL: **http://www.compaq.com/support/files**
- You can find general resources on AlphaServers at the following URL: **http://www.compaq.com/alphaserver/index.html**

Icons on Tru64 UNIX Printed Manuals

The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the manuals to help specific audiences quickly find the manuals that meet their needs. (You can order the printed documentation from HP.) The following list describes this convention:

G	Manuals for general users
S	Manuals for system and network administrators
P	Manuals for programmers
R	Manuals for reference page users

Some manuals in the documentation help meet the needs of several audiences. For example, the information in some system manuals is also used by programmers. Keep this in mind when searching for information on specific topics.

The *Documentation Overview* provides information on all of the manuals in the Tru64 UNIX documentation set.

Reader's Comments

HP welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

- Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32
- Internet electronic mail: `readers_comment@zk3.dec.com`

  A Reader's Comment form is located on your system in the following location:

  `/usr/doc/readers_comment.txt`

Please include the following information along with your comments:

- The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)
- The section numbers and page numbers of the information on which you are commenting.
- The version of Tru64 UNIX that you are using.
- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

This manual uses the following conventions:

MB1, MB2, MB3	MB*N* refers to the mouse button that you must press to select an item or initiate an action.
%	
$	A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells.
#	A number sign represents the superuser prompt.
file	Italic (slanted) type indicates variable values, placeholders, and function argument names.

[|] { | }	In syntax definitions, brackets indicate items that are optional and braces indicate items that are required. Vertical bars separating items inside brackets or braces indicate that you choose one item from among those listed.
. . .	In syntax definitions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
⋮	A vertical ellipsis indicates that a portion of an example that would normally be present is not shown.
`cat`(1)	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat`(1) indicates that you can find information on the `cat` command in Section 1 of the reference pages.
Ctrl/*x*	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example, Ctrl/C).
Return	In an example, a key name enclosed in a box indicates that you press that key.

1

System Administration Methods and Utilities

The operating system provides a number of methods and utilities you can use to perform administration tasks from initial configuration (setup) to ongoing maintenance and customizing your system environment. This chapter provides:

- An overview of administrative methods and utilities (Section 1.1)
- Pointers to other documentation available for the administrative utilities, such as online and Web-based help (Section 1.2)
- An explanation of the system setup utilities that are displayed automatically during the first root login to a system, that is, after a full installation (Section 1.3)
- An introduction to the different administrative methods and utilities (Section 1.4)
- A description of the administrative utilities that you launch from the Common Desktop Environment (CDE) (Section 1.5)
- An introduction to the SysMan Menu (Section 1.6)
- A description of the SysMan Menu command line interface (Section 1.7)
- An introduction to the SysMan Station (Section 1.8)
- A discussion of HP Insight Manager, which you can use to view system status, and launch the SysMan Menu and the SysMan Station from a Web browser (Section 1.9)
- Configuration information for the SysMan Menu and SysMan Station clients so that you can launch them directly from Windows on a PC (Section 1.10)
- A discussion on setting up a serial line console to access a remote system using a modem line (Section 1.11)

1.1 Overview of the SysMan Menu and Other Utilities

SysMan Menu utilities are independent of user environments, which can be as follows:

- X-compliant user environments, such as CDE.

- Microsoft® Windows® user environments running on a Personal Computer (PC), such as Windows 98 and Windows NT®.
- Web-based management using a Web browser, such as Internet Explorer, and HP Insight Manager.
- A terminal, or terminal window running under any of the previous user environments. In this case, terminal curses mode is used to display and use SysMan utilities.

For example, you can perform administrative tasks on a remote UNIX® system from a personal computer running Microsoft Windows NT using the SysMan Menu and SysMan Station clients running as Java applications. The utilities are consistent in appearance no matter what user environment is used.

Although you can use different methods to perform the same tasks, it is important to note that there may be minor differences in the options provided, depending which administrative utilities you use and how you invoke them. For example, many SysMan Menu utilities are designed to run in different user environments, and therefore contain no graphical elements such as icons. The X11-based utilities, designed to run in a windowing environment such as CDE, often contain graphical elements and support windowing features such as drag-and-drop. Examples of these are:

- Account Manager (`dxaccounts`) to administer user accounts and groups
- Kernel Tuner (`dxkerneltuner`) to customize your UNIX kernel
- File Sharing (`dxfileshare`) to share local directories and mount remote shares

Other legacy utilities, retained for backwards compatibility, are designed for use in character-cell terminals only. However, when invoked from the SysMan Menu, these utilities also run in any of the supported user environments. An example is the NIS configuration utility, `nissetup`, which appears on the SysMan Menu as `Configure Network Information Services(NIS)`.

In contrast to the X11–compliant utilities, the SysMan Menu utilities are not as highly functional and graphical. They enable you to perform the basic administrative tasks, independent of user environment. They also offer a greater breadth of administrative functions. The following usage constraints apply:

- There also may be minor differences in the appearance and layout of the SysMan Menu utilities, depending what user environment you are using. For example, invoking `Shutdown the system` when in the X11 CDE user environment displays the shutdown delay selection as a slider bar. You use the mouse button to select this bar and drag it to set a longer

time. When the same utility is invoked in a character-cell terminal, the slider bar is replaced by a field in which you type a number representing the shutdown delay time.

- There are also functional differences between administrative utilities. Some SysMan Menu utilities do not offer all the options available in the analogous command line (or X11-compliant) utility. For example, when managing user accounts, you can use the `useradd` command to set default characteristics that all newly created accounts inherit. You cannot set these characteristics from the SysMan Menu Accounts utilities. As a general rule, the SysMan Menu utilities provide the most frequently used options, while the command line interface (CLI) provides all options.

The advantage for the system administrator is that the SysMan Menu and SysMan Station provide a single consistent presentation format for administrative utilities, no matter where the administrator is located and what user environment is available. For example, you can log on to a remote UNIX system from your local PC and use the same familiar utilities to perform administrative tasks. You also can connect to any system using HP Insight Manager across the Web to view the system status and launch the SysMan Menu and the SysMan Station to perform tasks on the remote system.

1.2 Related Documentation

This guide does not document how to invoke and complete all fields in a given administration utility, but describes how you use the utility to perform administrative tasks. It includes examples of use, but not for all user environments or options. The following sections provide pointers to more detailed information on invoking and using administration utilities and methods. Documentation for the various options is provided in the following formats: reference pages, online help, and web-based help.

1.2.1 Reference Pages

Each utility has its own reference page that describes how to invoke the utility and the available options for that utility. For example, `sysman_cli`(8) describes how you invoke the command line version of the SysMan Menu data.

Reference pages also document the user environment options for a particular utility. You may be able to invoke an administrative utility in several different user environments, or you may only be able to invoke it in one.

1.2.2 Online Help

Each utility provides an online help volume that describes how you use it and gives a detailed description of the available options in a utility. Online help also identifies valid data that the user must supply, and provides reference information and definitions of terminology. The online help is accessed from a button on the first window of a utility, or from the CDE help library by invoking the library icon on the CDE front panel. System Management is the first help volume available.

In some graphical user environments, context-sensitive help is provided for the options and fields. As you move the pointer over the screen, a brief description of the screen fields or option buttons is displayed in a message field. In a `curses` user environment, a help message is displayed as you move between fields and options with the Arrow keys or Tab key. See `curses`(3) for more information.

Command line utilities have help that describes the command syntax. This usually is invoked with the `-h` or `-help` flag, or simply by entering the command without any arguments and parameters and pressing the Return key.

1.2.3 Web Based Help

When you configure and invoke the Netscape viewer as described in the *Installation Guide*, the home page defaults to the following:

`file:/user/doc/netscape/Tru64_UNIX.html`.

This page contains links to the following information:

Documentation — The online documentation for the operating system.

System Management — A link to `file:/user/doc/netscape/SYSMAN/index.html`, the HP Insight Manager Web-based Management page. The following information on administering the operating system is available from this page:

- Using SysMan Menu and the SysMan Station.
- Using an X-capable user environment such as CDE.
- Using a personal computer (PC) running Microsoft Windows. This section provides links

to the client software that you must download to your PC.

The SysMan Menu is running in Web/Java mode if it was launched from a web browser or from the SysMan Station. The SysMan tasks are running in Web/Java mode if they were launched from a web browser, SysMan Station, or from an instance of the SysMan Menu running in Web/Java mode.

To view online help for the SysMan Menu or any of the Menu tasks when running in Web/Java mode (such as from a PC), the HP Insight Manager daemon must be running on the server to which you are connecting. To start the daemon, run the following command on the server as root:

```
# /sbin/init.d/insightd start
```

You can find out which system is the server by looking at the title bar of the window from which you launched the `help` command.

The SysMan Station also requires the `insightd` daemon to display online help.

World Wide Web	Links to product information on the World Wide Web.

When HP Insight Manager is configured, you also can connect to the Web agents of any system in the local network domain that is running the HP Insight Manager agents. For example, to connect to the local host on a UNIX system, invoke Netscape and specify the following URL in the Location field:

```
http://<host>:2301
```

Where `<host>` is either the fully qualified network name of the system, such as `bender.fut.ram.ma`, or the TCP/IP address, such as `111.22.333.11`. The port is always `:2301`. See Section 1.9 for more information on configuring HP Insight Manager.

Choose HP Insight Manager Agents and then select `Summary?` to access the HP Insight Manager Web-based user guide.

There are restrictions on using HP Insight Manager, depending on your user environment. See Section 1.9 for information.

1.3 Setting Up Your System

The initial configuration of your system (setup) usually is performed as a postinstallation task and System Setup is invoked automatically at first root (superuser) login after an installation. During installation, you may have used some of the utilities documented in this chapter. You use the same utilities for initial setup as you do for ongoing maintenance and custom configuration of your system.

The System Setup utility (also known as the clipboard) is presented as a graphical user interface if your system has a graphics board and you are running an X11 user environment such as the default CDE. If you first log in at a character-cell terminal, System Setup is presented as a text interface.

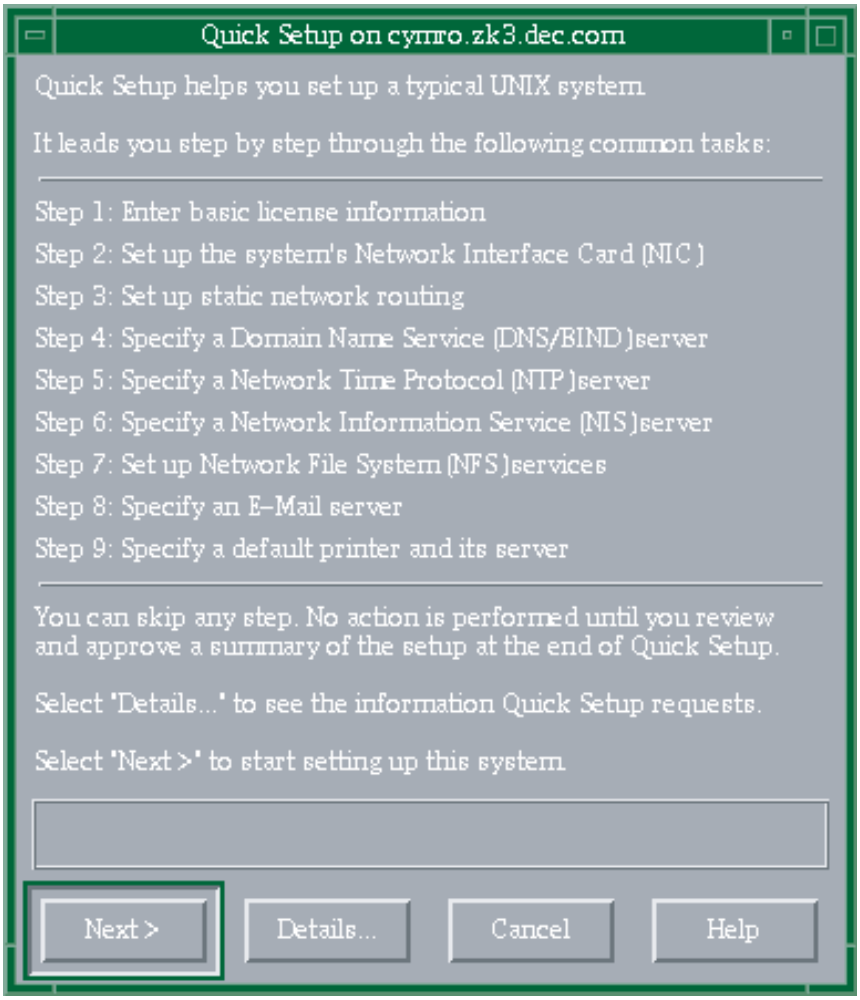
Figure 1–1 shows the System Setup in graphical format.

Figure 1–1: System Setup Graphical User Interface

You can invoke System Setup at any time to modify the existing system configuration, simply by typing `setup` at the command line, or by invoking the System Setup icon in the CDE Application Manager – System Admin folder. The following options are provided:

Quick Setup — Enables you to complete basic configuration of system services such as networking, mail, and printers. This option is useful if you want to get a system up and running quickly, leaving advanced configuration options for later. Figure 1–2 shows the initial quick setup window.

Figure 1–2: Quick Setup



Custom Setup

Enables you to run a wide range of system configuration utilities to perform all the Quick Setup tasks and run additional setup options such as custom disk configuration or set up the point-to-point protocol.

Figure 1–3 shows part of the Custom Setup graphical interface.

Figure 1–3: Custom Setup

Cloning Information — This option provides a link to information on the SysMan Menu option to clone your system configuration so that it can be applied to other systems. See the *Installation Guide — Advanced Topics* manual and `sysman_clone`(8) for more information.

See Section 1.5.2 for more information.

1.4 Administrative Methods

Most of the tasks described in this manual can be accomplished by using one or more of the following methods. Because of its versatility in different user

environments, SysMan is the recommended method of performing system administration tasks.

- The SysMan Menu

  The SysMan Menu integrates most available system administration utilities in a single menu that enables you to run the utilities from:

  – Any local or remote character-cell terminal

  – Any X11-compliant windowing environment, such as CDE

  – Microsoft Windows on a personal computer (PC)

  – The Web browser using HP Insight Manager

  See Section 1.6 for more information.

- The SysMan Station

  The SysMan Station is a graphical representation of a system (or cluster) that enables you to monitor system status from the CPU down to the level of individual system components such as disks. You also can view and monitor logical groups such as file systems or AdvFS domains and create customized views. When viewing any system component, you can obtain detailed information on its properties or launch utilities that enable you to perform administrative tasks on the component. Unlike the SysMan Menu, the SysMan Station requires a graphics capability and cannot be run from the character-cell or `curses` user environments.

  See Section 1.8 for more information.

- Graphical user interfaces in the CDE Application Manager – System_Admin

  A set of X11-compliant graphical user interfaces (GUIs) that run under CDE or other X11-compliant windowing environments. Use of the GUIs requires a graphics (windowing) terminal or workstation, and the installation of the windowing software subsets. These graphical utilities support features of the windowing environment, such as using cut-and-paste to create duplicated versions of user accounts in `dxaccounts`.

  See Section 1.5 for more information.

- Command line scripts

  For compatibility reasons, older administrative utilities have been preserved in most cases. Some command line utilities have migrated to become the new command line options. For example, the `adduser` script is still available, but it is superseded by the following utilities:

  – The SysMan Menu Accounts utilities, which provide tasks enabling you to manage users and groups in local and NIS environments.

  - The `useradd` command line utility, which you run from a character-cell terminal.
  - The Account Manager graphical user interface, available from Application Manager - Daily Admin in the CDE environment, or by invoking `dxaccounts` from a terminal window. (The interface runs in other X-compliant windowing environments)
  - The Accounts option on the SysMan Menu, available from Application Manager - System_Admin in the CDE environment, or by invoking `sysman` from a terminal window.

    You should migrate your system administration processes from the older command line scripts to the appropriate SysMan Menu method. These command line utilities have been moved to optional `OSFRETIREDxxx` subsets. See the *Installation Guide* for information on installing the retired command subsets.

- Serial line console

  In addition to networked methods of administration, the serial line console provides a dial-up facility that enables you to connect to remote systems through a modem. See Section 1.11 for more information.

- Manual file changes by editing system files (not recommended)

  Traditionally, experienced UNIX system administrators have used a combination of individual shell commands, scripts, and utilities, or simply edited the system files. Most sections of this manual describe the various system files that are updated or modified when you perform an administrative task, and you may still want to make manual changes. The use of system utilities maintains the integrity and consistency of system files such as `/etc/sysconfigtab`. We recommend that you use the appropriate utilities to update system files so that the structure of these files is preserved.

  Important considerations are:

Context Dependent Symbolic Links (CDSLs)	Many system files are now special symbolic links, created to facilitate clusters. If these links are broken, the system cannot be joined to a cluster in future without recreating the links. See Chapter 6 and `hier`(5) for more information.
Binary databases, configuration definitions	Many system components write data both to text and binary files, and their administrative utilities often recreate the binaries. Other system information often is preserved so that when you update your system it can be recovered

	and used again, saving you time and effort on administering the system.
Latent support for clusters	Individual systems are capable of being joined into clusters and many UNIX system files have been modified recently to provide latent support for clusters. For example, the `rc.config` file now has two related files, `rc.config.common` and `rc.config.site`, which can store run-time configuration variables. Using the `rcmgr` utility ensures the integrity and consistency of these files.
Update installation – preserved customized files	During an update installation, the installation process merges changed information into existing system files. The `.new..*` and `.proto..*` files may be important in this process. See the *Installation Guide — Advanced Topics* manual for more information.

1.5 Administrative Utilities Under CDE

The Common Desktop Environment (CDE) is the default X11 windowing user environment, although the utilities described in the following sections run on other X11–compliant user environments. After you complete the full installation, the System Setup graphical user interface is displayed to guide you through the process of configuring the system for initial use. From System Setup, you invoke the same graphical user interfaces (GUIs) that you use regularly to administer and customize the system. System Setup is described in Section 1.5.2.

Many of the administrative utilities that you invoke from within CDE start a SysMan Menu task option. However, some of the utilities are graphical, and either have no analogous SysMan Menu option, or offer features that only can be used under CDE. Examples are:

- CDE Setup, used to configure the CDE environment.
- Disk Configuration (`diskconfig`), an application that you use to configure disk partitions.
- Archiver (`dxarchiver`), an application used to create `tar`, `pax`, or `cpio` archives. You can use drag-and-drop to easily add folders to an archive.

Under CDE, The GUIs are located in the Application Manager, which is the tool drawer option on the CDE front panel, as shown in Figure 1–4. The icon

next to the tool drawer only appears on the CDE front panel for the root login and is used to invoke the SysMan Station as described in Section 1.8.

Figure 1–4: CDE Tool Drawer and SysMan Station Icons

If you are using an X11–compliant user environment other than CDE, invoke the individual GUIs from the command line as shown in the following examples:

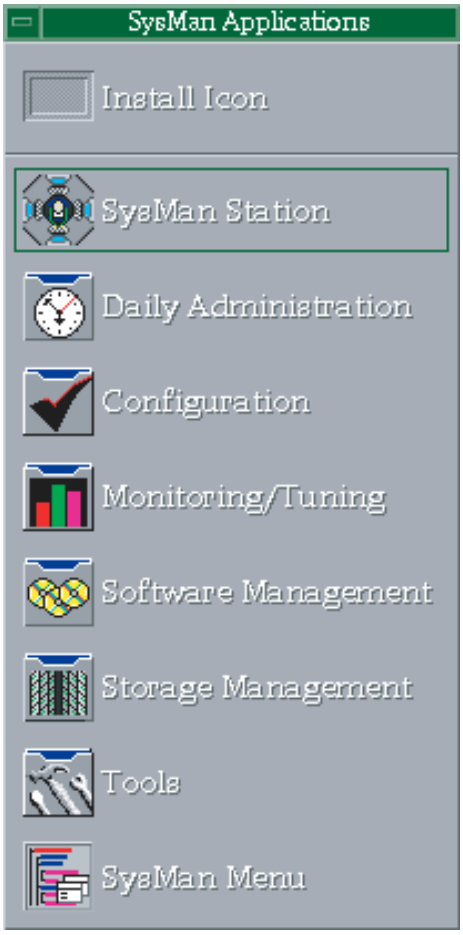
```
# /usr/sbin/X11/dxaccounts

# /usr/sbin/X11/dxarchiver
```

1.5.1 Accessing SysMan Under CDE

In CDE, certain SysMan Menu utilities are available in the Application Manager folder, which you can access as follows:

1. From the CDE Front Panel, select the arrow for the SysMan Applications panel. You can see this arrow above the icon for the SysMan Station, shown in Figure 1–4. When you select this arrow, the panel appears as shown in Figure 1–5.

Figure 1–5: SysMan Applications Panel



From this panel you can select one of the following icons, to launch a utility or open a folder containing more administration utilities:

- Launch the SysMan Station, which is described in Section 1.8. This icon appears on the front panel of a root login to CDE, as shown in Figure 1–4.
- Select a folder icon, such as Configuration to open the applications folders, which contain utilities described in Appendix A.
- Launch the SysMan Menu.

2. From the CDE Front Panel by selecting its tool-drawer icon, shown in Figure 1–4. When the top-level folder is opened, double click on the `System_Admin` group to access `System Setup`, the `Welcome to SysMan` online help volume, and the five utility groups. See Section 1.5.2 for more information.

Online help is available for the SysMan Menu utilities without actually running any utility. Select the `Help Manager` icon on the CDE front panel to invoke the online help browser. The browser includes help volumes for CDE, the CDE Desktop, and System Management. You also can customize your CDE workspace with the `Create Action` utility in the `Desktop_Apps` folder. Customized icons enable you to start SysMan applications directly from the workspace. See the *CDE Companion* manual for more information.

In other X-Windows environments, the SysMan utilities can be invoked from the command line. See `sysman_intro`(8) for a list of the utilities. This reference page also describes how to invoke the online help browser in graphical environments other than CDE. The SysMan Station icon also is located on the CDE Front Panel on the root user display.

More information is available from these reference pages:

`sysman`(8)	Describes the SysMan Menu and explains how to invoke it for various environments. See Section 1.6.
`sysman_station`(8)	Describes the SysMan Station and explains how to invoke it. See Section 1.8.
`sysman_cli`(8)	Describes the command line option for SysMan Menu, and defines the command options. See Section 1.7.

1.5.2 System Setup

System Setup guides you through the process of configuring the system for initial use. System Setup is a graphical representation of a clipboard that contains an icon for each configuration application. After the initial root login following a full installation, System Setup is invoked automatically, prompting you to complete system configuration tasks. The initial window contains two options, Quick Setup and Custom Setup.

1.5.2.1 Quick Setup

This option provides a step-by-step guide (or wizard) that navigates you through a typical system configuration. Use Quick Setup to perform a basic configuration, which may be all that is required for some systems. You can perform any advanced or site-specific configuration tasks at a later time using the Custom Setup.

The Quick Setup wizard guides you through the following tasks:

- Entering your software licenses (PAKs)
- Configuring the network interface card (NIC)

- Configuring static network routing
- Specifying the following networking services and naming servers:
  - Domain Name Service (DNS, formerly BIND)
  - Network Time Protocol (NTP)
  - Network Information Service (NIS, formerly YP or Yellow Pages)
  - Network File System (NFS)
  - Electronic mail server
- Configuring a default local or remote printer and server

You can skip any options that you do not require, details of which are provided later in this section.

1.5.2.2 Custom Setup

This option invokes a version of System Setup that contains an icon for each configuration application. You can select only the options you require for your site-specific configuration or custom configuration, for example configuring a system as a server. Not all configuration applications are available on all systems. The file `/etc/checklist.desc` contains a list of configuration applications.

When you select an icon, the appropriate SysMan Menu utility, X11–based GUI, or character-cell script is invoked. The following list describes the available utilities:

License Manager

Invokes the License Manager (`dxlicenses`) , which enables you to register the Product Authorization Keys (PAKs or licenses) for the operating system and any layered software applications. Paper copies of software licenses are provided with the product media. See `dxlicenses`(8) and `lmf`(8), and the *Software License Management* manual for more information.

Disk Configuration

Invokes Disk Configuration (`diskconfig`), which enables you to configure and administer disk devices on the system. See `diskconfig`(8) and `disklabel`(8), and the *Hardware Management* manual for more information.

Network Configuration Step By Step

Invokes the SysMan Menu Network Setup Wizard, which is a guide that leads you through the process of configuring and administering networking components on the system. See `sysman`(8) and `network_manual_setup`(7), and the *Network Administration:*

Connections manual. The following configuration options are presented:

- Configuring network interface cards (NIC)
- Setting up static routes and configuring the `/etc/routes` file
- Setting up routing services – `gated`, `routed`, or an IP router
- Set up remote who services (`rwhod`)
- Set up a DHCP server (`joind`)
- Specifying the contents of the `/etc/hosts.equiv` file
- Specifying the contents of the `/etc/networks` file

In addition to the options offered in the Network Setup Wizard, you also may need to set up other options, such as NTP, depending on your site-specific networking requirements. See the *Network Administration: Connections* manual and the *Network Administration: Services* manual for more information.

DNS (BIND) Configuration

Selecting `Configure system as a DNS client` invokes the DNS Client Configuration utility, which enables you to configure the domain name server (DNS). See `bindconfig`(8) and `network_manual_setup`(7), and the *Network Administration: Connections* manual and the *Network Administration: Services* manual for more information.

NIS Configuration

Invokes the `nissetup` script, which enables you to configure NIS, the network information service. This is also known as `ypsetup`. See `nissetup`(8) and `network_manual_setup`(7), and the *Network Administration: Connections* manual and the *Network Administration: Services* manual for more information.

NFS Configuration

Invokes the SysMan Menu and presents the `Network File Systems (NFS)` utilities, which enables you to configure and administer NFS components on the system. See `sysman`(8) and `nfs_intro`(4), and the *Network Administration: Connections* manual and the *Network Administration: Services* manual for more information.

File Sharing

Invokes the `dxfileshare` option, which enables you to access and share file systems. See `dxfileshare`(8) and the *Network*

Administration: Connections manual and the *Network Administration: Services* manual. See Chapter 6 for more information on file systems.

NTP Configuration

Invokes the SysMan Menu `Network Time Protocol Configuration` option, which enables you to configure network time. See `sysman`(8), `ntp`(1), and `ntp_intro`(7), and the *Network Administration: Connections* manual and the *Network Administration: Services* manual for more information.

PPP Configuration

Invokes the SysMan Menu and presents the `Serial Line Networking` options, which enables you to configure options and secrets files for the point-to-point protocol (PPP). See `sysman`(8), `ppp_manual_setup`(7), and `pppd`(8), and the *Network Administration: Connections* manual and the *Network Administration: Services* manual for more information.

SLIP Configuration

See the entry for PPP and `startslip`(8) for more information.

Account Manager

Invokes the Account Manager (`dxaccounts`) GUI, which enables you to create user accounts and manage groups for both UNIX and Windows NT domain users on client PCs. See `dxaccounts`(8) and `adduser`(8), and Chapter 7 for more information.

Mail Configuration

Invokes the Mail Configuration utility, which enables you to configure the system to send and receive electronic mail. See `sysman`(8), `mail_intro`(7), and `mailconfig`(8), and the *Network Administration: Connections* manual and the *Network Administration: Services* manual for more information.

LAT Configuration

Invokes the `latsetup` script, which enables you to configure the Local Area Transport service. See `latsetup`(8) and `lat_intro`(7), and the *Network Administration: Connections* manual and the *Network Administration: Services* manual for more information.

UUCP Configuration

Invokes the `uucpsetup` Connections Configuration script, which enables you to configure UNIX to UNIX connections and modems. See `uucpsetup`(8) and `uucp_intro`(7), and the *Network Administration: Connections* manual and the *Network Administration: Services* manual for more information.

Printer Configuration

Invokes the SysMan Menu Configure line printers option, which enables you to configure local and remote printers. See `sysman`(8), `printconfig`(8), and `lprsetup`(8), and Chapter 8 for more information.

Security Configuration

Invokes the SysMan Menu `Security` utilities, which enable you to configure base or enhanced security. See `secconfig`(8) and the *Security Administration* manual for more information.

Audit Configuration

Invokes the SysMan Menu `Security` utilities, which enable you to configure the audit subsystem. See `auditconfig`(8) and the *Security Administration* manual for more information.

DOP (Division of Privileges)

Invokes the SysMan Menu option `Configure Division of Privileges (DOP)`, which enables you to assign privileges to nonprivileged users so that they can run utilities that usually are run only by the root user. See `dop`(8) and `sysman`(8) for more information.

Prestoserve I/O Acceleration Configuration

Invokes the `prestosetup` script, which enables you to configure Prestoserve. See `presto`(8) and `presto_setup`(8), and the *Guide to Prestoserve* for more information.

GUI Selection

Enables you to configure the display manager to CDE or `xdm`.

ATM

Invokes a script that enables you to configure Asynchronous Transfer Mode (ATM).

HP Insight Manager

Invokes a utility that you use to enable and configure the HP Insight Manager.

You do not need to use all the options presented on System Setup, and you can opt to defer any option to a later time. If you choose to defer any configuration options and exit from System Setup, you need to invoke System Setup manually from the Application Manager – System Admin folder, from the SysMan Menu, or from the command line as follows:

```
# /usr/sbin/sysman
# /usr/sbin/checklist
# /usr/sbin/setup
```

1.6 The SysMan Menu

SysMan integrates most system administration utilities and makes them available under several different user environments. You can access utilities from the SysMan Menu, a hierarchical, task-oriented menu interface.

All the tasks in the SysMan Menu can be performed from an X11-capable display, a personal computer running Microsoft Windows, such as Windows NT Version 4.0, or a character cell terminal. There are several ways to start the SysMan Menu:

To start the SysMan Menu from a CDE desktop:

- Log in as root and choose the SysMan Menu icon from the CDE front panel's SysMan Applications panel.
- Choose the SysMan Menu icon from the System Management group in the Application Manager.
- To start the SysMan Menu from a command prompt in a terminal window, enter the following command:

  ```
  # /usr/sbin/sysman
  ```

- To start the SysMan Menu from the SysMan Station, select the system icon in a view window and then choose `SysMan_Menu` from the SysMan Station Tools menu.

You can start a specific task directly from the command line using its name in the menu or its accelerator, which is a unique keyword for each option in the sysman menu. For example, to run the task that invokes the menu option `Configure Division of Privileges (DOP)`, use its accelerator `dopconfig` and enter the following command at the system prompt:

```
# /usr/sbin/sysman dopconfig
```

Use the following command to obtain a complete listing of the available tasks and their accelerators.

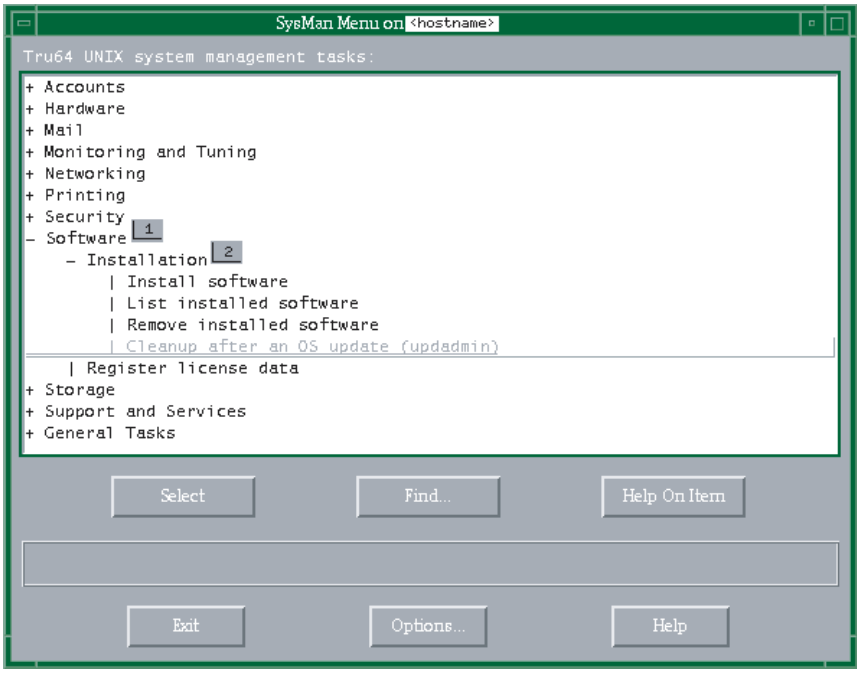
```
# /usr/sbin/sysman -list
```

The SysMan Menu contains a text list of options organized in a hierarchy (or tree). Each option appears as a branch on the tree, leading to suboptions, which may be further branches or end in a task. You can collapse or expand each option if suboptions are available, as indicated by a character preceding each menu item. The plus sign (+) indicates that further menu items are available; the minus sign (-) indicates that the branch is fully expanded.

Tasks at the end of a branch are preceded by a vertical bar (|) indicating that no further expansion of the branch is possible and you only can select a task to invoke an administrative utility.

Figure 1–6 shows the SysMan Menu invoked in the CDE user environment. The contents of this menu may be different on your system:

Figure 1–6: The SysMan Menu



As shown in Figure 1–6, the `Software` branch (label 1) is expanded fully, showing the `Installation` branch and the `Register license data` task. The `Installation` branch (label 2) contains several tasks such as `Install software` and `List installed software`. When you select a task, the appropriate utility is invoked.

How you move through and select menu items or invoke tasks is dependent on the user environment that you are using such as a `curses` terminal or a windowing environment. When using a terminal, you use the arrow keys or Tab key to move around the menu, highlighting options and buttons as you move. Use the Enter key to select an item, which expands a branch or select a task to invoke the associated utility. When using a mouse in a windows environment, you can move the pointer to a branch or task and double click MB1 to expand a branch or select a task and invoke the associated utility. See the online help for detailed instructions on navigating through the utilities. The following options appear on the SysMan Menu:

Select	Chooses the highlighted item. Selecting a branch expands or contracts it. Selecting a task invokes the associated utility.
Find...	Invokes the search window, enabling you to search on a keyword and find associated tasks.
Help On Item	Invokes context-sensitive help on any branch or task.
Exit	Closes the SysMan Menu window.
Options...	Provides options for configuring the SysMan Menu display, such as displaying the accelerators.
Help	Invokes general help on the SysMan Menu.

Context-sensitive help also is displayed in the pane located between the two rows of buttons. This online help describes the content of the window as you move the mouse pointer or use the Tab key to move to an item. Selecting a task invokes its associated utility in a format that is most appropriate for your current user environment, such as the X11–compliant windowing environment or `curses` format in a character-cell terminal.

More information is available in `sysman`(8) and in the online help. See the tables in Appendix A for information on related utilities.

1.7 Using the SysMan Command Line

The `sysman -cli` utility is a command line alternative to the SysMan Menu, which enables you to implement SysMan Menu tasks from the command line, view SysMan data, or write scripts to customize your configuration tasks. When you set up different parts of the system, such as configuring the network using SysMan Menu tasks, you are manipulating system configuration files such as `/etc/rc.config.common` or `/etc/hosts`. The

sysman -cli utility enables you to view and manipulate entries in these files directly from the command line or from within a shell script.

You must have root privileges to use the sysman -cli options, although unprivileged users can use it to view system setup data. See dop(8) for information on using the division of privileges (DoP) utilities to enable nonroot users to become privileged users of SysMan tasks.

A brief introduction to the many features of the sysman -cli utility is presented here. See sysman_cli(8) for a complete list of options and flags. A set of shell script examples is provided in /usr/examples/systems_management/sysman_cli. Some command line examples follow.

You can use the sysman -cli command to display all the manageable components in the Menu. For example, the following command is used to list the main components in the SysMan Menu hierarchy:

```
# sysman -cli -list components

Component(s):
  account_management
  atm
  auditconfig
  bindconfig
  bttape
  ciconfig
  clsschl
  doprc
.
.
.
networkedSystems
.
.
.
```

The following command displays the groups included in the networkedSystems component:

```
#  sysman -cli -list group -comp networkedSystems
Component: networkedSystems
  Group(s):
    hostEquivalencies
    hostEquivFileText
    hostFileText
    hostMappings
    joinMappingService
    componentid
    digitalmanagementmodes
```

The following command displays the current data values in the hostMappings group of the component networkedSystems. This data is the content of the /etc/hosts file.

```
# sysman -cli -list values -group hostMappings /
-comp networkedSystems
Component: networkedSystems
  Group: hostMappings
    {} {} 127.0.0.1 localhost
    argnot {local host} 16.140.112.139 argnot.xxx.yyy.com
    jason server 16.140.112.3 jason.xxx.yyy.com
    fleece {backup server} 16.140.112.28 fleece.xxx.yyy.com
    {} {} 150.2.3.4 newshst.pubs.com
```

For every option in the SysMan Menu, the sysman -cli command lets you view and manipulate system data without invoking the utilities. For example, the following command shows how you can remove a host from the /etc/hosts file:

```
# sysman -cli -delete row -group hostMappings /
-comp networkedSystems

Please enter key 1 [systemName]: newshst.pubs.com
Please enter key 2 [networkAddress]: 150.2.3.4
```

You are prompted to enter key data that enables the utility to identify the correct entry in the /etc/hosts file. Because the SysMan Menu options sometimes work on data that is stored in tables, you need to identify the correct row in the table to delete or modify. Every row has some unique identifiers, called keys, which you must specify with the sysman -cli command option. If you do not supply the keys, you are prompted to enter them. The following command shows how you determine the keys for a particular table:

```
# sysman -cli -list keys -group hostMappings -comp /
networkedSystems


Component: networkedSystems
  Group: hostMappings          Keys: systemName,networkAddress
```

You also can use sysman -cli commands to add or remove user data entries from the system data files that are updated by the SysMan Menu. For example, the following command adds a mail user interactively:

```
# sysman -cli -add row -comp mailusradm -group mailusers


Attribute Name: user_name (key attribute)
Attribute Description: user name
```

```
Attribute Type: STRING(8), Default Value:
Enter Attribute Value: davisB

Attribute Name: nis
Attribute Description: NIS User
Attribute Type: INTEGER, Default Value: 0
Enter Attribute Value ( to use default): 1

Attribute Name: mail_type (key attribute)
Attribute Description: mail user type
Attribute Type: INTEGER ENUM /
   { 0=Local/pop, 1=Secure Pop, 2=IMAP, 3=Secure IMAP }, /
Default Value: 0
Enter Attribute Value ( to use default): 2

Attribute Name: acl
Attribute Description: acl list
Attribute Type: INTEGER ENUM /
   { 0=all, 1=read, 2=post, 3=append }, Default Value: 0
Enter Attribute Value ( to use default): 0

Attribute Name: quota
Attribute Description: user name
Attribute Type: STRING(8), Default Value:
Enter Attribute Value:

Attribute Name: passwd
Attribute Description: password
Attribute Type: STRING(20), Default Value:
Enter Attribute Value: change_me

Attribute Name: orig_mailtype
Attribute Description: original mail user type
Attribute Type: INTEGER ENUM /
  { 0=Local/pop, 1=Secure Pop, 2=IMAP, 3=Secure IMAP }, /
Default Value: 0
Enter Attribute Value ( to use default):

#:
```

You also can enter the command as a single line, specifying all attribute values as follows:

```
# sysman -cli -add row -comp mailusradm -group mailusers /
  -data "{davisB} {1} {2} {0} {0} {pls_chg} {1}"
```

1.8 The SysMan Station

The SysMan Station enables you to monitor a system, group of systems, or an entire cluster and administer system resources. You also can launch the SysMan Menu or invoke utilities directly from the Tools menu, or by selecting the icon representing a system component, and pressing MB3 to display a menu of options that apply to the selected device. Unlike the SysMan Menu, the SysMan Station is a highly graphical interface, and only can run in a windowing user environment such as CDE or Microsoft Windows.

Note

You only can connect between compatible server and client versions of the SysMan Station. An attempt to connect to an incompatible server, results in an error message or dialog similar to the following:

```
System Management Server on host host name running version N,
This client running incompatible version N
```

Upgrade your client software to the appropriate version by downloading it from the server.

This section provides a brief introduction to the main features of the SysMan Station, including customized views. See the online help for SysMan Station for more information.

To start the SysMan Station fom CDE:

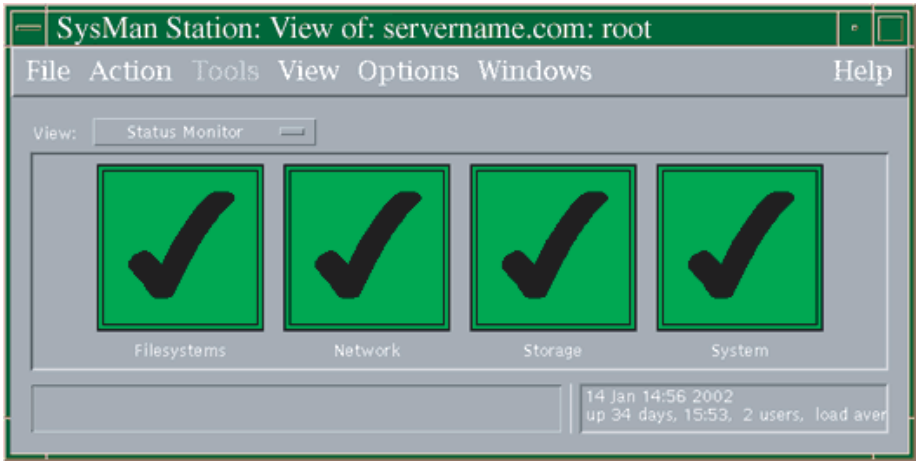
1. Log in as root and select the SysMan Station icon from the CDE Front Panel or from the SysMan Applications subpanel. (This assumes the default CDE configuration, where the SysMan Station icon appears on the Front Panel under the SysMan Applications subpanel.)
2. Choose the SysMan Station icon from the System Management group in the Application Manager.

To start SysMan Station from the command prompt, enter:

```
#sysman -station
```

After invoking SysMan Station, you are connected to the local host. The main SysMan Station window appears similar to the example shown in Figure 1–7, except that the default display shows the Filesystems..., Network..., Storage, and System... options that can be monitored. These options are known as attention groups.

Figure 1–7: SysMan Station Main Window



You can obtain event data for any of these groups by moving the pointer to an attention group, and double—clicking MB1. A window displaying a list of events is displayed.

The SysMan Station is a graphical representation of the system, in a hierarchical (tree) structure. For example, in the Storage option, you can view all disks on all buses for all processors on the system. You can select a specific device to monitor, and invoke utilities to administer that device. You can display many details (properties) of individual devices. SysMan Station also enables you to create a customized view of a system or an attention group such as storage devices. You can launch your custom views quickly and verify device status.

The main window of SysMan Station provides the following features:

Status	The Status pane lets you monitor attention groups. Status options are described in Section 1.8.1
Views	The Views pane lets you select a particular view of system components. View options are described in Section 1.8.2. This pane also displays any customized views that you create with SysMan Station.
Menu	Menu options lets you change views or select tasks. These options are described in Section 1.8.3. That section also contains brief instructions on saving customized views.

1.8.1 Using SysMan Station Status Options

When you invoke the SysMan Station, the Status pane displays a large check mark icon if the status of the attention group is normal. If the status degrades, the icon changes color, becoming a cross (X) on a red background to indicate a serious problem. These icons also enable you to instantly display any system events posted by any component in the attention group.

The default attention groups that you can monitor are:

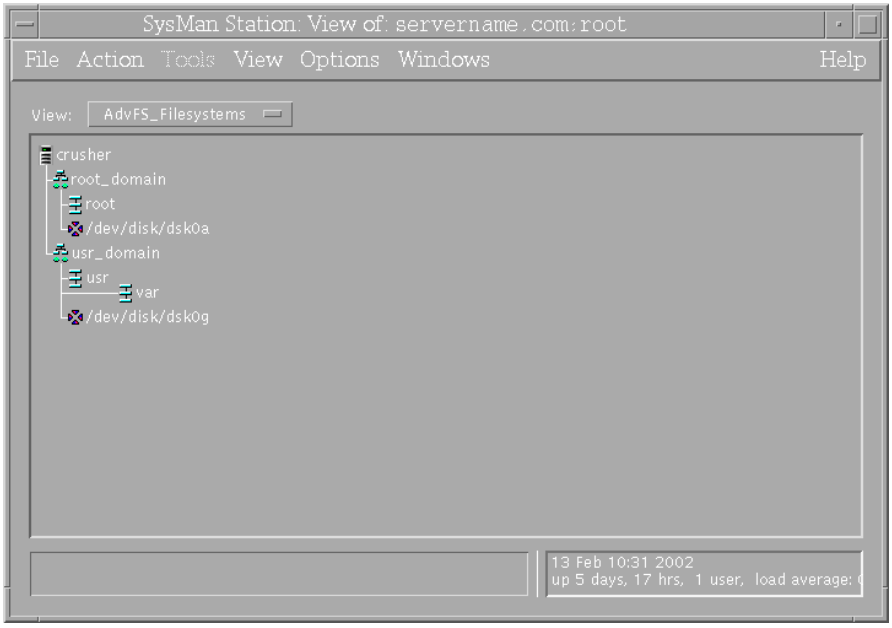
File systems	Any UFS file systems or AdvFS domains.
Network	The network and devices connected to the local host, such as `tu0`.
Storage	Storage devices connected to buses and device interfaces, such as `floppy`, the floppy drive unit that is connected to an `fdi` interface such as `fdi0`.
System	The events associated with the system components.

1.8.2 Using SysMan Station Views

The Views option menu provides a list of attention groups that can be displayed. You can select any menu option to display the Status monitor or a window showing the hierarchical structure of the group in the Views Pane. These groups are:

AdvFS_Filesystems	A view of all AdvFS domains. Figure 1–8 shows a typical AdvFS domains view on a small single-disk system.

Figure 1–8: AdvFS_Filesystems View



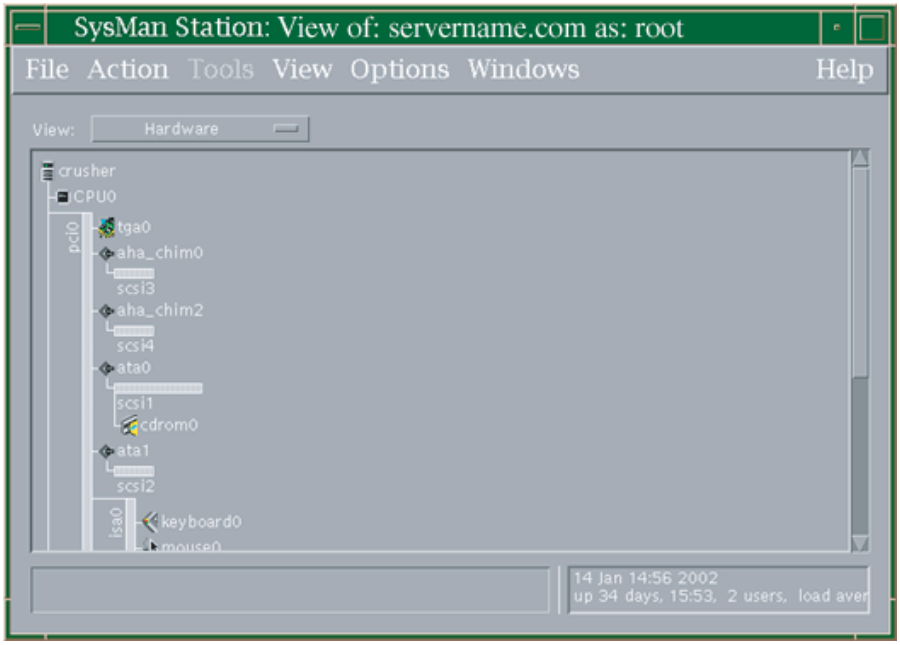
Hardware

Displays a view of all devices, from the CPU down to individual disks.

Figure 1–9 shows a typical hardware view on a small single-processor system.

In the previous example, you can see the system buses, and the various devices attached to a bus, such as the CD-ROM reader `cdrom0`.

Figure 1–9: Hardware View



Mounted_Filesys-tems	Displays a view of the file systems that are currently mounted, in a similar format to the AdvFS_Filesystems view.
Physical_Filesystems	Displays a view of all (UFS, AdvFS) file systems available, in a similar format to the AdvFS_Filesystems view.

You can customize views and save them so that you only monitor those parts of the system that are most important to you, or run applications to administer the components displayed in a view. When you customize a view, you have the opportunity to save it, and assign it an icon as described in Section 1.8.3.

In any of the system component screens, you can click MB1 on any component to select individual system components and expand or collapse sections of the display hierarchy. On selecting a component, MB3 displays a menu that contains one or more of the following options (depending on whether an option applies to the object that is selected):

Display hierarchy functions:

Expand and Contract	Display or remove the subcomponents under a component. For example, select the Expand option when selecting a SCSI bus, and all the attached devices are displayed. Select Contract to remove the displayed devices.
Hide and Unhide Children	Allows you to prevent some components and their subcomponents from being displayed, or to reveal hidden components. For example, select the Hide option when selecting a PCI bus such as `pci0`. All the devices attached are hidden. This means that you cannot display the devices by double clicking MB1 on the bus or by selecting the Expand menu option. Select Unhide Children to enable the display of the PCI bus devices.
Available SysMan Menu Utilities	Displays any administration or configuration utilities that can be launched for a component. For example, you can select a disk device, and launch the disk configuration utility.
Properties	Additional detail about the characteristics and current configuration settings for the selected device.

Options are dimmed when they are unavailable.

1.8.3 Using SysMan Station Menu Options

The main window of the SysMan Station offers the following pull-down menus and options, provided to enable keyboard selection rather than using a mouse:

File	This menu contains options to close the SysMan Station and exit, or to connect to another system.
Monitoring	This menu enables you to customize the Status view by removing an entire attention group, such as the Filesystems... attention group.
Options	This option enables you to further customize SysMan Station by selecting the initial window.

Windows	This menu enables you to cycle between the different displayed views.

You are prompted to save your custom view before you exit SysMan Station. Then you can assign a name and an icon to the custom view. When you next invoke SysMan Station, your custom view is added to the Views pane.

The component views provide pull-down menus of the following options:

File	Provides options to print the current screen, create a new connection, close the current window and to exit from SysMan Station.
Action	Offers options to change the grouping of components and the default appearance of displays, such as Expand and Hide.
Tools	Provides a launch point for any SysMan Menu utilities that are applicable to the selected component. The content of this window varies, depending on the type of component or device that is selected. The menu is blank if no utilities are applicable to the component, or if nothing is selected.
View	Allows you to control the current system view, and switch or cycle between views.
Options	Allows you to control the appearance of the views, such as the icon size.
Windows	Allows you to invoke other windows, such as the main window.

You are prompted to save your custom view before you exit SysMan Station. Then you can assign a name and an icon to the custom view. When you next invoke SysMan Station, your custom view is added to the Views pane.

See Section 1.10 for information on installing the SysMan Station under Microsoft Windows.

1.9 HP Insight Manager

HP Insight Manager is a Web-based management utility that enables you to look across a heterogeneous computing environment and access information about any device connected to the network. Devices can be

computer systems, networked printers, or network components such as routers. You can obtain information about the configuration of systems and their components or peripherals and, in some cases, perform certain administrative tasks such as asset management, asset security, work load management, and event management.

In its present implementation, HP Insight Manager provides a consistent wrapper for SysMan and other UNIX based utilities, enabling you to manage supported systems from a Web browser. On a PC or server running Windows NT, you can view details of devices and invoke administrative tasks. On a UNIX system, you can use HP Insight Manager to view details of devices, but you must invoke the SysMan Menu or SysMan Station to perform administrative tasks.

HP Insight Manager features are fully implemented in some operating environments, but are not yet implemented in others. This means that you can use many features on Windows NT systems, but you cannot use certain features onTru64 UNIX.

The main server component of HP Insight Manager is HP Insight Manager, a software console that provides full administrative services for Windows NT. The console communicates with any device in the local area network or domain that is running the agent. In this context, a device is any entity connected to the network. It can be a computer system with all its peripheral devices, a networked printer, or a router. Any network entity that has an address and can run the agents can communicate with the XE server, although some devices may require additional hardware.

A device must have an operating environment that is recognized by HP Insight Manager so that you can manage it using the web browser. Such an operating environment must be able to communicate device information to the WBEM network, and to receive and execute instructions sent from other (authorized) devices in the WBEM network.

The operating environments must be able to run HP Management Agents, which communicate with each other using a standard protocol. Devices, and their operating environments, provide information about hardware and software status using a data model, such as a Management Information Base (MIB) and Simple Network Management Protocol (SNMP). These can be thought of as a database of objects, with attributes and values, representing the manageable components of a device.

HP Insight Manager uses its standard protocol to poll a device for such data, and present it to the user in a consistent format, no matter how different the database. It is this standard protocol that puts a consistent wrapper around the device data that can be obtained (or manipulated).

In an environment consisting of client PCs and UNIX servers or Windows NT servers, you can use HP Insight Manager as your common interface to administrative tasks. For example, as an authorized (root) user working at your PC, you can invoke HP Insight Manager to view the general system status of an AlphaServer running the UNIX operating system, then to examine the specific status of a peripheral, such as the status of file systems on a disk. You also can launch a SysMan Menu task to perform operations on that file system.

You use HP Insight Manager by connecting your Web browser to a port on any system in the local area network that is running the agents. For example, if your UNIX system has the host name and address of `trout.cu.da.com`, enter the following URL in the Location (or address) field of the browser:

```
HTTP://trout.cu.da.com:2301
```

You also can specify the TCP/IP address, such as `20.111.333.10` in place of the host name and address. After you connect to a system, you can view the local system, status, or select other hosts on the local network. You also can connect to another host by selecting its address from the list of local devices.

See `insight_manager`(5) and the *HP Management Agents for AlphaServers for Tru64 UNIX Reference Guide*, which is accessible from the HP Management Agents for Tru64 UNIX home page **`http://tru64unix.compaq.com/cma`** for more information on configuring and using the Management Agents.

1.10 Using SysMan on a Personal Computer

In addition to using Java applets as described in the `insight_manager`(5), you also can install SysMan clients on a PC and launch them from an icon on the desktop or from within the Start menu.

This feature supports Microsoft Windows, MacOS, and Linux. Full information on this feature, together with an address from which you can download the required software, is provided in a Web page available from the UNIX system. This page is located at `http://<host>:2301/SysMan_Home_Page`, where *<host>* is the host name and address or the TCP/IP address. The procedure is as follows:

1. Use the Netscape Web browser on the PC to launch the `.../SysMan_Home_Page` page. A link to this page is provided on the default UNIX home page, by selecting the Tru64 UNIX SysMan icon.
2. Scroll down the page until you reach the section titled *PC SysMan Client Software*.

Verify the requirements and restrictions, noting any requirements for your client system. For example, you must be using the Internet Explorer Web browser on the client system.

3. Download the requisite software.
4. You are prompted to either save the kits to a location on your client system, or run them directly. The latter option begins installation and configuration of the software, using the typical Windows installation process. For example, you are prompted for a location for the installed software. You can create shortcuts in the existing Program group, a new Program group, in the Start menu, on the desktop, or in a folder of your choice.
5. When the installation process is complete, SysMan Station and SysMan Menu are listed as Java applications in the location you chose. Launch the required application.
6. When you launch either application, a dialog box opens, giving you the following connect options:

Host name	Enter the name and address or TCP/IP number for the host that you want to work on. The local host is displayed by default.
Login as...	Select whether you want to log in as yourself, or as a new user. For example, if you are logged in to your client system as yourself, you may need to connect as new user root in order to perform privileged tasks on the host.
Set X/Motif® display	Check this box and specify a display address if you want to redirect the output display.

When you select OK, the application window opens (the time to start up depends on the current network speed and traffic). Then you can use the SysMan Station or SysMan Menu as described in the preceding sections.

1.11 Setting Up a Serial Line Console

You can manage remote systems through a modem connection. A serial line console enables you to connect a local terminal to the remote system console through modems attached to your local system and to the communications port `COMM1` of the remote system. The local system can be any terminal or terminal emulation device that enables a modem connection such as a dumb terminal, an X terminal window, or a personal computer (PC). To perform

administrative tasks, you must be able to log in as root (or as an account with administration privileges).

This connection is referred to as the console port. The terminal connection supports a limited set of communication rates up to 57,600, depending on the console firmware supported by your processor. Currently, this feature is only available on systems that support modems as console devices, such as the AlphaServer 1000A. See your system hardware documentation to find out if your system has such capabilities.

The console port enables you to do the following:

- Connect to a remote system using a utility such as `tip`, `telnet`, or a PC terminal emulation utility
- Remotely boot or shut down a system and observe all the boot messages
- Start the kernel debugger and observe debugging messages
- Perform any system administration tasks using commands and utilities

Running the Environment Configuration Utility (ECU) on the remote system causes the modem to disconnect. For this reason, you should use the ECU to complete any environment configuration before setting up and using a modem as a console device.

1.11.1 Setting Up a Console Port

The following sections provide an overview of the steps required to set up a serial line console port and set up the remote modem for dial-in. It is assumed that your local (dial-out) modem is installed and configured for use already.

1.11.1.1 Connecting the Modem to COMM1

The `CONSOLE` environment variable on the remote system should be set to `serial`.

See the hardware documents supplied with your modem for connecting the modem to your system. See `modem`(7) to obtain the correct modem settings and for instructions on how to create the appropriate system file entries. In particular, the `cons` entry in `/etc/inittab` file should be modified so that the `getty` or `uugetty` process sets up the COMM port correctly. This line is similar to the following example:

```
cons:1234:respawn:/usr/sbin/getty console console vt100
```

This line should be changed as follows if you are using a modem set to run at a baud rate of 38,400 as a console device:

```
cons:1234:respawn:/usr/sbin/getty console M38400 vt100
```

1.11.1.2 Setting the Configurable DCD Timer Value

The serial driver has been modified to allow the Carrier Detect (DCD) timeout value to be configurable. The default value for this timer is 2 seconds, which is in accordance with the DEC STD-052 standard and is acceptable for most modems. This timer is used to determine how long the driver must wait when the DCD signal drops, before declaring the line disconnected and dropping the DTR and RTS signals. Some modems expect DTR to drop in a shorter time interval; see your modem documentation to verify the interval.

The timer can be modified by the `/etc/sysconfigtab` file or the `sysconfig` command to set the timer to 0 (no timeout period), 1, or 2 seconds. To set the timer via the `/etc/sysconfigtab` file, edit the file and include the following:

```
ace:
  dcd_timer=n
```

n can be 0, 1, or 2.

The syntax for modifying the timer via the `sysconfig` command is as follows:

```
# sysconfig -r ace dcd_timer=n
```

n can be 0, 1, or 2.

By modifying the value with the `sysconfig` command, the setting is lost when the system is rebooted. To preserve the setting across reboots, edit the `/etc/sysconfigtab` file.

1.11.1.3 Setting the Console Environment Variables

The `COM1_MODEM`, `COM1_FLOW`, and `COM1_BAUD` console environment variable settings must be equivalent to the `getty` or `uugetty` settings used when you created your system file entries for the modem.

See your hardware documentation for information on how to set the console environment variables. Typically, the variables are set when the system is shut down and in console mode, as shown in the following example:

```
>>> set COM1_MODEM ON
>>> set COM1_FLOW SOFTWARE
>>> set COM1_BAUD 9600
```

Valid settings are as follows:

- `COM1_MODEM`: ON or OFF
- `COM1_FLOW`: NONE, HARDWARE, SOFTWARE, BOTH
- `COM1_BAUD`: See your system hardware documentation.

Changing the baud rate, flow control, or modem setting (for example, using the `getty` command), causes those values to be propagated down to the console level; the environment variables change automatically.

1.11.1.4 Verifying the Modem Setup

Dial the remote system and obtain a login prompt or console prompt, if the system is not booted. Log out or disconnect and ensure that the line hangs up correctly. Dial in again to ensure that you can reconnect.

1.11.2 Initiating a Console Port Connection

You can initiate a connection between the local and remote systems by different methods. A `tip`, `kermit`, or `cu` connection can be initiated from a terminal or X-terminal window or you can use a PC-based terminal emulator.

For example, use the `tip` command as follows:

```
# tip [telephone number]
# tip cons
```

Where `telephone_number` is the telephone number of the remote system, including any prefixes for outside lines and long-distance codes. The second line is an example of an entry in the `/etc/remote` file, which you can use to specify details of remote systems and tip settings.

After you have initiated the dial-out command, and the two modems have established a connection, the word `connect` is displayed on your local terminal window. Press the Return key and the console prompt (>>>) or the `login:` prompt is displayed.

See `tip`(1) for more information.

1.11.2.1 Using the Console Port

After you have access to the system and are logged in to a privileged account, you can perform any of the administration tasks described in this volume that do not require access to a graphical user interface, such as using commands and running utilities. The following features may be useful for remote administration:

`uucp` — The UNIX to UNIX system copy utility for copying scripts and files to the remote system. See `uucp`(1) for more information.

`ikdebug` — A kernel debugging tool, `ikdebug` can be invoked and used remotely. See `ikdebug`(8) for more information. You may need to change an entry in the `/etc/remote` file to correct the baud rate. For

example you may need to change the baud rate from 9600 baud in the following lines:

```
# access line for kernel debugger
kdebug:dv=/dev/tty00:br#9600:pa=none:
```

See the *Kernel Debugging* manual for more information.

1.11.2.1.1 Turning Off Console Log Messages

The `syslogd` daemon now has an internal switch to disable and enable messages to the console. This feature is invoked by the `-s` flag on the `syslogd` command line, or by running the following command:

```
# /usr/sbin/syslog
```

See `syslog`(1) for more information.

1.11.2.1.2 Shutting Down the Remote System

When you shut down the remote system, the modem connection is dropped. To avoid this, use the following command before you shut down the system:

```
# stty -hupcl
```

See `stty`(1) for more information.

When the shutdown is complete, you still have access to the console prompt.

1.11.2.1.3 Ending a Remote Session

To end a remote session from the operating system shell prompt, type Ctrl/d to log out and terminate the remote session. Otherwise, type `+++` to put the modem into local command level, and type ATH followed by the Return key to hang up the connection.

1.11.3 Troubleshooting

If you have problems setting up your systems and connecting, verify the set up as follows:

- The local modem does not dial out.

  Examine the cables and connections and ensure that the telephone lines are plugged into the correct sockets, and that you have a dial tone.

- The remote modem fails to answer.

  Ensure that the remote modem is set to auto-answer, `ATS0=n`, where `n` is the number of rings before the modem answers.

  See `modem`(7) and verify the settings for dial-in access.

- The remote modem answers and then disconnects.

This is most likely because of incorrect settings for dial-in access. See `modem`(7) and verify the settings for dial-in access.

- The remote modem answers but only random characters are printed.

  This problem usually is caused by a mismatch between the baud rate of the COMM port and that of the modem. See `modem`(7) and verify the settings for dial-in access.

- The connection is dropped when the remote system is shut down by the `shutdown` command.

  The `stty` attribute `hupcl` is at the default setting. To prevent the line from disconnecting during a shut down, use the following command:

  ```
  # stty -hupcl
  ```

2

Starting Up and Shutting Down the System

This chapter contains the following information:

- An overview of starting up and shutting down the system (Section 2.1)
- A discussion of the boot operation (Section 2.2)
- Information on how to prepare to boot your system (Section 2.3)
- Information on how to boot your system (Section 2.4)
- A description of the different system run levels (Section 2.5)
- Information on how to change the system run level (Section 2.6)
- Boot considerations for multiprocessor systems (Section 2.7)
- An explanation of how to set the system date and time (Section 2.8)
- Information on how to troubleshoot boot problems (Section 2.9)
- A description of options for shutting down the system (Section 2.10)
- Information on how to shut down the system from multiuser mode (Section 2.11)
- Information on how to shut down the system from single user (root) mode (Section 2.12)

2.1 Overview of the Shutdown and Boot Operations

Shutting down the system and then restarting it are routine tasks that you need to perform periodically. In some computing environments, it is important to keep the system running and available at all times, and to shut down intentionally only for scheduled maintenance or software upgrades.

Tru64 UNIX enables you to minimize the number of shutdowns and, thus, maximize the time your system is running with these features:

- The Advanced File System, AdvFS, provides features that enable you to take a backup without taking the system off line.
- Diagnostic tools, such as the Event Manager, sys_check, and Memory Trolling, enable you to detect system problems early so that they can be corrected before they cause a shutdown. Event Manager is discussed

in Chapter 13. See Chapter 12 for information on sys_check. Memory Troller is described in *Managing Online Addition and Removal*.

- Hot swapping of CPUs and storage components such as disks and tapes is a feature that helps to prevent unplanned shutdowns caused by component failure. Hot swapping of CPUs is supported only on some recent hardware platforms; see the *Managing Online Addition and Removal* manual for more information. Hot swapping of storage devices is supported on most recent platforms. See the hardware documentation for your processor, the *Hardware Management* manual and `hwmgr`(8) for more information.
- With clustered systems, even software upgrades can be accomplished on one cluster member while the cluster is up and running. This feature is referred to as a "rolling upgrade."

Usually, you can shut down the system easily and with minimal disruption to system users. Occasionally, you must shut down the system rapidly, causing a moderate degree of disruption to users. Under some circumstances (that are out of your control), the system shuts itself down suddenly, causing substantial disruption to users. Develop a site-specific operations manual to define your:

- Procedures and schedule for planned shutdowns.
- Procedure for determining the cause of a shutdown and:
  - Correcting any errors or problems. See Chapter 11, Chapter 12, and Chapter 14 for information on troubleshooting.
  - Bringing the system back on line as quickly as possible.
  - Recovering lost data, if required. See Chapter 9 for information on backing up your system.

Shutting down a system requires root (superuser) privileges. Depending on the system configuration, there are several options available for intentionally shutting down and rebooting the system.

2.1.1 Shutdown Methods

You can shut a system down automatically or manually.

Automatic Shutdown

Configure system-monitoring tools such as environmental monitoring to shut down the system automatically if certain system events occur. See Chapter 13 for information on event management.

Manual Shutdown

- The SysMan Menu and SysMan Station enable you to shut down a local or remote system or cluster. The General Tasks branch of the SysMan Menu contains the task "Shutdown the System" that invokes the appropriate user interface, depending on how you access the SysMan Menu. Also, you can invoke the task from the command line by entering the following command:

  ```
  # sysman shutdown
  ```

  See Chapter 1 for more information.
- Run the `/usr/sbin/shutdown` command line interface from a character-cell terminal. See `shutdown`(8) for the available command options.
- The Shutdown icon in the CDE Application Manager – DailyAdmin folder invokes the SysMan Menu task named "Shutdown the System".

2.1.2 Boot Methods

You boot the operating system by using the system's console. When a system is powered on, the symbol >>> indicates the console prompt. At this prompt, you enter commands or set system configuration variables, such as variables that control what happens when a system is booted. Throughout this chapter, the symbol >>> is referred to as the console prompt. The console is sometimes called the System Reference Manual (SRM) console or the firmware console. See the owner's manual that came with your system for information on the commands you can enter at the console prompt.

You can boot a system as follows:

- Manually from the console.
- Using a network or modem connection, such as the remote console method documented in Chapter 1.
- Specifying boot actions that happen after a shut down. For example, if you use SysMan Menu or the SysMan Station to initiate a shut down, you can set the system to reboot automatically to single user mode after the shutdown is completed.
- Setting the `auto_action` console variable to boot the system automatically, especially after an unintentional shutdown, such as that caused by a power disruption. This is sometimes referred to as an unattended boot.

2.1.3 Related Documentation

Additional documentation relevant to system shutdowns and reboots can be found in manuals, reference pages, and online help.

2.1.3.1 Manuals

The following list refers to information on using system shutdowns and reboots in the Tru64 UNIX operating system documentation set.

- See the Owner's Manual that came with your system for information on the console commands and variables. See `consvar`(8), which describes `consvar`, a command that enables you to manipulate console environment variables from within the operating system, depending on the firmware revision.
- See the *AdvFS Administration* and *Logical Storage Manager* manuals for information on file systems, should you need to examine and repair damaged file systems before rebooting.
- See the *Installation Guide* for information about installing the system and performing the initial boot operation. The information in this chapter assumes that you are booting or rebooting an installed operating system.
- The *Kernel Debugging* manual provides information on analyzing crash dump files.
- See the *Hardware Management* manual for information on diagnosing hardware problems.

This manual also contains the following topics of relevance to planning and managing shutdowns and error recovery:

- Some systems support environmental monitoring, which you can use to shut down a system automatically in the event of a problem such as loss of a cooling fan. See Chapter 12 for information on configuring this feature.
- See Chapter 12 for information on error conditions, log files, and crash dumps.
- The Event Manager and the SysMan Station provide integrated monitoring and event reporting facilities that enable you to monitor local and remote systems and clusters. See Chapter 1 for information on invoking these features.
- See Section 1.11 in Chapter 1 for information on remote serial consoles if you administer systems at remote locations, or if there is a network failure that requires dial-up communications.
- See Chapter 9 for information on implementing a backup schedule, from which you can recover lost data if necessary.

2.1.3.2 Reference Pages

The following reference pages provide more information on the command options and interfaces:

`shutdown`(8)	Invocation and use of the `shutdown` command line interface.
`sysman`(8) and `sysman_station`(8)	Information on the use of the SysMan options and a description on invoking these utilities so that you can then run the "Shutdown the System" task.
`wall`(1) and `rwall`(1)	Utilities to write to users (warning them of a system shutdown).
`halt`(8) and `fasthalt`(8)	Utilities to halt the system.
`reboot`(8) and `fastboot`(8)	Utilities to boot the system.
`fsck`(8)	Utility to examine and repair the UFS file system.
`init`(8)	Utility for initializing the system.
`rc0`(8), `rc2`(8), and `rc3`(8)	Command scripts that are run when stopping the system, entering run level 2, and entering run level 3.
`consvar`(8)	Command to manipulate system firmware console environment variables.

2.1.3.3 Online Help

The following online help is available:

- The `shutdown -h` command provides help on the command line options.
- The shutdown utility, available from the SysMan Menu, features an extensive online help facility.
- An online help volume is provided for each SysMan Menu and SysMan Station task. See also the introductory online help available at: `/usr/doc/netscape/sysman/index.html`.

See Chapter 1 for information on invoking online help.

2.1.4 System Files

The following system files are used during boot and shutdown operations:

`/etc/inittab`	Provides the `init` program with instructions for creating and running initialization processes.
`/vmunix`	The default name of the custom kernel. When you build a custom kernel, you can choose any legal file name.
`/genvmunix`	The default name of the generic kernel. You boot the generic kernel to build a custom kernel, or if the custom kernel is corrupt and non-bootable.
`/sbin/rc0`, `/sbin/rc2`, and `/sbin/rc3`	Run level command scripts. The `rc0` script contains run commands that enable a smooth shutdown and bring the system to a single-user state. The run commands are contained in the `/sbin/rc0.d` directory. The `rc2` script contains run commands that enable initialization of the system to a multiuser state; run level 2. The run commands are contained in the `/sbin/rc2.d` directory. The `rc3` script contains run commands that enable initialization of the system to a multiuser state; run level 3. The run commands are contained in the `/sbin/rc3.d` directory.

2.1.5 Related Utilities

You also may use the following utilities during the boot operation:

`fsck`	The `fsck` command is a wrapper program for the `ufs_fsck` program, which examines and repairs UFS file systems. See `advfs`(4) and the *AdvFS Administration* manual for information on verifying AdvFS file systems
`consvar`	The `consvar` command gets, sets, lists, and saves console environment variables while the operating system is still running. To see if your system supports `consvar`, use the following command:

```
# /sbin/consvar -l
auto_action = HALT
boot_dev = dsk0
bootdef_dev = dsk0
booted_dev = dsk0
boot_file =
booted_file =
boot_osflags = A
.
.
.
```

If `consvar` is supported, the current settings of several console variables are displayed.

2.2 Understanding the Boot Operation

When you boot the operating system, you initiate a set of tasks that the system must perform to operate successfully. The system is vulnerable during startup because it is loading the kernel into memory and initializing routines that it depends on for operation. Consequently, you must understand what is happening during the system boot operations, and be prepared to respond if problems occur.

2.2.1 Booting Automatically or Manually

The system boots either automatically or manually. In an automatic boot, the system begins the initialization process and continues until completion or failure. You need only to intervene manually if the automatic boot fails for some reason. For example, if the `fsck` command cannot verify file systems.

In a manual boot, the system controls the initial operation, turns control of the procedure over to you and then reinstates control to complete the operation. When you boot the system to single-user mode, you are relying on a manual boot. In an automatic or a manual boot, the operation either succeeds or fails:

- If the boot operation succeeds, the system is initialized. In single-user mode, the system displays the superuser prompt (#) on the console or on the terminal screen. In multiuser mode, the system displays the `login` prompt or a startup display. The prompt or startup display differs according to hardware capability and available startup software.
- If the boot operation fails, the system displays an error message followed by a console prompt (>>>). In the worst case, the system hangs without displaying a console prompt.

2.2.2 Booting to Single-User or Multiuser Mode

The system boots to either single-user or multiuser mode.

Because the `init` operation does not invoke the startup script prior to turning control over to you, the root file system is mounted read only. Startup of the network and other daemons does not occur, file verification and correction are not enabled, and other operations necessary for full system use are not automatically available to you.

Usually you boot to single-user mode to perform specific administrative tasks that are best accomplished without the threat of parallel activity by other users. You perform these tasks manually before exiting from the Bourne shell. For example, you may verify new hardware, mount and verify aberrant file systems, change disk partitions, or set the system clock. When you finish your work, you return control to the system, and the `init` operation continues with its startup tasks and boots to multiuser mode.

In a boot to multiuser mode, the system loads the kernel and moves through various phases such as hardware and virtual memory initialization, resource allocation, scheduling, configuration and module loading.

At the conclusion of the main initialization tasks (process 0), `init` (process 1) starts an additional set of tasks that includes reading the `/etc/inittab` file, acting on instructions found there, and executing the relevant run command scripts. These scripts contain entries that initiate activities such as mounting and verifying file systems, removing temporary files, initializing the clock daemon, initializing the network daemon, setting up printer spooling directories and daemons, enabling error logging, and performing other tasks specified within the scripts or in related directories.

At the conclusion of these activities, the system is enabled and accessible to users.

The operating system allows you to boot an alternate kernel if your custom kernel is not bootable. You can boot the generic kernel (`/genvmunix`) to troubleshoot the problem with your system. Also, you can boot an alternate custom kernel to test new drivers or to add options to the existing kernel.

2.3 Preparing to Boot the Installed System

As the system administrator, you set up or encounter various preboot or postshutdown states. The following sections describe and recommend procedures for preparing and initiating a reboot from a variety of system states. The states include the following:

- A powered-down system (Section 2.3.1)
- A powered-up, halted system (Section 2.3.2)

- A powered-up system in single-user mode (Section 2.3.3)
- A crashed system (Section 2.3.4)
- A networked system that was taken out of the network (Section 2.3.5)

Note

If the system is running in single-user mode and you want to use the `ed` editor, you must change the protections of the root file system to read-write. At the superuser prompt, enter the following command:

```
# mount -u /
```

2.3.1 Preparing to Boot a Powered-Down System

Follow these steps to power up and boot your system:

1. Confirm that the hardware and all peripheral devices are connected. See the operator's manual for your hardware for information and instructions for interpreting diagnostic output.
2. Power up peripheral devices. See the operator's manual or the hardware user's manual for instructions on starting them.
3. Power up the processor.
4. Confirm that the hardware completed its restart and diagnostic operations. Most hardware provides a diagnostic examination as a routine part of its startup operation. See the operator's manual for your hardware for information about your hardware's restart and diagnostic operations.
5. Wait for the console prompt (>>>). If you enabled your system to boot automatically when it is powered up, press the Halt button to display the console prompt. See the hardware operator's manual for the location of the Halt button on your system. See Section 2.4 for more information on setting the default boot action for your system.
6. Decide which startup mode you want to initiate:

   single-user mode — Plan to use this mode if you have tasks you need to accomplish and want the system to restrict access to all users but root.

   multiuser modes — Plan to boot to one of the multiuser modes (multiuser without networking or multiuser with networking) if you do not require

single-user access and you want the system to initialize all functions.

7. Enter the boot command that corresponds to the startup mode you want. See Section 2.4 for the commands and procedures required to boot your system.

2.3.2 Preparing to Boot a Powered-Up, Halted System

When your machine is powered up and enabled but the processor is halted, the system is in console mode. For example, after you shut down the processor with the `shutdown -h` command or when you run the `halt` command, your system displays the console prompt (>>>).

When the system displays the console prompt, follow these steps to prepare to boot your system:

1. Decide which startup mode you want to initiate:

single-user mode	Plan to use this mode if you have tasks you need to accomplish and want the system to restrict access to all users but root.
multiuser modes	Plan to boot to one of the multiuser modes (multiuser without networking or multiuser with networking) if you do not require single-user access and you want the system to initialize full functionality.

2. Enter the boot command that corresponds to the startup mode you want. See Section 2.4 for the commands and procedures required to boot your system.

2.3.3 Preparing to Transition from Single-User Mode

The system is in single-user mode when it is powered up and enabled, the processor is running, and access is limited to root.

When the system displays the superuser prompt (#), follow these steps to prepare to go to multiuser mode:

1. Decide if you need to continue in single-user mode or if you require multiuser mode:

single-user mode	Continue in this mode if you have additional tasks to perform and you want the system to restrict access to all users but root.

multiuser modes — Plan to go on to one of the multiuser modes (multiuser without networking or multiuser with networking) if you do not require single-user access, or if you have completed your tasks and you want the system to initialize full functionality.

2. When you are ready to go to multiuser mode, press Ctrl/d. See Section 2.4 for the commands and procedures required to boot your system.

2.3.4 Preparing to Boot a Crashed System

If your system crashes and is unable to recover automatically and reboot itself, follow these steps to prepare to boot the system:

1. See Chapter 12 for information on saving crash dump files, and to verify system log files for any information on the causes of the crash.
2. Confirm that the hardware and all peripheral devices are connected.
3. Power up the hardware, if necessary. Always power up peripherals and devices before the processor.
4. Monitor the hardware restart and diagnostic operations. See the operator's manual for your hardware for information and instructions for interpreting diagnostic output:
   - If the diagnostic test indicates hardware failure, contact your field service representative. Because hardware damage is a serious problem, do not continue or try to bypass the defective hardware.
   - If you have enabled your system to boot automatically, press the Halt button to display the console prompt. See the hardware operator's manual for the location of the Halt button on your system.
5. Decide which startup mode you want to initiate:

   single-user mode — Plan to work in this mode if you need to deny access to all users but root. After a crash, it is wise to work initially in single-user mode. Verify all file systems thoroughly for inconsistencies and perform other post crash operations before enabling system access to other users.

   multiuser modes — Plan to boot to one of the multiuser modes (multiuser without networking or multiuser with networking) if you need to allow access to you and to all other users with login permission

6. Enter the required boot command. See Section 2.4 for the commands and procedures required to boot your system.

2.3.5 Preparing to Boot a System Taken Off the Network

If a system is configured to support a network, the boot operation tries to start all the network services that are configured. This results in the boot process hanging or taking a very long time to test for the presence of services. You may want to remove a functioning system from a network to use the system in standalone mode or to correct a system problem, such as a failed network device.

If you take a system out of a network without reconfiguring the services, or if a system crashes and you must disconnect it from the network, perform the additional steps before rebooting the system

There may be instances when you want to remove a functioning system from a network, for example:

- To use the system in standalone mode
- To correct a system problem such as a failed network device

The following procedure assumes that the system is halted at the console prompt:

1. At the console prompt, set the `boot_osflags` environment variable to `s`, to stop the boot at single-user mode as follows:

   ```
   >>> set boot_osflags s
   ```

   If you intend to do things such as boot from an alternate disk, set the appropriate console variables at this time. See Section 2.4.1 for more information.

2. Boot the system to single-user (standalone) mode:

   ```
   >>> boot
   ```

3. When the system displays the superuser (`#`) prompt, mount the root file system as writable by using the following command:

   ```
   # mount -u /
   ```

   Mounting the root file system as writable enables you to use the `ed` line editor to edit system files and to access commands and utilities. Other editors such as `vi` are not available at this time, as they do not reside on the root file system (`/`).

4. Copy the `/etc/rc.config`, `/etc/rc.config.common` and `rc.config.site` files for safe keeping. For example:

   ```
   # cp /etc/rc.config /etc/orig_rc.config
   # cp /etc/rc.config.common /etc/orig_rc.config.common
   ```

```
# cp /etc/rc.config.site /etc/orig_rc.config.site
```

Note

The integrity of the /etc/rc.config, /etc/rc.config.common and /etc/rc.config.site files is important for startup operations and for system configuration. Do not modify these files with anything other than the rcmgr command. If the format of the files is not correct, other subsystems or utilities may not parse the files correctly. See rcmgr(8) for more information. See the TruCluster documentation for more information on performing boot operations on cluster members.

5. Use the rcmgr line editor to modify entries in the configuration file that invoke networking services. For example, to test for and turn off Network Information Service (NIS), enter the following command:

```
# rcmgr get NIS_CONF
YES
# rcmgr set NIS_CONF NO
```

Repeat this operation for each network service that is currently called, such as NTP or NFS.

6. When you complete the modifications, halt the system and reset any console environment variables. For example:

```
>>> set boot_osflags a
>>> boot
```

Your system reboots to multiuser mode, without attempting to start any network services.

There are variations in the console commands depending on your system model and the firmware revision. See the hardware documentation for a description of console commands for your processor.

2.4 Booting the System

The command that you use to boot the kernel depends on several factors:

- Processor type
- Run level
- Location of the kernel that you are booting (on the system disk or on a remote server)
- Whether you are booting all processors or a single processor (in a multiprocessor system)

- Whether any console environment variables are defined
- Whether you are booting the default kernel or an alternate kernel

2.4.1 Defining the Console Environment Variables and Using the Boot Commands

Examples of typical console settings are shown in this section. See the hardware documentation that came with your system for specific information. See also the information on booting systems in the *Installation Guide* and the *Installation Guide — Advanced Topics* manual.

If you are using RAID storage arrays or fibre channel controllers in a storage area network, you must use the appropriate storage management software to get and set boot device information. See your storage array documentation.

To boot your system you need to understand the use of certain console environment variables and their role in affecting the boot process. Table 2–1 lists each of the console environment variables and their associated actions.

Table 2–1: Console Environment Variables

Variable	Action
`boot_reset`	When set to `on`, resets the hardware on boot
`boot_osflags`	A combination of flags used to control the boot loader and kernel
`bootdef_dev`	Identifies the boot device
`boot_file`	Identifies the kernel to boot
`cpu_enable`	Selectively enables particular processors from the console

To prepare the hardware for the boot operation, perform the following operations at the console prompt:

1. Set the `auto_action` variable to `halt`:

   ```
   >>> set auto_action halt
   ```

   This command halts the system at the console prompt each time your system is turned on, when the system crashes, or when you press the Halt button.

2. If required for your processor, set the `boot_reset` variable to `on` to force the resetting of the hardware before booting:

   ```
   >>> set boot_reset on
   ```

3. If required for your processor, set the time to wait to reset the SCSI device before booting:

   ```
   >>> set scsi_reset 4
   ```

4. Use the following procedure to set the boot_osflags variable and the boot device:

   a. Determine which options to the boot_osflags variable you want. Table 2–2 lists the options.

Table 2–2: Options to the boot_osflags Variable

Option	Action
a	Boot to multiuser mode. (By default, the kernel boots to single-user mode.)
k	Use the kdebug debugger to debug the kernel. See the *Kernel Debugging* manual for more information.
d	Use full crash dumps. (By default, partial dumps are used.) See Chapter 12 for information on crash dumps.
i	Prompt for the kernel and special arguments. (By default, no prompts are displayed). See Section 2.4.3 for an example of an interactive boot.

   The options are concatenated into the boot_osflags variable to achieve the effect you want. For example, to boot to multiuser mode and use full crash dumps, enter:

```
>>> set boot_osflags ad
```

   If you want the defaults, clear the variable as shown in the following example:

```
>>> set boot_osflags ""
```

   b. Determine the unit numbers for your system's devices:

```
>>> show device
```

   c. Set the default boot device.

   By default, you must provide a boot device when you boot your system. If you always boot from the same device, use the following command with the bootdef_dev variable to set a default boot device. For example, to boot the system off of disk dka0, enter:

```
>>> set bootdef_dev dka000
```

   Hardware configurations can include HSZ controllers that are connected to dual KZPBA-CB buses and configured for multibus failover. In this case, you specify both bus paths to the boot disk devices when setting the bootdef_dev console variable. During configuration of a dual-controller system, one of the controllers is designated as the preferred path. Specify the boot devices on this controller as the first arguments to the bootdef_dev console variable.

For example, a system has two controllers A and B connected to four logical volumes dka0, dka1, dkb0, and dkb1. If controller B is designated as the preferred controller, then the bootdef_dev console variable must specify the **b* devices first, as follows:

For example:

```
  >>> set bootdef_dev dkb0.0.0.0.6.0, \
dka0.0.0.5.0
```

Separate each device path with a comma; do not use spaces or tab characters. If the console is unable to boot from the first device, it tries the next device.

d. You have the option of booting from an alternate kernel. If you want to do this, enter:

```
>>> set boot_osflags i
```

When booting, the system prompts you to enter a path to the kernel. For example:

```
Enter [kernel_name] [option_1 ... option_n]: \
genvmunix
```

The system displays informational messages.

On some processors, you can boot an alternate kernel by setting the boot_file variable to the name of the kernel you want to boot. For example, to boot a generic kernel (/genvmunix), enter:

```
>>> set boot_file genvmunix
```

Depending on your processor, you may need to clear the boot_file variable if you want to boot the default kernel (/vmunix). For example:

```
>>> set boot_file ""
```

In a multiprocessor configuration, you can use the set cpu_enable command to selectively enable processors from the console. The mask is a bit field, where each bit represents a slot position. The easiest way to ensure all processors are enabled is to set the CPU mask to ff. After setting the mask, cycle the system power.

The operating system also provides a mechanism for enabling or disabling processors at system boot time. See the description of the cpu-enable-mask attribute in the *System Configuration and Tuning* manual for more information.

After you have set the console variables, use the following command to boot the system:

```
>>> b
```

2.4.2 Overriding the Boot Commands

The following list describes how to override the commands presented in Section 2.4.1.

- Overriding the `bootdef_dev` console variable.

  To override the `bootdef_dev` console variable, supply the boot device you want as an argument to the boot command. For example, if your boot device is set to boot from disk `dka0` and you want to boot from disk `dkb0`, enter:

  ```
  >>> b dkb0
  ```

- Overriding the `boot_osflags` console variable.

  The `boot_osflags` variable is ignored if you specify the `–fl` option to the boot command, as follows:

  ```
  >>> b -fl
  ```

  To override the `boot_osflags` variable, specify your choices with the `–fl` option. For example, the following command boots to the interactive prompt so you can specify an alternate kernel, and then boots to multiuser mode:

  ```
  >>> b -fl ai
  ```

  See Table 2–2 for a list of options. An example of an interactive boot session is provided in Section 2.4.3.

- Overriding the `boot_file` console variable.

  Specify the path to a kernel file to boot a kernel other than that specified by the `boot_file` console variable. For example, to boot the generic kernel (`/genvmunix`), enter the following command:

  ```
  >>> b -fi genvmunix
  ```

2.4.3 Using Interactive Boot to Verify the Root File System

Use the `-flags i` option with the console boot command to invoke an interactive boot session. Depending on the console command options available for your system, you enter other boot options and parameters with the `-i` option. (See the owner's manual for your processor for more information on interactive boot options.)

The interactive boot session runs the `osf_boot` command that is located in the root file system (`/`). It enables you to examine the root file system without fully booting the system. Use the following procedure to perform this task. It is assumed that your system is shut down and at the console prompt:

1. From the console prompt (>>>) enter the following command to boot the system in interactive mode:

```
>>> boot -flags i
```

2. The following message is displayed:

```
UNIX Boot - date

Enter: <kernel_name> [option_1...option_n]
or: ls [name]['help'] or quit to return to console
Press return to boot 'vmunix'#
#
```

The following options are now available:

a. Enter the name of an alternate kernel and specify required boot options. See the owner's manual for your system for a list of boot options.

b. Enter the following command to obtain help on the `ls` command:

```
# help
```

The `ls` command options are described in Step 3 below.

c. Enter the `quit` command to return to the console prompt.

d. Press Return to boot the default custom kernel (`/vmunix`) if no other kernel is specified by the `boot_file` console variable.

3. Use the `ls` command to list the content of root file system directories or to list specific files. If you do not specify a file name, the entire content of the directory is displayed. The following are examples of valid commands:

- This command lists the entire content of the top-level root directory (`/`):

```
# ls /
```

- Because you are displaying to the console and other commands are not available, you have no control over the display output and it may scroll off the screen. Use the question mark (`?`) and asterisk (`*`) wildcard characters to match characters and strings. For example:

```
# ls /etc/*rc*
```

This command returns any file in the `/etc` directory that matches the string rc, such as `/etc/rc.config`.

Wildcard characters are supported for file names, but not directory names.

2.5 Identifying System Run Levels

A run level (mode) specifies the state of the system and defines which processes are allowed to run at that state. The most commonly used run levels are as follows:

Run Level	System State
0	Specifies the halt state
S or s	Specifies single-user mode
2	Specifies multiuser mode without network services
3	Specifies multiuser mode with network services
null	Specifies the console mode

The `inittab` file contains line entries that define the specific run levels and the run command scripts that are associated with the run level. When the `init` process starts, it reads the `inittab` file and executes the relevant run command scripts. The scripts, in turn, define which processes run (and which processes are killed if the system changes from one level to another) at a specific run level. See `init`(8), `inittab`(4), and Chapter 3 for information about reading and modifying the `inittab` file.

Section 2.6.2 describes how you use the `init` command to change the run level.

2.6 Changing System Run Levels

Before changing to a new run level, see the `inittab` file to confirm that the run level to which you intend to change supports the processes you need. Of particular importance is the `getty` process because it controls the terminal line access for the console and other logins. Make sure that the `getty` entry in the `inittab` file allows system console access at all run levels. See `inittab`(4) for more information about defining run levels. See `getty`(8) for more information about defining terminal lines and access.

A change in run level can terminate a user's `getty` process, disabling their login capability and may terminate other user processes. Before changing to a new run level, use the `wall` or `write` command to warn users that you intend to change the run level.

Examine the `getty` entry for user terminals to verify that the new run level is specified in the entry. If it is not, request that users log off so that their processes are not terminated in response to a `kill` signal from the `init` process.

When the system is initialized for the first time, it enters the default run level that is defined by the `initdefault` line entry in the `inittab` file. The

system continues at that run level until the init process receives a signal to change run levels. The following sections describe these signals and provide instructions for changing run levels.

2.6.1 Changing Run Levels in Single-User Mode

Use the Bourne shell when working in single-user mode and press Ctrl/d to change run levels. When you press Ctrl/d, the shell terminates and the following message is displayed:

```
INIT: New run level: 3
```

You typically see this message when you transition from single-user mode to multiuser mode during a boot operation. At other times, you are prompted to supply a run level. See init(8) for more information about run level transitions.

The init process searches the inittab file for entries (at the new run level) with the boot or bootwait keywords, and then acts on these entries before it continues with the normal processing of the inittab file. The init process next scans the file for other entries with processes that are allowed to run at the new run level, and then acts on these entries.

2.6.2 Changing Run Levels from Multiuser Mode

When the system is running at one of the two multiuser run levels, you can use the init command to change run levels as follows:

Run Level	System State
0	Specifies the halt state.
2	Specifies a multiuser run level with local processes and daemons.
3	Specifies a multiuser run level with remote processes and daemons.
1, 4, 5 – 9	Changes the run level to that specified by the number flag in the /etc/inittab file. If no such entry exists, no action is taken and no message is displayed.
M, m	Moves control to the console device and halts to single-user mode.
Q, q	Specifies that the init process should reexamine the inittab file.
S, s	Changes the run level to a single user state with only the essential kernel services.

2.6.2.1 Changing to a Different Multiuser Run Level

To change from the current multiuser run level to a different multiuser run level, enter the `init` command with the argument that corresponds to the run level that you want to enter. For example, to change from run level 2 to run level 3, enter the following command:

```
# init 3
```

In response to your entry, the `init` process reads the `inittab` file and follows the instructions that correspond to the change in run level.

2.6.2.2 Changing to Single-User Mode

The `init` command provides a way to change from the current multiuser mode to single-user mode by using the `s` run level argument. For example, to change from the current run level to single-user mode, enter:

```
# init s
```

To change from a multiuser mode to single-user mode, giving users a 10-minute warning, enter:

```
# /usr/sbin/shutdown +10 Bringing system down to single-user for testing
```

To return to multiuser mode from single-user mode, type Ctrl/d or enter the `exit` command at the prompt. This causes the `init` command as process 1 to prompt you for the run level. In response to the prompt, enter 2 to return to multiuser mode without networking daemons activated, or enter 3 to return to multiuser mode with networking daemons activated.

Alternatively, you can reboot the system by using one of the following commands:

```
# /usr/sbin/shutdown -r now
```

```
# /sbin/reboot
```

2.6.2.3 Reexamining the inittab File

To reexamine the `inittab` file, enter the `init` command with the `q` argument, as follows:

```
# init q
```

In response, the `init` process reexamines the `inittab` file and starts new processes, if necessary. For example, if you recently added new terminal lines, the `init` process activates the `getty` process for these terminal lines in response to the `init q` command.

See `getty`(8) for more information about the relationship between terminal lines and the `init` command.

2.7 Symmetric Multiprocessing

Symmetric Multiprocessing (SMP) consists of two or more processors that execute the same copy of the operating system, address common memory, and can execute instructions simultaneously. In a multiprocessor system, multiple threads can run concurrently through simultaneous execution on multiple processors.

If your system is a multiprocessor system and it is running Tru64 UNIX, it is running in an SMP environment. The objective of the operating system in an SMP environment is to take advantage of the incremental computing power available to the system as additional processors are added. To do this, the operating system must allow multiple threads of execution to operate concurrently across the available processors.

2.7.1 Adding CPUs to an Existing System

At boot time, the system determines the number of CPUs available. To add computing power to your multiprocessing system, install the processor board and reboot the system. You do not have to reconfigure the kernel but you may need to modify any tuning that limits the number of processors available. See the *System Configuration and Tuning* manual for more information. If you need to install a Product Authorization Key (PAK) see the *Software License Management* manual.

2.7.2 Unattended Reboots on Multiprocessor Systems

If a processor in a multiprocessor system fails, the operating system records which processor failed, then automatically reboots the system. Although the operating system continues, you must restart the failed processor manually. For instructions, see the *Installation Guide*.

2.8 Setting and Resetting the System Clock

The system has an internal clock that you set when you install the system. The clock maintains the time and date whether the power is on or off. Nevertheless, there are occasions when you may need to reset the time or date. For example, with battery-powered clocks, you may need to reset the time as a result of battery failure; or you may need to synchronize system time with standard time.

To set the date and time, log in as root and use the `date` command. The sequence of date and time parameters can vary depending on what command options you use. (See `date`(1) for more information.) Table 2–3 shows the value of the parameters:

Table 2–3: Parameters of the date command

Parameter	Description
cc	Designates the first two numbers of the year (century) as a 2-digit integer
yy	Designates the year as a 2-digit integer
MM	Designates the month as a 2-digit integer
dd	Designates the day as a 2-digit integer
HH	Designates the hour as a 2-digit integer, using a 24-hour clock
mm	Designates the minutes as a 2-digit integer
.	Serves as a delimiter
ss	Designates the seconds as a 2-digit integer (this field is optional)

For example, to set the date to 09:34:00 a.m. Sep 7, 2002 using the mmddHHMM[[cc]yy][.ss] format, enter one of the following commands:

```
# date  090709342002
# date  0907093402.00
# date  090709342002.00
```

If you change the year, update the system disk with the new year information. In single-user mode, enter the `mount –u /` command after you enter a date containing a new year. This command writes the new year into the superblock on the system disk. The root file system is mounted read-write.

2.9 Troubleshooting Boot Problems

If your system does not boot, the following suggests some areas for further investigation.

2.9.1 Hardware Failure

See the hardware manual accompanying your system for hardware test procedures. If a hardware problem exists, follow the instructions in the manual for resolving the problem.

2.9.2 Software Failure

Software can fail for the following reasons:

- You specified an incorrect boot path.

  See Section 2.4 or your system's hardware manual for instructions on specifying the correct boot path.

- The kernel is corrupt.

If you suspect that the kernel is corrupt, boot the generic kernel (`/genvmunix`). This provides you with a fully functional system and you can begin debugging procedures by using the `kdbx` or `dbx` utilities to analyze crash dumps. See `kdbx`(8) or `dbx`(1) for more information. See Section 2.4.1 for information on booting an alternate kernel.

- A disk or file system is corrupt.

  If a disk or file system is corrupt, run the `fsck` command on the file system. The `fsck` command verifies and repairs UNIX File Systems (UFS). If the `fsck` process finds something wrong, you are prompted to choose a recovery option. Use extreme care under these circumstances so that you do not inadvertently overwrite or remove any files. See `fsck`(8) for more information.

  If you have an Advanced File System (AdvFS), disk corruption is very unlikely. AdvFS provides disk recovery during the mount procedure that corrects the disk structures. You do not need to run the `fsck` command or any other command. Consequently, recovery of AdvFS is very rapid. See the *AdvFS Administration* manual for more information.

2.10 Shutting Down the System

The following sections describe the shutdown procedures and the recovery strategies that you use for both controlled and unexpected shutdowns. The first part discusses procedures for controlled shutdowns. The second part discusses guidelines and recommendations for recovering from unexpected shutdowns.

Typical reasons for shutting down a system are:

- You need to upgrade your software or add new hardware to your configuration. You shut down the system to set up the additions, make the necessary adjustments to your configuration files, and build a new kernel.
- You are monitoring the hardware error log and you notice repeated warning messages. You suspect that your hardware may soon fail, so you shut down the system and examine the problem.
- You notice that system performance is degrading rapidly. You examine the system statistics and conclude that some changes to the system would improve performance. You shut down and tune the system.
- You notice signs of possible file system corruption. You shut down the system and run the `fsck` program to fix problems or to confirm that none exist.
- The environmental monitoring utility, or the Event Manager (EVM) has given notification that a parameter is being exceeded, and failure is a possibility.

In each of these and similar situations a variety of options are available to you. Regardless of how you decide to resolve the situation, your first step is to initiate a controlled shutdown of the system. There are practical and reasonable ways to shut down your system from single-user mode or multiuser mode.

A system that has panicked or crashed presents you with a different set of circumstances than a system that has shut down in an orderly fashion. This chapter discusses orderly shutdowns only. See Chapter 12 for information on system crashes.

2.11 Stopping Systems While in Multiuser Mode

To shut down the system while running in multiuser mode, use the `shutdown` command or invoke the SysMan Menu task "Shut Down the System". When you issue the `shutdown` command with the `-h` or `-r` flags, the program typically performs the following operations in the order shown:

1. Runs the `wall` program to notify all users of the impending shutdown
2. Disables new logins
3. Stops all accounting and error-logging processes
4. Runs the `killall` program to stop all other processes
5. Runs the `sync` program to synchronize the disks
6. Logs the shutdown in the log file
7. Dismounts file systems
8. Halts the system

The following sections describe typical shutdown operations and provide examples of what happens when you use the command flags. See `shutdown`(8) for more information.

2.11.1 Using SysMan shutdown

Use the `sysman shutdown` command to invoke the SysMan Menu shut down task. You can invoke this interface from the SysMan Station or the SysMan Menu. See Chapter 1 for information on invoking the different SysMan interfaces, such as choosing the "Shutdown the System" option from the "General Tasks" branch of the SysMan Menu.

When you enter `sysman shutdown`, a window titled "Shutdown Targeted on *host name*" is displayed, where *host name* is the local system name. The shutdown task provides you with additional options if you are shutting down cluster members. See the TruCluster documentation if you are shutting down one or more members of a cluster.

The following options are available:

Shutdown type — Use this option menu to select one of the following shutdown options:

Halt	Halt the operating system and display the console prompt
Reboot	Shut down and halt the system, then automatically reboot it
Single user	Shut down to single-user mode, displaying the superuser prompt (#)
Message only	Broadcast a message to all current system users without shutting down the system

Minutes until shutdown — Hold down mouse button 1 (MB1) and move the slider bar to select the elapsed time in minutes before the shutdown operation begins (the shutdown delay). The time is displayed adjacent to the bar. You can select from a range of 0–60 minutes by using the slider bar. In some user environments, such as on a character-cell terminal, the slider bar is not available and you type a number to specify the shutdown delay. In these interfaces you can specify a time greater than 60 minutes.

Shutdown message — Type a message to users warning of the impending shutdown and requesting that they log out. This message, if any, is in addition to the message that is sent by default.

In a shutdown that is not `now`, messages are issued when the shutdown is started, and at regular intervals thereafter. For example, if a shutdown is requested in 55 minutes, messages are issued at 55, 50, 40, 30, 20, 10, 5, and 1 minute before shutdown, at 30 seconds before shutdown, and at shutdown time.

Broadcast message to NFS clients	Check this box if you want to broadcast a message to remote users of local NFS-served file systems. If a remote user is connected to any file system that is exported by the local system, that user receives a warning of the impending shutdown. To send such messages, ensure that the `rwalld` daemon is running on the remote user's system.
Execute run-level transition scripts	Check this box if you want to run the existing run-level transition scripts in the `/sbin/rc[`*N*`.d]/[K`*nn_name*`]` file. For example, `/sbin/rc0.d/K45.syslog`. See the `-s` option in `shutdown`(8) for more information.
Preshutdown Script	Specify a path to a custom script that you want to run before the shutdown completes. The script is run at shutdown time and completes any tasks that you specify prior to shutting down the system. If your script (or any intermediate scripts that it calls) fails to complete successfully, the system may not shut down correctly.
Other options	Check this box to enable options that make the shutdown faster:

Fast	Performs a fast shutdown, bypassing messages to users and NFS clients
No disk sync	Shuts down without synchronizing the disks by using the `sync` operation.

After you initiate a shutdown by using the SysMan Menu, the system shuts down as described in Example 2–1 in Section 2.11.2, except that a continuous countdown is displayed in the Shutdown: Countdown window. You can cancel the shutdown at any time.

See the online help for more information on the various options and `shutdown`(8) for more information on shutdown command behavior.

2.11.2 Shutting Down the System and Warning Other Users

You can perform this task by using the `shutdown` command or by invoking the SysMan Menu task `Shut down the system`.

To shut down the system from multiuser mode to single-user mode at specific times and warn users of the impending shutdown, follow these steps:

1. Log in as root and change to the root directory:

   ```
   # cd /
   ```

2. Use the shutdown command to initiate a shutdown. For example, to shut down and halt the system in 10 minutes with a warning to users that the system is shutting down for routine maintenance tasks, enter:

   ```
   # /usr/sbin/shutdown +10 "Planned shutdown, log off now"
   ```

Example 2–1 shows a typical shutdown sequence.

Example 2–1: A Typical Shutdown Sequence

```
# /usr/sbin/shutdown +6
"Maintenance shutdown, please log off"   1
System going down in 6 minutes
                ...Maintenance shutdown, please log off   2
System going down in 5 minutes
                ...Maintenance shutdown, please log off   3

No Logins, system going down @ <time>
                ...Maintenance shutdown, please log off   4

System going down in 60 seconds
                 ...Maintenance shutdown, please log off
System going down in 30 seconds
                 ...Maintenance shutdown, please log off
System going down immediately
                 ...Maintenance shutdown, please log off   5
.
.  process shutdown messages   6
.
Halting processes ...
INIT: SINGLE USER MODE   7
# halt
.
. <hardware reset messages>   8
.
resetting all I/O buses
>>>   9
```

1 This command initiates a shutdown, delayed for six minutes, and broadcasts a message to all users warning them to log off.

2 This message is echoed to the console terminal immediately, and to the terminal window from which you invoked the shutdown command.

3 These messages are echoed to the console terminal, and to the terminal window from which you invoked the `shutdown` command immediately. The messages are repeated at intervals, depending on the length of the original shutdown delay, becoming more frequent as shutdown time approaches.

4 When five minutes remain, new logins are disabled automatically. If anyone attempts to log in at this time, this message is displayed at the login terminal and it is not broadcast to other users.

5 This final message warns that the system is shutting down immediately and user processes are halted. The system stops processes such as accounting and error logging and logs the shutdown in the log file. It then sends the `init` program a signal that causes the system to transition to single-user mode.

If you do not specify a shutdown delay (`shutdown now`) only this message is broadcast before the system begins to shut down and user processes are killed.

6 As processes are stopped, notification messages are displayed to the console and are logged.

7 As the system halts, all login terminals (or graphical displays, such as CDE and XDM) are halted, and output is redirected to the console. Various system messages are displayed at the console as processes are shut down and the shutdown ends in single-user mode, displaying the superuser prompt (#). Now only the root user can use the system and perform standalone tasks or use the `halt` command to shut down the system completely.

8 Various messages are displayed as system components are initialized.

9 The console prompt (>>>) is displayed. Now you can turn off power to the system, reboot the system, or enter console commands.

2.11.3 Shutting Down and Halting the System

Use this procedure to shut down the system from multiuser mode, warn all users, and halt all systems. You also can invoke the SysMan Menu task "Shut Down the System" to perform the same operation.

1. Log in as root and change to the root directory:

   ```
   # cd /
   ```

2. Use the `shutdown` command to shut down and halt the system. For example, to shut down and halt the system in 5 minutes with a warning to users that the system is going down for maintenance, enter:

   ```
   # shutdown -h +5 /
   Maintenance shutdown in five minutes
   ```

The system begins to shut down as described in Example 2–1. However, the system also halts automatically and does not stop at the superuser prompt (#) . Instead, the console prompt is displayed and you can turn off power to the system, reboot, or use the console commands as described in the owner's manual for your system.

2.11.4 Shutting Down and Automatically Rebooting the System

Use this procedure to shut down the system from multiuser mode, warn all users, and automatically reboot the system to multiuser mode. You also can invoke the SysMan Menu task "Shut Down the System" to perform this operation.

1. Log in as root and change to the root directory:

   ```
   # cd /
   ```

2. Use the `shutdown` command to initiate a shut down followed by an automatic reboot. For example, to shut down and automatically reboot the system in 15 minutes with a warning to users that the system is going down for a reboot, enter the following command:

   ```
   # shutdown -r +15 \
   Shutdown and reboot in 15 minutes
   ```

The system begins to shut down as described in Example 2–1, notifying users of the impending shutdown, disabling logins, and then proceeds with the standard shutdown activities. When it completes these activities, the `shutdown` procedure automatically starts the reboot operation, which involves running the `fsck` command for a consistency check of all mounted file systems. If problems are not encountered, the system reboots to multiuser mode.

Note

If the `fsck` command finds file system inconsistencies, it displays a warning message recommending that you run the `fsck` command again from single-user mode before operating the system in multiuser mode.

2.11.5 Shutting Down and Halting Systems Immediately

Use the following procedure to shut down and halt the system immediately. Also, you can invoke the SysMan Menu task "Shut Down the System" to perform this operation:

1. Log in as root and change to the root directory. For example, enter the following command:

   # **cd /**

2. Enter the `shutdown` command as follows:

   # **shutdown -h now**

The system begins to shut down as described in Example 2–1 except that the shutdown is immediate and without prior warning to users. When all processes are shut down, the system is halted and the console prompt (>>>) is displayed. You can turn off power to the system, reboot it, or use the console commands as described in the owner's manual for your system.

Note

Use this form of the `shutdown` command if no other users are logged in to the system or if you need to shut down in an emergency. User processes are stopped without warning and you may lose user data.

2.12 Stopping Systems While in Single-User Mode

Although the `shutdown` command is your best choice for shutting down systems, there are other commands available (but not recommended) for stopping systems, namely: `halt`, `fasthalt`, `fastboot`, and `reboot`. Invoke these commands only from single-user mode.

If you are working in single-user mode, you can stop systems by entering the following commands:

```
# /sbin/sync
# /sbin/sync
# /usr/sbin/halt
```

The following events occur in response to the `halt` command:

- The shutdown is logged in the log file
- All running processes are killed
- A `sync` system call is issued
- All data is written to disk

- The system halts

Entering the sync command at least twice ensures that all data in memory is written safely to disk. See halt(8) for a description of the command and its flags.

See fasthalt(8), fastboot(8), and reboot(8) for more information on the other options.

2.12.1 Stopping and Rebooting Systems with the reboot Command

If you are working in single-user mode, you can safely shut down and automatically reboot your system to multiuser mode with the reboot command, as follows:

```
# /usr/sbin/reboot
```

When you run the reboot command without options, it stops all processes, synchronizes the disks, then initiates and logs the reboot. However, if you need to shut down and reboot the system abruptly, enter the following command:

```
# reboot -q
```

In response to this command, the system shuts down abruptly without stopping processes and performing other shutdown activities. The command initiates a reboot without logging the event. See reboot(8) for a description of the command and its flags.

2.12.2 Stopping Systems with the fasthalt Command

If you are working in single-user mode, you can halt a system immediately by using the fasthalt command as follows:

```
# /usr/sbin/fasthalt -n
```

When you invoke the fasthalt command without options, it halts the system and flags the subsequent reboot to prevent the execution of the fsck command. The program creates the fastboot file, then invokes the halt program. The system startup script contains instructions to look for the fastboot file. If present, the script removes the file and skips the invocation of the fsck command. If you invoke the command without the -l, -n, or -q flag, the halt program logs the shutdown by using the syslogd command and places a record of the shutdown in the login accounting file, /var/adm/wtmp.

See fasthalt(8) for more information.

2.12.3 Stopping Systems with the fastboot Command

If you are working in single-user mode and do not need to verify file systems, you can halt and reboot the systems with the `fastboot` command, as follows:

```
# /usr/sbin/fastboot
```

When you invoke the `fastboot` command without options, it creates a file named `/fastboot`, halts the system, then immediately reboots the system without verifying file systems by using the `fsck` command. See `fastboot`(8) for more information.

3

Customizing the System Environment

This chapter provides information that enables you to customize your system environment. During the initial installation and configuration of your system, you may have performed some of these tasks already. As your system needs change, you may need to perform some of these additional tasks to meet new workload requirements. For example, during installation, you created the initial swap space (virtual memory). If you add physical memory to a system, you may need to increase the swap space accordingly.

The following topics are covered in this chapter:

- A description of the system initialization files, which you use to initialize and control the system's run levels (Section 3.1)
- Information on using the national language directories to provide support for language-specific and country-specific programs (Section 3.2)
- A discussion of the internationalization features, which you tailor to support programmers and users developing and running programs for international audiences (Section 3.3)
- A discussion on the system time zone directories and environment variables, which you use to administer local and worldwide time zone information on your system (Section 3.4)
- A description of power management, which you set up and use to control power consumption in Energy Star-compliant peripherals and processors (Section 3.5)
- Information on how to customize swap space. See the *System Configuration and Tuning* manual as there are implications for performance tuning (Section 3.6)

See the following documents for information about customizing security and the network environment:

- The *Technical Overview* briefly describes the security components of the operating system.
- The *Security Administration* and *Security Programming* manuals are the principal source of security-related information for users, administrators, and programmers dealing with the security components.

- The *Network Administration: Connections* and *Network Administration: Services* manuals are the principal sources of information for customizing the system's networking components.

3.1 Identifying and Modifying the System Initialization Files

To define and customize the system environment, you modify certain initialization files that specify and control processes and run levels. The operating system provides you with default files that define the available run levels and the processes associated with each run level. You can change or customize the system environment easily by using these files as templates. In addition, if you support internationalization standards, you must be familiar with the structure and requirements of the corresponding files on your system.

The following sections describe this feature and provide instructions for identifying, using, and modifying the files that initialize and control the system environment. To understand and utilize available features, you should familiarize yourself with the `init` program and the specific files and commands associated with the program. See `init`(8) for a description of the program and its behavior.

Before you make any changes to the system initialization files, examine the default setup, evaluate the needs of your system, and make a copy of the entire set of default files. Taking precautions is wise when making changes to system files or to files that alter the working environment. If you discover that your modifications do not create the environment that you intended, you can reinstate the default files while you fix the problems in your customization.

The following system files and directories influence system startup and operation:

`/etc/inittab`	One of the key initialization files whose entries define run levels and associated processes and administer terminals. Section 3.1.1 describes this file.
`/etc/securettys`	A text file that marks whether a given terminal (`tty`) line allows root logins. Section 3.1.1.6 describes this file.
`/sbin/bcheckrc`	A system initialization run command script associated with verifying and mounting file systems at startup time. Section 3.1.1.2 describes this file.

`/sbin/init.d`	The initialization directory that contains executable files associated with system startup and the available run levels. Section 3.1.2.1 describes the directory structure and contents.
`/sbin/rc`*n* `.d`	The `/sbin` directory contains a set of individual subdirectories that correspond to the various run levels. Each subdirectory contains linked files that the system acts on when starting or changing a particular run level. There are three `/sbin/rc`*n* `.d` directories available: `/sbin/rc0.d`, `/sbin/rc2.d`, and `/sbin/rc3.d`. Section 3.1.2.2, Section 3.1.2.3, and Section 3.1.2.4 describe the `rc` directory structure and contents.
`/sbin/rc`*n*	These are the run command scripts that correspond to a particular run level. There are three `/sbin/rc`*n* scripts available: `/sbin/rc0`, `/sbin/rc2`, and `/sbin/rc3`. Section 3.1.2.2, Section 3.1.2.3, and Section 3.1.2.4 describe the contents and use of these scripts.
`/etc/rc.config` and `/etc/rc.config.common`	This is a file that contains run-time configuration variables. Scripts in the `/sbin/init.d` directory use these variables to configure various subsystems (for example, NFS or NTP). You (or a program) can use the `rcmgr` command to define or access variables in the `/etc/rc.config` file. See `rcmgr`(8) and the *Network Administration: Connections* and *Network Administration: Services* manuals for more information.
`/etc/sysconfigtab`	This is the database file that contains information about dynamically configurable subsystems. Chapter 4 describes this file.

`/usr/sbin/getty`	This is the executable file that sets and manages terminal lines. Section 3.1.1.3 and Section 3.1.1.4 describe this program. See `getty`(8) for more information.
`/etc/gettydefs`	The file used by `getty` that contains entries to identify and define terminal line attributes. See `gettydefs`(4) for more information.
`/var/spool/cron/crontabs/*`	These are the files that contain entries to identify and define the regular or periodic activation of specific processes. See Section 3.1.3 for more information about these files.
`/var/spool/cron/atjobs/*`	This is a file that contains entries to identify and define the once-only activation of specific processes. See `at`(1) for more information.

The following files contain information on kernel configuration:

`/usr/sys/conf/`*`NAME`*	This is a text file that defines the components that the system builds into your configuration. The *`NAME`* variable usually specifies the system name. Chapter 4 describes this file.
`/usr/sys/conf/`*`NAME`* `.list`	The optional configuration file that stores information about the layered product subsystems and is used to automatically configure static subsystems. The *`NAME`* variable usually specifies the system name. Chapter 4 describes this file.
`/usr/sys/conf/param.c`	The text file that contains default values for some tunable system parameters used in building the system's kernel. Chapter 4 describes this file.

3.1.1 Using the /etc/inittab File

One of the first actions taken by the `init` program is to read the `/etc/inittab` file. The `inittab` file supplies the `init` program with instructions for creating and running initialization processes. The `init` program reads the `inittab` file each time `init` is invoked. The file typically contains instructions for the default initialization, the creation and control of processes at each run level, and the `getty` line process that controls the activation of terminal lines.

The operating system provides you with a basic `/etc/inittab` file that contains line entries for the most common and necessary initialization processes. For example, the `/etc/inittab` file available with the distribution software would look similar to the following:

```
is:3:initdefault:
ss:Ss:wait:/sbin/rc0 shutdown </dev/console> \
    /dev/console 2>&1
 s0:0:wait:/sbin/rc0 off < /dev/console > /dev/console 2>&1
 fs:23:wait:/sbin/bcheckrc < /dev/console > /dev/console 2>&1
 kls:Ss:sysinit:/sbin/kloadsrv < /dev/console > /dev/console 2>&1
 hsd:Ss:sysinit:/sbin/hotswapd < /dev/console > /dev/console 2>&1
 sysconfig:23:wait:/sbin/init.d/autosysconfig start \
   < /dev/console > /dev/console 2>&1
 update:23:wait:/sbin/update > /dev/console 2>&1
 smsync:23:wait:/sbin/sysconfig -r vfs smoothsync-age=30 > \
   /dev/null 2>&1
 smsyncS:Ss:wait:/sbin/sysconfig -r vfs smoothsync-age=0 > \
   /dev/null 2>&1
 it:23:wait:/sbin/it < /dev/console > /dev/console 2>&1
 kmk:3:wait:/sbin/kmknod > /dev/console 2>&1
 s2:23:wait:/sbin/rc2 < /dev/console > /dev/console 2>&1
 s3:3:wait:/sbin/rc3 < /dev/console > /dev/console 2>&1
 cons:1234:respawn:/usr/sbin/getty console console vt100
```

The `inittab` file is composed of an unlimited number of lines. Each line in the `inittab` file contains four fields that are separated by a colon (`:`). The fields and syntax for entries in the `inittab` file are as follows:

Identifier: *Runlevel*: *Action*: *Command*

`Identifier`

This 14-character field uniquely identifies an object entry.

`Runlevel`

This 20-character field defines the run levels in which the object entry is to be processed. The *`Runlevel`* variable corresponds to a configuration of processes in a system. Each process spawned by the

`init` command is assigned one or more run levels in which it is allowed to exist. The run levels are as follows:

`0`	Specifies the halt state
`s` or `S`	Specifies single-user mode
`2`	Specifies multiuser mode without network services
`3`	Specifies multiuser mode with network services

The *`Runlevel`* field can define multiple run levels for a process by specifying more than one run level character in any combination.

`Action`

This 20-character field tells `init` how to treat the specified process. The most common actions that `init` recognizes are as follows:

`respawn`	If the process does not exist or dies, `init` starts it. If the process currently exists, `init` does nothing and continues scanning the `inittab` file.
`wait`	When `init` enters a run level that matches the run level of the entry, it starts the process and waits for its termination. While `init` continues in this run level, it does not act on subsequent reads of the entry in the `inittab` file.
`initdefault`	A line with this action is processed when `init` is first invoked. The `init` program uses this line to determine which run level to enter. To do this, it takes the highest run level specified in the run-level field and uses that as its initial state. If the run-level field is empty, this is interpreted as 0s23, so `init` enters run level 3. If `init` does not find an `initdefault` line in the `inittab` file, it requests an initial run level from the operator.

Other action keywords are available and recognized by the `init` program. See `inittab`(4) for more information.

Command

This is a data field limited to 1024 characters that contains sh commands. The entry in the command field is prefixed with exec. Any legal sh syntax can appear in the command field.

You can insert comments in the inittab file by specifying a # (number sign) at the beginning of a line. You can also place a \ (line continuation character) at the end of a line.

Before you modify or add entries to the /etc/inittab file, ensure that you are familiar with the function and contents of the associated files and the command scripts.

The following sections provide information to help you to use the /etc/inittab file.

3.1.1.1 Specifying the Initialization Default Run Level

At boot time, the init program examines the inittab file for the initdefault keyword to find the definition of the run level to enter. If there is no entry in inittab for initdefault, the system prompts you for a run level. In the previous inittab file example, the following line indicates that the run level for initdefault is set to 3, which is the multiuser with network services mode:

```
is:3:initdefault:
```

3.1.1.2 Specifying wait Run Levels

The init program looks in the inittab file for the wait entries. In the previous inittab file example, the following line contains a wait entry:

```
fs:23:wait:/sbin/bcheckrc < /dev/console > /dev/console 2>&1
```

In this case, the init program invokes the /sbin/bcheckrc script for the fs entry. Processes associated with this entry execute at run levels 2 and 3. Input comes from the system console (/dev/console). System and process error messages are sent to the console (> /dev/console 2>&1).

The bcheckrc run command script contains procedures associated with file system verification and mounting. See the /sbin/bcheckrc file for details.

3.1.1.3 Specifying Console Run Levels

Before you or anyone else can log in to your system, either the getty program or the xdm program must run. These programs set up a process that runs the login and shell programs for each terminal or workstation. Because a large portion of your initial work is done at the system console, the /etc/inittab file contains an entry for setting up a getty process

for the console. The xdm process is started by a run-level script in the /sbin/rc3.d directory.

In the example of the inittab file shown in Section 3.1.1, the following line contains the entry for the system console:

```
cons:1234:respawn:/usr/sbin/getty console console vt100
```

The init program is instructed to invoke the getty program, which sets the terminal line attributes for the system console (/dev/console). The run-level field specifies that the getty process should execute at run levels 1, 2, 3, and 4. The respawn keyword tells init to recreate the getty process if the active process terminates. If the process is active, init does not respawn the process; if it terminates, the process is recreated.

Note

In general, you should not modify the system console entry in the inittab file unless you want to limit the system console's access to different run levels. By placing limitations on the range of run levels for this terminal line, you risk disabling the system console if the system enters a run level that prohibits execution of the console's getty process.

3.1.1.4 Specifying Terminals and Terminal Run Levels

To enable user logins at each terminal supported by your system, you must maintain support for the terminal types available at your site and define the run level and getty process for each supported terminal type. Use the following database and file:

- The /usr/lib/terminfo database (a symbolic link to /usr/share/lib/terminfo) defines the various terminal types.
- Entries in the /etc/inittab file define the run level and getty process for the supported terminal types.

The operating system supports a wide variety of terminal types. The terminfo database contains entries that describe each terminal type and its capabilities. The database is created by the tic program, which compiles the source files into data files. The terminfo source files typically consist of at least one device description that conforms to a particular format. See terminfo(4) for specific details on creating and compiling source files.

The /usr/lib/terminfo directory contains the source files, each of which has a .ti suffix, for example *name*.ti. After you compile the source files with the tic command, it places the output in a directory subordinate to /usr/lib/terminfo.

Various commands and programs rely on the files in these directories. Set your TERMINFO environment variable to the /usr/lib/terminfo directory to instruct programs that rely on the database for information to look there for relevant terminal information.

See getty(8), gettydefs(4), and inittab(4) for information about defining terminal lines and managing terminal access.

3.1.1.5 Specifying Process Run Levels

Specific entries in the inittab file define the run command scripts that are to be executed when the system enters or changes to a particular run level. For example, the following inittab file entries specify the action to be taken by the init program at each of the available run levels:

```
ss:Ss:wait:/sbin/rc0 shutdown < /dev/console > /dev/console 2>&1
s0:0:wait:/sbin/rc0 off < /dev/console > /dev/console 2>&1
s2:23:wait:/sbin/rc2 < /dev/console > /dev/console 2>&1
s3:3:wait:/sbin/rc3 < /dev/console > /dev/console 2>&1
```

These entries are associated with the rc directory structure and are discussed in detail in Section 3.1.2.

3.1.1.6 Securing a Terminal Line

The /etc/securettys file indicates to the system whether terminals or pseudoterminals can be used for root logins. To enable root logins on a terminal line, include the path name in the /etc/securettys file. To enable root login on pseudoterminals, include the ptys keyword. You enable X displays for root login by including their display name, for example :0. By default, only the console and the X server line are set secure.

The following example of an /etc/securettys file shows root logins enabled on the console, on the X display, on two hard-wired or LAT lines, and on all pseudoterminals:

```
/dev/console
:0
/dev/tty00
/dev/tty01
ptys
```

3.1.2 Using the init and rc Directory Structure

The operating system provides you with an initialization and run command directory structure. The structure has four main components:

- The init.d directory
- The rc0.d directory

- The rc2.d directory
- The rc3.d directory

In addition, each of the rc*n* .d directories has a corresponding rc*n* run command script.

3.1.2.1 The init.d Directory

The /sbin/init.d directory contains the executable files associated with system initialization. For example, a listing of the directory contents would look similar to the following:

```
.mrg..autosysconfig    evm                    recpasswd
.new..autosysconfig    gateway                rmtmpfiles
.new..rmtmpfiles       inet                   route
.proto..autosysconfig  inetd                  rwho
.proto..rmtmpfiles     insightd               savecore
admincheck             kmod                   security
advfsd                 lat                    sendmail
asudllink              lpd                    settime
asudna                 mfsmount               sia
asunbelink             motd                   smauth
asutcp                 ms_srv                 smsd
audit                  named                  snmpd
autosysconfig          netrain                startlmf
bin                    nfs                    streams
binlog                 nfsmount               syslog
crashdc                niffd                  timed
cron                   nis                    uucp
dhcp                   paging                 write
dia_s_k                preserve               ws
enlogin                presto                 xlogin
envmon                 quota                  xntpd
```

3.1.2.2 The rc0.d Directory and rc0 Run Command Script

The /sbin/rc0 script contains run commands that enable a smooth shutdown and bring the system to either a halt state or single-user mode. As described previously, the inittab file contains entries that the init program reads and acts on when the system is shutting down to single-user mode (level s) or halting (level 0). For example:

```
ss:Ss:wait:/sbin/rc0 shutdown < /dev/console > /dev/console 2>&1
s0:0:wait:/sbin/rc0 off < /dev/console > /dev/console 2>&1
```

In both cases, the rc0 script is the specified command. In addition to commands listed in the script itself, rc0 contains instructions to run commands found in the /sbin/rc0.d directory. These commands are linked to files in the init.d directory. The script defines the conditions under

which the commands execute; some commands run if the system is being halted while others run if the system is being shut down and rebooted to single-user mode.

By convention, files in the `/sbin/rc0.d` directory begin with either the letter "K" or the letter "S" and are followed by a 2-digit number and a file name. For example, a long listing of the `rc0.d` directory contents would look similar to the following:

```
lrwxr-xr-x 1 root bin    17 May  8 16:35 K00enlogin -> ../init.d/enlogin
lrwxrwxrwx 1 root bin    16 May 10 10:05 K02.0ms_srv -> ../init.d/ms_srv
lrwxrwxrwx 1 root bin    16 May 10 10:03 K02.1asutcp -> ../init.d/asutcp
lrwxrwxrwx 1 root bin    20 May 10 10:03 K02.2asunbelink -> \
                                              ../init.d/asunbelink
lrwxrwxrwx 1 root bin    16 May 10 10:03 K02.3asudna -> ../init.d/asudna
lrwxrwxrwx 1 root bin    19 May 10 10:03 K02.4asudllink -> \
                                               ../init.d/asudllink
lrwxrwxrwx 1 root bin    13 May  8 16:39 K05lpd -> ../init.d/lpd
lrwxrwxrwx 1 root bin    13 May 10 11:06 K07lat -> ../init.d/lat
lrwxr-xr-x 1 root bin    15 May  8 16:35 K08audit -> ../init.d/audit
lrwxrwxrwx 1 root bin    14 May 10 11:06 K09dhcp -> ../init.d/dhcp
lrwxr-xr-x 1 root bin    15 May  8 16:37 K10inetd -> ../init.d/inetd
lrwxr-xr-x 1 root bin    15 May  8 16:37 K14snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root system 16 May 10 11:06 K16envmon -> ../init.d/envmon
lrwxr-xr-x 1 root bin    16 May  8 16:37 K19xlogin -> ../init.d/xlogin
lrwxr-xr-x 1 root bin    15 May  8 16:37 K20xntpd -> ../init.d/xntpd
lrwxr-xr-x 1 root bin    15 May  8 16:37 K21timed -> ../init.d/timed
lrwxr-xr-x 1 root bin    14 May  8 16:35 K22cron -> ../init.d/cron
lrwxr-xr-x 1 root bin    18 May  8 16:35 K25sendmail -> \
                                              ../init.d/sendmail
lrwxrwxrwx 1 root bin    13 May  8 16:37 K30nfs -> ../init.d/nfs
lrwxr-xr-x 1 root bin    16 May  8 16:35 K31presto -> ../init.d/presto
lrwxrwxrwx 1 root bin    18 May  8 16:37 K35nfsmount -> \
                                              ../init.d/nfsmount
lrwxr-xr-x 1 root bin    13 May  8 16:37 K38nis -> ../init.d/nis
lrwxrwxrwx 1 root bin    15 May 10 11:06 K40named -> ../init.d/named
lrwxr-xr-x 1 root bin    14 May  8 16:37 K42rwho -> ../init.d/rwho
lrwxr-xr-x 1 root bin    15 May  8 16:37 K43route -> ../init.d/route
lrwxr-xr-x 1 root bin    17 May  8 16:37 K44gateway -> \
                                                ../init.d/gateway
lrwxr-xr-x 1 root bin    16 May  8 16:35 K45syslog -> ../init.d/syslog
lrwxrwxrwx 1 root bin    14 May 10 11:07 K46uucp -> ../init.d/uucp
lrwxr-xr-x 1 root bin    15 May  8 16:35 K47write -> ../init.d/write
lrwxr-xr-x 1 root bin    16 May  8 16:35 K48binlog -> ../init.d/binlog
lrwxr-xr-x 1 root bin    14 May  8 16:37 K50inet -> ../init.d/inet
lrwxr-xr-x 1 root bin    17 May  8 16:37 K50netrain -> \
                                               ../init.d/netrain
lrwxr-xr-x 1 root bin    15 May  8 16:37 K51niffd -> ../init.d/niffd
lrwxr-xr-x 1 root bin    15 May  8 16:35 K52quota -> ../init.d/quota
lrwxr-xr-x 1 root bin    13 May  8 16:35 K95evm -> ../init.d/evm
lrwxr-xr-x 1 root bin    14 May  8 16:35 K96acct -> ../init.d/acct
```

In general, the system starts commands that begin with the letter "S" and stops commands that begin with the letter "K." The numbering of commands in the `/sbin/rc0.d` directory is important because the numbers are sorted and the commands are run in ascending order.

See `rc0`(8) for more information.

3.1.2.3 The rc2.d Directory and rc2 Run Command Script

The /sbin/rc2 script contains run commands that enable initialization of the system run level 2 (multiuser, but disconnected from the network). The inittab file contains entries that are read by the init program. The init program reads and acts on the inittab file entries when the system is booting or changing its state to run level 2. For example:

```
s2:23:wait:/sbin/rc2 < /dev/console > /dev/console 2>&1
```

The rc2 script is the specified command. In addition to commands listed in the script itself, rc2 contains instructions to run commands found in the /sbin/rc2.d directory. These commands are linked to files in the init.d directory. The script defines the conditions under which the commands execute; some commands run if the system is booting, other commands run if the system is changing run levels.

By convention, files in the /sbin/rc2.d directory begin with either the letter "K" or the letter "S" and are followed by a 2-digit number and a file name. For example, a listing of the /sbin/rc2.d directory contents would look similar to the following:

```
lrwxr-xr-x 1 root  bin    17 May  8 16:35 K00enlogin -> ../init.d/enlogin
lrwxrwxrwx 1 root  bin    16 May 10 10:05 K02.0ms_srv -> ../init.d/ms_srv
lrwxrwxrwx 1 root  bin    16 May 10 10:03 K02.1asutcp -> ../init.d/asutcp
lrwxrwxrwx 1 root  bin    20 May 10 10:03 K02.2asunbelink -> \
                                                  ../init.d/asunbelink
lrwxrwxrwx 1 root  bin    16 May 10 10:03 K02.3asudna -> ../init.d/asudna
lrwxrwxrwx 1 root  bin    19 May 10 10:03 K02.4asudllink -> \
                                                    ../init.d/asudllink
lrwxrwxrwx 1 root  bin    13 May  8 16:39 K05lpd -> ../init.d/lpd
lrwxrwxrwx 1 root  bin    13 May 10 11:06 K07lat -> ../init.d/lat
lrwxr-xr-x 1 root  bin    15 May  8 16:35 K08audit -> ../init.d/audit
lrwxrwxrwx 1 root  bin    14 May 10 11:06 K09dhcp -> ../init.d/dhcp
lrwxr-xr-x 1 root  bin    15 May  8 16:37 K10inetd -> ../init.d/inetd
lrwxr-xr-x 1 root  bin    15 May  8 16:37 K14snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root  system        16 May 10 11:06 K16envmon -> \
                                                      ../init.d/envmon
lrwxr-xr-x 1 root  bin    16 May  8 16:37 K19xlogin -> ../init.d/xlogin
lrwxr-xr-x 1 root  bin    15 May  8 16:37 K20xntpd -> ../init.d/xntpd
lrwxr-xr-x 1 root  bin    15 May  8 16:37 K21timed -> ../init.d/timed
lrwxr-xr-x 1 root  bin    14 May  8 16:35 K22cron -> ../init.d/cron
lrwxr-xr-x 1 root  bin    18 May  8 16:35 K25sendmail -> \
                                                    ../init.d/sendmail
lrwxrwxrwx 1 root  bin    13 May  8 16:37 K30nfs -> ../init.d/nfs
lrwxr-xr-x 1 root  bin    16 May  8 16:35 K31presto -> ../init.d/presto
lrwxrwxrwx 1 root  bin    18 May  8 16:37 K35nfsmount -> \
                                                    ../init.d/nfsmount
lrwxr-xr-x 1 root  bin    13 May  8 16:37 K38nis -> ../init.d/nis
lrwxrwxrwx 1 root  bin    15 May 10 11:06 K40named -> ../init.d/named
lrwxr-xr-x 1 root  bin    14 May  8 16:37 K42rwho -> ../init.d/rwho
lrwxr-xr-x 1 root  bin    15 May  8 16:37 K43route -> ../init.d/route
lrwxr-xr-x 1 root  bin    17 May  8 16:37 K44gateway -> \
                                                     ../init.d/gateway
lrwxr-xr-x 1 root  bin    16 May  8 16:35 K45syslog -> ../init.d/syslog
```

In general, the system starts commands that begin with the letter "S" and stops commands that begin with the letter "K." Commands that begin with

the letter "K" run only when the system is changing run levels from a higher to a lower level. Commands that begin with the letter "S" run in all cases. The numbering of commands in the /sbin/rc2.d directory is important because the numbers are sorted and the commands are run in ascending order.

See rc2(8) for more information.

3.1.2.4 The rc3.d Directory and rc3 Run Command Script

The /sbin/rc3 script contains run commands that enable initialization of the system to a networked multiuser state, run level 3. As described previously, the inittab file contains entries that the init program reads and acts on when the system is booting or changing its state to run level 3. For example:

```
s3:3:wait:/sbin/rc3 < /dev/console > /dev/console 2>&1
```

The rc3 script is the specified command. In addition to commands listed in the script itself, rc3 contains instructions to run commands found in the /sbin/rc3.d directory. These commands are linked to files in the init.d directory. The script defines the conditions under which the commands execute; some commands run if the system is booting, other commands run if the system is changing run levels.

By convention, files in the /sbin/rc3.d directory begin with the letter "S" and are followed by a 2-digit number and a file name. For example, a long listing of the rc3.d directory contents would look similar to the following:

```
lrwxr-xr-x 1 root  bin    15 May  8 16:37 S00cniffd -> ../init.d/niffd
lrwxr-xr-x 1 root  bin    17 May  8 16:37 S00fnetrain -> ../init.d/netrain
lrwxr-xr-x 1 root  bin    14 May  8 16:37 S00inet -> ../init.d/inet
lrwxr-xr-x 1 root  bin    15 May  8 16:35 S01quota -> ../init.d/quota
lrwxrwxrwx 1 root  bin    14 May 10 11:07 S04uucp -> ../init.d/uucp
lrwxr-xr-x 1 root  bin    18 May  8 16:35 S08startlmf -> ../init.d/startlmf
lrwxr-xr-x 1 root  bin    16 May  8 16:35 S09syslog -> ../init.d/syslog
lrwxr-xr-x 1 root  bin    16 May  8 16:35 S10binlog -> ../init.d/binlog
lrwxr-xr-x 1 root  bin    17 May  8 16:37 S11gateway -> ../init.d/gateway
lrwxr-xr-x 1 root  bin    15 May  8 16:37 S12route -> ../init.d/route
lrwxr-xr-x 1 root  bin    14 May  8 16:37 S13rwho -> ../init.d/rwho
lrwxr-xr-x 1 root  bin    17 May  8 16:35 S14settime -> ../init.d/settime
lrwxrwxrwx 1 root  bin    15 May 10 11:06 S15named -> ../init.d/named
lrwxr-xr-x 1 root  bin    13 May  8 16:37 S18nis -> ../init.d/nis
lrwxrwxrwx 1 root  bin    13 May  8 16:37 S19nfs -> ../init.d/nfs
lrwxrwxrwx 1 root  bin    18 May  8 16:37 S20nfsmount -> ../init.d/nfsmount
lrwxr-xr-x 1 root  bin    15 May  8 16:35 S21audit -> ../init.d/audit
lrwxr-xr-x 1 root  bin    18 May  8 16:35 S25preserve -> ../init.d/preserve
lrwxr-xr-x 1 root  bin    20 May  8 16:35 S30rmtmpfiles -> ../init.d/rmtmpfiles
lrwxr-xr-x 1 root  bin    16 May  8 16:35 S36presto -> ../init.d/presto
lrwxr-xr-x 1 root  bin    18 May  8 16:35 S40sendmail -> ../init.d/sendmail
lrwxr-xr-x 1 root  bin    15 May  8 16:37 S45xntpd -> ../init.d/xntpd
lrwxr-xr-x 1 root  bin    15 May  8 16:37 S46timed -> ../init.d/timed
lrwxr-xr-x 1 root  bin    15 May  8 16:37 S49snmpd -> ../init.d/snmpd
lrwxrwxrwx 1 root  bin    18 May  8 16:44 S50insightd -> ../init.d/insightd
lrwxrwxrwx 1 root  system 16 May 10 11:06 S51envmon -> ../init.d/envmon
```

```
lrwxrwxrwx 1 root  bin    16 May  8 16:41 S53advfsd -> ../init.d/advfsd
lrwxr-xr-x 1 root  bin    15 May  8 16:37 S55inetd -> ../init.d/inetd
lrwxrwxrwx 1 root  bin    14 May 10 11:06 S56dhcp -> ../init.d/dhcp
lrwxr-xr-x 1 root  bin    14 May  8 16:35 S57cron -> ../init.d/cron
lrwxrwxrwx 1 root  bin    13 May 10 11:06 S58lat -> ../init.d/lat
lrwxr-xr-x 1 root  bin    14 May  8 16:35 S60motd -> ../init.d/motd
lrwxrwxrwx 1 root  bin    19 May 10 10:03 S61.0asudllink -> \
                                            ../init.d/asudllink
lrwxrwxrwx 1 root  bin    16 May 10 10:03 S61.1asudna -> ../init.d/asudna
lrwxrwxrwx 1 root  bin    20 May 10 10:03 S61.2asunbelink -> \
                                           ../init.d/asunbelink
lrwxrwxrwx 1 root  bin    16 May 10 10:03 S61.3asutcp -> ../init.d/asutcp
lrwxrwxrwx 1 root  bin    16 May 10 10:05 S61.4ms_srv -> ../init.d/ms_srv
lrwxr-xr-x 1 root  bin    15 May  8 16:35 S63write -> ../init.d/write
lrwxrwxrwx 1 root  bin    13 May  8 16:39 S65lpd -> ../init.d/lpd
lrwxr-xr-x 1 root  bin    17 May  8 16:35 S80crashdc -> ../init.d/crashdc
lrwxr-xr-x 1 root  bin    12 May  8 16:45 S90ws -> ../init.d/ws
lrwxr-xr-x 1 root  bin    16 May  8 16:37 S95xlogin -> ../init.d/xlogin
lrwxr-xr-x 1 root  bin    13 May  8 16:35 S97evm -> ../init.d/evm
lrwxr-xr-x 1 root  bin    16 May  8 16:35 S98smauth -> ../init.d/smauth
lrwxr-xr-x 1 root  bin    20 May  8 16:35 S99admincheck -> \
                                          ../init.d/admincheck
lrwxr-xr-x 1 root  bin    14 May  8 16:38 S99smsd -> ../init.d/smsd
```

In general, the system starts commands that begin with the letter "S" and stops commands that begin with the letter "K." Commands that begin with the letter "K" run only when the system is changing run levels from a higher to a lower level. Commands that begin with the letter "S" run in all cases.

Usually, only commands that begin with the letter "S" are placed in the `rc3.d` directory. By default, run level 3 is the highest run level. The numbering of commands in the `/sbin/rc3.d` directory is important because the numbers are sorted and the commands are run in ascending order.

See `rc3`(8) for more information.

3.1.3 Using the crontabs Directory

The `crontab` command submits a schedule of commands to the `cron` system clock daemon. The `cron` daemon runs shell commands according to the dates and times specified in the files in the `/var/spool/cron/crontabs` directory. Commands that you want to run on a regular schedule are in these files. Commands that you want to run only once are in the `/var/spool/cron/atjobs/*` files and are submitted with the `at` command.

The following example of an entry from a file in the `/var/spool/cron/crontabs` directory specifies that the `runacct` command runs at 2:00 A.M., Monday through Saturday, and output is sent to the `/var/adm/acct/nite/fd2log` file:

```
  1                  2                        3
0 2 * * 1-6 /usr/sbin/acct/runacct > /var/adm/acct/nite/fd2log&
```

Each entry has the following syntax:

1. Specifies the minutes past the hour, the hour, day of month, month, and day of week. For the day of week, the value 0 (zero) indicates Sunday, the value 1 indicates Monday, and so on. You can specify a single value, more than one value separated by commas, or two values separated by a dash (–) to indicate a range of values. You can also specify an asterisk (*) to indicate no specific value. For example, if an asterisk (*) is specified for the hour, the command is run every hour.
2. Specifies the command to be executed at the specified time.
3. Specifies, optionally, arguments to the command.

To add a comment to a file, specify a # (number sign) at the beginning of the line.

The files in the /var/spool/cron/crontabs directory are named for system users, and the commands in the files are run under the authority of the user. For example, the commands in the adm file are run under adm authority.

To use the crontab command, you must be the user that matches the file name you want to act upon. For example, if you are user adm and you run the crontab command, the action is performed on the /var/spool/cron/crontabs/adm file.

To submit commands to the cron daemon to be run under adm authority:

1. Become user adm.
2. Enter the crontab command with the –l option to copy the /usr/spool/cron/crontabs/adm file to a temporary file in your home directory.

   ```
   % crontab –l > temp_adm
   ```
3. Edit the temporary file and add the commands you want to run at a specified time.
4. Enter the crontab command and specify the temporary file to submit the commands to the cron daemon.

   ```
   % crontab temp_adm
   ```

The /var/adm/cron/log file contains a history of the commands executed by the cron daemon.

You can use the /usr/spool/cron/crontabs/root file to back up and clean system log files. The root crontab file /usr/var/spool/cron/crontabs/root contains a model entry to clean up the /var/adm/wtmp log file at 2:00 A.M. every Sunday. One compressed

backup of the log file is retained until the next cleaning. This crontab entry is enabled by default as follows:

```
# To get the standard output by e-mail remove the output redirection.
#
0 2 * * 0 /usr/lbin/logclean /var/adm/wtmp > /dev/null
```

Add additional tasks, or modify the existing task, to suit your local system requirements.

In the preceding example, output is directed to /dev/null by default. You can redirect it to an e-mail address to receive notification when a task finishes. This cron task backs up the login log file and creates a new empty file. (The login log records all user logins on the system.)

If you want to preserve your log files for a longer period of time, you can either change the frequency of the cleanup or comment out the applicable ./crontabs/root entry. Also, you may want to create cleanup cron tasks for other system log files, such as those relating to print services.

To edit the root crontab file, you must be root (superuser) and you should use only the following command:

```
# crontab -e
```

The environment variable EDITOR should be set and exported beforehand if you want to use an editor other than /usr/bin/ed.

See crontab(1) for more information.

3.2 Using National Language Support

The operating system provides language-specific and country-specific information or support for programs.

The support components that concern you most directly as system administrator are the directories and files that reside at /usr/lib/nls.

An internationalized system presents information in a variety of ways. The word *locale* refers to the language, territory, and code set requirements that correspond to a particular part of the world. The system stores locale-specific data in two kinds of files:

Locale files	These files contain month and day names, date formats, monetary and numeric formats, valid yes/no strings, character classification data, and collation sequences. These files reside in the /usr/lib/nls/loc directory.

Message catalogs — These files contain translations of messages that are used by programs. These files reside in the `/usr/lib/nls/msg/`*`locale-name`* directory.

Table 3–1 lists examples of the locales moved to the `/usr/lib/nls/loc` directory when you install the optional Single-Byte European Locales subset. Additional locales are installed by language variant subsets with special licensing requirements.

Table 3–1: Locale Support Files

Language/Territory	Locale Filename
Danish-Denmark	da_DK.ISO8859-1
Dutch-Netherlands	nl_NL.ISO8859-1
Dutch_Belgium	nl_BE.ISO8859-1
English_U.K	en_GB.ISO8859-1
English_U.S.A.	en_US.ISO8859-1
Finnish-Finland	fi_FI.ISO8859-1
French_Belgium	fr_BE.ISO8859-1
French_Canada	fr_CA.ISO8859-1
French_France	fr_FR.ISO8859-1

Note

The `/usr/lib/nls/loc` directory also contains environment tables (.en files) and character tables (.8859* files) that correspond to some of the files listed in Table 3–1. These tables and variants are provided only to ensure system compatibility for old programs and should not be used by new applications.

For more information on internationalization options, and features provided to support the development of international software, see:

`code_page`(5) — Lists the coded character sets that are used on Microsoft Windows and Windows NT systems.

`iconv_intro`(5) — Provides an introduction to codeset conversion.

`iconv`(1) — Documents the command to convert encoded characters to another codeset.

`i18n_intro`(5)	Provides an introduction to internationalization (I18N).
`i18n_printing`(5)	Provides an introduction to internationalization (I18N) printer support.
`l10n_intro`(5)	Provides an introduction to localization (L10N).
`locale`(1)	Provides information about locales.

This is not a definitive list of all the reference pages that document internationalization. The See Also section of each reference page, and the *Writing Software for the International Market* manual are definitive sources.

3.2.1 Setting Locale

The default system-wide locale for internationalization is the C locale. The default system-wide locale is the one that the `setlocale` function uses when a user does not set the internationalization environment variables, such as LANG, LC_COLLATE, and so on.

To change the system-wide default locale for Bourne and Korn shell users, edit the `/etc/profile` file and include the name of the locale you want to be the system-wide default. Then the `setlocale` function uses the locale specified in this file. Those using the C shell can set a system-wide locale by editing the `/etc/csh.login` file and including the name of the locale you want to be the default system-wide locale.

You can set the native locale to any of the locales in the `/usr/lib/nls/loc` directory.

To set a locale, assign a locale name to one or more environment variables in the appropriate shell startup file. The simplest way is to assign a value to the LANG environment variable because it covers all components of a locale.

Note

The C locale is the system default. The C locale specifies U.S. English and uses the 7-bit ASCII codeset. The main difference between the C locale and the U.S. English locale (`en_US.ISO8859-1`) is that the latter has enhanced error messages.

The following example sets the locale to French for the C shell in which it is invoked and for all child processes of that shell:

```
% setenv LANG fr_FR.ISO8859-1
```

If you want another shell to have a different locale, you can reset the LANG environment variable in that particular shell. The following example sets the locale to French for the Korn and Bourne shells:

```
$ LANG=fr_FR.ISO8859-1
$ export LANG
```

Setting the LANG environment variable on the command line sets the locale for the current process only.

In most cases, assigning a value to the LANG environment variable is the only thing you need to do to set the locale. This is because when you set the locale with the LANG environment variable, the appropriate defaults are automatically set for the following functions:

- Collation
- Character classification
- Date and time conventions
- Numeric and monetary formats
- Program messages
- Yes/no prompts

In the unlikely event that you need to change the default behavior of any of the previous categories within a locale, you can set the variable that is associated with that category. See the following section for more information.

3.2.2 Modifying Locale Categories

When you set the locale with the LANG environment variable, defaults are set automatically for the collation sequence, character classification functions, date and time conventions, numeric and monetary formats, program messages, and the yes/no prompts appropriate for that locale. However, should you need to change any of the default categories, you can set the environment variables that are associated with one or more categories.

Table 3–2 describes the environment variables that influence locale categories.

Table 3–2: Locale Environment Variables

Environment Variable	Description
LC_ALL	Overrides the setting of all other internationalization environment variables, including LANG.
LC_COLLATE	Specifies the collating sequence to use when sorting names and when character ranges occur in patterns.

Table 3–2: Locale Environment Variables (cont.)

Environment Variable	Description
LC_CTYPE	Specifies the character classification information to use.
LC_NUMERIC	Specifies the numeric format.
LC_MONETARY	Specifies the monetary format.
LC_TIME	Specifies the date and time format.
LC_MESSAGES	Specifies the language in which system messages appear. In addition, specifies the strings that indicate "yes" and "no" in yes/no prompts.

As with the LANG environment variable, you can assign locale names to all the category variables. For example, suppose that your company's main language is Spanish. You can set the locale with the LANG environment variable for Spanish, but set the numeric and monetary format for U.S. English. To do this for the C shell, you would make the following variable assignments:

```
% setenv LANG es_ES.ISO8859-1
% setenv LC_NUMERIC en_US.ISO8859-1
% setenv LC_MONETARY en_US.ISO8859-1
```

Locale names may include *@modifiers* to indicate versions of the locales that meet special requirements for different categories.

For example, a locale may exist in two versions to sort data two ways: in dictionary order and in telephone-book order. Suppose your site is in France, and it uses the default French locale, and suppose the standard setup for this locale uses dictionary order. However, in this example, your site also needs to use a site-defined locale that collates data in telephone-book order. You may set your environment variables for the C shell as follows:

```
% setenv LANG fr_FR.ISO8859-1
% setenv LC_COLLATE fr_FR.ISO8859-1@phone
```

The explicit setting of LC_COLLATE overrides LANG's implicit setting of that portion of the locale.

3.2.3 Limitations of Locale Variables

The LANG and LC_* environment variables allow you to set the locale the way you want it, but they do not protect you from mistakes. There is nothing to protect you from setting LANG to a Swedish locale and LC_CTYPE to a Portuguese locale.

Also, there is no way to tie locale information to data. This means that the system has no way of knowing what locale you set when you created a file, and it does not prevent you from processing that data in inappropriate ways

later. For example, suppose LANG was set to a German locale when you created file `foo`. Now suppose you reset LANG to a Spanish locale and then use the `grep` command for something in `foo`. The `grep` command uses Spanish rules on the German data in the file.

3.2.4 Setting Environment Variables for Message Catalogs and Locales

To define the location of message catalogs, set the NLSPATH environment variable. The default path is as follows:

```
NLSPATH=/usr/lib/nls/msg/%L/%N:
```

In this example, `%L` specifies the current locale name, and `%N` specifies the value of name of the message catalog.

There is also a LOCPATH environment variable that defines the search path for locales. The default path is as follows:

```
LOCPATH=/usr/lib/nls/loc:
```

3.3 Customizing Internationalization Features

The operating system provides many internationalization features. You, or your local site planners, determine which elements of the operating system's internationalization features (commonly called worldwide support features) are required. The worldwide support features are optional subsets that you can select during installation. Your job as an administrator is to set up and maintain these features for:

- Software developers who produce internationalized applications
- Users who run internationalized applications on your system

There are three sources of information about worldwide support:

- For a list of optional software subsets that support internationalization, see the *Installation Guide*.
- For information about setting up and maintaining an operating system environment for programmers who write internationalized software, see the *Writing Software for the International Market* manual.
- To set up and maintain your system for users of internationalized applications, see the System Setup graphical user interface and click on the Configuration icon and then the internationalization icon. From the internationalization window, you can select tasks to configure or modify several of the worldwide support capabilities on your system. To make this option available, you must install at least one international support software subset. You also can launch this option from the CDE Application Manager. See Chapter 1 for information on using CDE.

3.4 Customizing Your Time Zone

Time zone information is stored in files in the `/etc/zoneinfo` directory. The `/etc/zoneinfo/localtime` file is linked to a file in the `/etc/zoneinfo` directory and specifies the local time zone. These files are linked during system installation, but, as superuser, you can change your local time zone by relinking the `/etc/zoneinfo/localtime` file. For example, the following command changes the local time zone to be consistent with the city of New York on the American continent:

```
# ln -sf /etc/zoneinfo/America/New_York /etc/zoneinfo/localtime
```

The `/etc/zoneinfo/sources` directory contains source files that specify the worldwide time zone and daylight savings time information that is used to generate the files in the `/etc/zoneinfo` directory. You can change the information in the source files and then use the `zic` command to generate a new file in the `/etc/zoneinfo` directory. See `zic`(8) for more information on the format of the time zone database files.

You also can change the default time zone information by setting the TZ environment variable in your `.login` file or shell environment file. If you define the TZ environment variable, its value overrides the default time zone information specified by `/etc/zoneinfo/localtime`. By default, the TZ variable is not defined.

The TZ environment variable has the following syntax:

stdoffset [*dst*[offset] [,*start*[/*time*], *end*[/*time*]]]

You also can specify the following syntax:

stdoffset [*dst*[*offset*]]

The TZ environment variable syntaxes have the following parameters:

std and *dst*
: Specifies the three or more characters that designate the standard (*std*) or daylight savings time (*dst*) zone.

  Note

  Daylight savings time is called daylight summer time in some locales.

  The *dst* variable is not specified, daylight savings time does not apply. You can specify any uppercase and lowercase letters. A leading colon (:), comma (,), hyphen (-), plus sign(+), and ASCII NUL are not allowed.

offset

Specifies the value to be added to the local time to arrive at GMT. The *offset* variable uses 24-hour time and has the following syntax:

hh [:*mm* [:*ss*]]

If you do not specify the *offset* variable after the *dst* variable, daylight savings time is assumed to be 1 hour ahead of standard time. You can specify a minus sign (–) before the *offset* variable to indicate that the time zone is east of the prime meridian; west is the default, which you can specify with a plus sign (+).

start and *end*

Specifies when daylight savings time starts and ends. The *start* and *end* variable has the following syntaxes:

J*j*
n

M*m.w.d*

In the first syntax, the *j* variable specifies the Julian day, which is between 1 and 365. The extra day in a leap year (February 29) is not counted.

In the second syntax, the *n* variable specifies the zero-based Julian day, which is between zero (0) and 365. The extra day in a leap year is counted.

In the third syntax, the *m* variable specifies the month number (from 1 to 12), the *w* variable specifies the week number (from 1 to 5), and the *d* variable specifies the day of the week (from 0 to 6), where zero (0) specifies Sunday and six (6) specifies Saturday.

time

Specifies the time, in local time, when the change occurs to or from daylight savings time. The *time* variable uses 24-hour time and has the following syntax:

hh [:*mm* [:*ss*]]

The default is 02:00:00.

The following example of the TZ environment variable specification specifies:

- EST (eastern standard time) specifies the standard time, which is 5 hours behind GMT.

- EDT (eastern daylight time) specifies the daylight savings time, which is 4 hours behind GMT.
- EDT starts on the first Sunday in April and ends on the last Sunday in October; the change to and from daylight savings time occurs at 2:00 A.M., which is the default time.

```
EST5EDT4,M4.1.0,M10.5.0
```

You can specify the following syntax:

:pathname

The *pathname* variable specifies the pathname of a file that is in the `tzfile` file format and that contains the time conversion information. For example:

```
:America/New_York
```

See `tzfile`(4) for more information on the file format.

If the pathname begins with a slash (/), it specifies an absolute pathname; otherwise, the pathname is relative to the `/etc/zoneinfo` directory. If the specified file is unavailable or corrupted, the system defaults to Greenwich Mean Time (GMT).

The time zone formats differ for SVID 2 and SVID 3. For SVID 2, `/usr/sbin/timezone` creates the `/etc/svid2_tz` file. The contents of the TZ and TZC variables are based on the information you supply when you run `/usr/sbin/timezone`.

For SVID 3, the `/etc/svid3_tz` file is created during the installation process. The contents of the TZ variable is based upon answers you supply to time zone-related questions at installation time.

See `timezone`(3) for more information.

3.5 Customizing Power Management

The operating system contains features that allow you to conserve power on certain systems that have the appropriate hardware. Read the system owner's manual for information on whether your system supports power management. Power management utilities allow you to:

- Enable energy-saving features on supported monitors (Energy Star) and control the power modes and idle time.
- Select which disks you want to spin down after a selected idle time. Some systems are delivered for use with certain energy saving capabilities enabled by default. If disk drives spin down unexpectedly or data transfer sometimes seems to take a long time, verify whether this feature is enabled.

- Set the CPU power usage. This feature is available only on supported systems in which the CPU supports a slow down, power saving mode.
- View and set these features on single workstations or groups of systems through the System Administration utilities or through command line interfaces. The operating system provides utilities for managing and monitoring hardware across a network of systems.
- Use the Event Management (EVM) interface to monitor power management events.

There are several methods to invoke and manage power conservation by using the following utilities:

- Manage an individual workstation by using the X11–compliant graphical user interface `/usr/bin/X11/dxpower` utility. See the online help and `dxpower`(8) for information on invoking this interface.
- Use `sysconfig` and `sysconfigdb` to load and set kernel attributes. See `sysconfig`(8) and `sysconfigdb`(8) for a list of command options. This method of power management will be retired in a future release.

3.5.1 Using the dxpower Utility's Graphical User Interface

The graphical user interface `dxpower` can be used on the graphics console of a host system or invoked from the command line. Certain features are password-protected, and can be used only by the system administrator on a root login. A nonprivileged user can control features such as the energy-saving features of a monitor. If you are using CDE, you can open the `dxpower` power management utility by performing the following steps:

1. Click on the Application Manager icon.
2. Double click on the System_Admin application group icon.
3. Double click on the DailyAdmin application group icon.
4. Double click on the Power Management icon.

If you are using a terminal or other X11 windowing environment, you can start the `dxpower` utility from the command line as follows:

```
# /usr/bin/X11/dxpower
```

When the `dxpower` utility runs, a power management window is displayed on your screen. The window provides check boxes that you use to select modes of operation, and sliding scales (bars) that you use to specify idle time limits. Idle time is the amount of time elapsed before the device goes into power saving mode and can be set from 1 to 60 minutes. Depending on your login privileges, the graphical interface allows you to:

- Enable or disable power management for all supported devices on the host system.
- Specify the time of day when power management is enabled. For example, you can set systems to only go into power saving modes during the night.
- Enable the energy-saving features of the graphics monitor, and set the minimum idle time before standby, suspend, and power-off modes are selected. For example, if a system is rarely used, you can set it to go straight to power-off mode after only a few minutes of idle time.
- Enable power saving mode for each individual disk. For example, you may want to keep the boot disk in full power mode, but spin down any unused user file systems after a specified idle time to conserve power.

Caution

Monitors (displays) that do not support DPMS (Display Power Management Signaling) may be damaged by the activation of the DPMS feature. It is important that you verify the specifications for your monitor in the owner's manual. Monitors that support DPMS and are put in a power savings state vary in the time it takes to come out of power savings. The longer the monitor is in power-off state, the longer it takes for the display to return as a result of mouse or keyboard activity. This is the result of the monitor phosphor cooling down and the time required to heat it back up, and not a function of the power management software.

For more information about how to use the `dxpower` utility, start the application and then select `Help` in the lower right-hand corner of the window.

3.5.2 Using the sysconfig Command

You can control power management attributes from the command line by using the `sysconfig` command to manage the `sysconfigdb` database. For example, you need to use the `sysconfig` command if you activate power management for a system from a remote terminal or from a local console terminal.

If you activate the power management tools from a console terminal where CDE is not running, only the `graphics_powerdown` and `graphics_off_dwell` attributes apply. Changing the `graphics_standby_dwell` and `graphics_suspend_dwell` attribute values has no effect. See Section 3.5.2.1 for descriptions of these attributes.

Caution

Do not attempt to use the `sysconfig` and `dxpower` commands simultaneously. If you do, you could encounter unpredictable behavior.

3.5.2.1 Changing Power Management Values

To change the power management values that take effect every time you restart the kernel, you create a stanza. See `stanza`(4) for more information. The stanza file can contain the following power management attributes:

- `default_pwrmgr_state`

  The global power management state. Specify 1 to enable or 0 to disable this attribute.

- `cpu_slowdown`

  The current state of CPU slowdown. Specify 1 to enable or 0 to disable this attribute.

- `disk_dwell_time`

  The default dwell time, in minutes, for registered disks.

- `disk_spindown`

  The current state of disk spindown. Specify 1 to enable or 0 to disable this attribute.

- `graphics_powerdown`

  The current state of graphics power down. Specify 1 to enable or 0 to disable this attribute.

- `graphics_standby_dwell`

  The default dwell time, in minutes, for `standby` Display Power Management Signaling (DPMS) mode. Specify a value of 0 to disable this attribute.

  When Monitor Power Management and Screen Saver are enabled simultaneously, DPMS-capable monitors are placed in active power-off mode. In addition, on Energy-Star compliant platforms, the disk spin down feature may be activated also. While these energy-saving features are active, the X server may override them so that it can continue to run the screen saver. To minimize power consumption, stop using active screen savers by doing any of the following:

  – In the Screen Saver panel of the Screen dialog box, under the Style Manager, select Blank Screen and deselect any active screen savers that may be running.

- Select Off in the same dialog box.
- Execute the xset s off command from a terminal client window.

- graphics_suspend_dwell

  The default dwell time, in minutes, for suspend DPMS mode. Specify 0 to disable this attribute or specify a value greater than or equal to the value for graphics_standby_dwell.

- graphics_off_dwell

  The default dwell time, in minutes, for off DPMS mode. Specify 0 to disable this attribute or specify a value greater than or equal to the values for graphics_standby_dwell and graphics_suspend_dwell.

For example, you can create a stanza file called power_mgr.stanza that defines the following values for the attributes:

```
pwrmgr:
  default_pwrmgr_state=1
  cpu_slowdown=1
  disk_dwell_time=20
  disk_spindown=1
  graphics_powerdown=1
  graphics_standby_dwell=5
  graphics_suspend_dwell=10
  graphics_off_dwell=15
```

For the disk_dwell_time, graphics_standby_dwell, graphics_suspend_dwell, and graphics_off_dwell attributes, the specified values indicate the number of minutes to wait before powering down the idle hardware. In this case, the power management subsystem waits 20 minutes before disk spindown, and 5, 10, and 15 minutes before DPMS standby, suspend, and off modes, respectively. The remaining attributes, have a value of 1, which indicates that the function is enabled.

After you create and save the stanza file, enter the following command to update the /etc/sysconfigtab database:

```
# sysconfigdb -a -f power_mgr.stanza pwrmgr
```

See sysconfigdb(8) for more information on using stanza files.

3.5.2.2 Changing a Running Kernel or X Server

To change the values of attributes in the running kernel, use the sysconfig –r command. For example:

```
# sysconfig -r pwrmgr cpu_slowdown=0
```

You can change more than one attribute at a time, as shown in the following example:

```
# sysconfig -r pwrmgr \
graphics_powerdown=1 graphics_standby_dwell=10
```

See sysconfig(8) for more information on modifying system attributes.

See the dpms switches described in Xdec(1X) and xset(1X) for information about changing DPMS modes and values in the X Server.

3.5.3 Using the SysMan Station

If you are using the SysMan Station, you can select system entities such as CPUs or disk devices from the system topology map.

Clicking MB3 on an icon enables a list of management actions for the selected device, one of which may be the power management application dxpower. Selecting this menu item runs dxpower on the device.

3.6 Adding Swap Space

The operating system uses a combination of physical memory and swap space on disk to create virtual memory, which can be much larger than the physical memory. Virtual memory can support more processes than the physical memory alone. This section and the sections that follow describe important virtual memory concepts that you should consider when configuring swap space.

Note

You may see messages implying that there is a shortage of virtual memory (vm) or processes may be killed because of an apparent lack of vm. In such cases, virtual memory does not always mean swap space, but can refer to resources required by the vm kernel subsystem.

If you do not observe any excessive use of swap space, and if you do not observe messages that specifically reference a lack of swap space, the problem may be a lack of per-process memory limits. See Section 3.6.5 for more information.

The virtual memory (vm) kernel subsystem controls the allocation of memory to processes by using a portion of physical memory, disk swap space, and various daemons and algorithms. A page is the smallest portion of physical memory that the system can allocate (8 KB of memory).

Virtual memory attempts to keep a process' most recently referenced virtual pages in physical memory. When a process references virtual pages, they are brought into physical memory from their storage locations on disk. Modified virtual pages can be moved to a temporary location on the disk (called swap space) if the physical pages (the pages in physical memory) that contain the virtual pages are needed by either a newly referenced virtual page or by a

page with a higher priority. Therefore, a process' virtual address space can consist of pages that are located in physical memory, stored temporarily in swap space, and stored permanently on disk in executable or data files. Virtual memory operation involves:

Paging	Reclaiming pages so they can be reused
Swapping	Writing a suspended process' modified (dirty) pages to swap space, which frees large amounts of memory

Paging involves moving a single virtual page or a small cluster of pages between disk and physical memory. If a process references a virtual page that is not in physical memory, the operating system reads a copy of the virtual page from its permanent location on disk or from swap space into physical memory. This operation is called a pagein. Pageins typically occur when a process executes a new image and references locations in the executable image that were not referenced previously.

If a physical page is needed to hold a newly referenced virtual page or a page with a higher priority, the operating system writes a modified virtual page (or a small cluster of pages) that was not recently referenced to the swap space. This operation is called modified page writing or a pageout. Only modified virtual pages are written to swap space because there is always a copy of the unmodified pages in their permanent locations on disk.

Swapping involves moving a large number of virtual pages between physical memory and disk. The operating system requires a certain amount of physical memory for efficient operation. If the number of free physical pages drops below the system-defined limit, and if the system is unable to reclaim enough physical memory by paging out individual virtual pages or clusters of pages, the operating system selects a low priority process and reclaims all the physical pages that it is using. It does this by writing all its modified virtual pages to swap space. This operation is called a swapout. Swapouts typically occur on systems that are memory constrained.

Caution

The ability of the system to save crash dumps after a system crash is also affected by the size and availability of swap space. If the swap space allocation is insufficient, the system is unable to save a crash dump, which can contain valuable information to assist you in recovering from errors. See Chapter 12 for information on crash dump space requirements.

3.6.1 Related Documentation and Utilities

The following documentation resources and utilities provide information on administering swap space.

3.6.1.1 Related Documentation

Information on administering swap space can be found in the following manuals:

Installation Guide — Advanced Topics

Describes how to plan for initial swap space, and set up initial swap during installation of the operating system.

System Configuration and Tuning

Describes advanced concepts of virtual memory and swap, including strategies for performance tuning that involve swap space configuration.

See `swapon`(8) and `swapon`(2) for information on creating additional swap space. See Section 3.6.1.2 for the reference pages for the related utilities.

3.6.1.2 Related Utilities

The following utilities are used during swap space administration:

Utility	Path Name	Description
Disk Configuration	`/usr/sbin/diskconfig`	This graphical user interface can be used to examine disks to locate unused partitions that can be assigned to swap. See `diskconfig`(8) for information on invoking and using the utility.
Kernel Tuner	`/usr/bin/X11/dxkerneltuner`	This graphical user interface can be used to modify kernel swap attributes in the system configuration file. See `dxkerneltuner`(8) for information on invoking and using this utility.

Utility	Path Name	Description
sysconfig	/sbin/sysconfig	This command line interface can be used to modify kernel swap attributes in the system configuration file. See sysconfig(8) for information on modifying system attributes.
disklabel	/sbin/disklabel	This command line interface can be used to modify kernel swap attributes in the system configuration file. See disklabel(8) for information on labeling a disk (otherwise known as formatting disk partitions).

3.6.2 Allocating Swap Space

Initially, swap space is planned and allocated during system installation, based on your requirements for the installed system. However, you may want to add swap space to improve system performance or if you added more physical memory to your system. A cue to increase swap space is provided by system console warning messages, stating that available swap space is depleted. Before adding swap space, verify that any sudden lack of space is not caused by a system problem. Use the following command to ensure that runaway processes or unusual user activities are not using up swap space:

```
# ps agx
```

(Alternatively, you can examine system log and event files for swap error messages.) If the resulting list of processes looks normal, you may need to add swap space.

Swap space can be added temporarily by running the swapon command. To make the additional swap permanent, you must add an entry to the vm section of the /etc/sysconfigtab file. The process is as follows:

1. The swapon command verifies a disk partition to ensure that you do not write over data or use overlapping partitions. If you have a choice of disks, you may want to choose a location for swap on a convenient fast disk that does not have excessive I/O usage. For example, the disk where your user files are located probably has higher I/O demands.

   Use the diskconfig utility to examine disks and choose a suitable partition.

2. Run swapon to create the swap partition, as shown in the following example:

   ```
   # /sbin/swapon /dev/disk/dsk0b
   ```

   You may require some temporary swap space, such as additional space to take a full crash dump instead of a partial dump. If this is the case, you do not need to take any further action and the swap partition is ready for use. To review the current swap configuration, use the following command:

   ```
   # /sbin/swapon -s
   ```

   You can repeat step 1 to add additional partitions, if required.

3. To make the additional swap space permanent, you must edit the vm section of the /etc/sysconfigtab file to include the new partition as follows:

   - Copy the current file to a temporary file name in case you need to recover it. Use a text editor to open the file, and search for the string vm:
   - You observe a swapdevice= entry for the initial swap space, created during installation. Add the device special file name for the new swap partition, separating each swap device entry with a comma, as follows:

     ```
     vm:
                  swapdevice=/dev/disk/dsk1b, /dev/disk/dsk3h
                  vm-swap-eager=1
     ```

     The new swap partitions open automatically when the system is rebooted, or when you use the command:

     ```
     # /sbin/swapon -a
     ```

See swapon(8) for information about how the command interacts with overlapping partitions.

The amount of swap space that your system requires depends on the swap space allocation strategy that you use and your system workload. Strategies are described in the following section.

3.6.3 Estimating Swap Space Requirements

There are two strategies for swap space allocation: immediate mode and deferred or over-commitment mode. The two strategies differ in the point in time at which swap space is allocated. In immediate mode, swap space is recovered when modifiable virtual address space is created. In deferred mode, swap space is not reserved or allocated until the system needs to write a modified virtual page to swap space.

Note

The operating system terminates a process if it attempts to write a modified virtual page to swap space that is depleted.

Immediate mode is more conservative than deferred mode because each modifiable virtual page reserves a page of swap space when it is created. If you use the immediate mode of swap space allocation, you must allocate a swap space that is at least as large as the total amount of modifiable virtual address space to be created on your system. Immediate mode requires significantly more swap space than deferred mode because it guarantees that there is enough swap space if every modifiable virtual page is modified.

If you use the deferred mode of swap space allocation, you must estimate the total amount of virtual address space to be both created and modified, and compare that total amount with the size of your system's physical memory. If this total amount is greater than half the size of physical memory, the swap space must be large enough to hold the modified virtual pages that do not fit into your physical memory. If your system's workload is complex and you are unable to estimate the appropriate amount of swap space by using this method, use the default amount of swap space and adjust the swap space as needed, first. Consider using a swap size of about half the size of physical memory.

Always monitor your system's use of swap space. If the system issues messages that indicate that swap space is almost depleted, you can use the `swapon` command to allocate additional swap space. If you use the immediate mode, swap space depletion prevents you from creating additional modifiable virtual address space. If you use the deferred mode, swap space depletion can result in one or more processes being involuntarily terminated.

For more information on virtual memory, see the *System Configuration and Tuning* manual.

3.6.4 Selecting the Swap Space Allocation Method

To determine which swap space allocation method is being used, you can examine the `vm:` section of the `/etc/sysconfigtab` file. Alternatively, use the `dxkerneltuner` or `sysconfig` utility to examine kernel attribute values. You observe an entry similar to the following:

```
vm:
                swapdevice=/dev/disk/dsk1b, /dev/disk/dsk3h
                vm-swap-eager=1
```

The entry for `vm-swap-eager=` determines the allocation method as follows:

`vm-swap-eager=1` The system is using immediate swap mode.

`vm-swap-eager=0` The system is using deferred swap mode.

Either edit the `/etc/sysconfigtab` file to change the current value, or alternatively, use the `dxkerneltuner` or `sysconfig` utility to modify the attribute dynamically.

You must reboot the system for the new swap method to take effect. You may receive the following boot time informational messages when you switch to deferred mode or when you boot a system that is using the deferred method:

```
vm_swap_init:  warning sbin/swapdefault swap device not found
vm_swap_init:  in swap over-commitment mode
```

3.6.5 Correcting an Apparent Lack of Swap Space

There are limits on the amount of virtual memory that an individual process can use. These limits are not related to the total amount of available swap space. Consequently, you may see error messages stating that a process has run out of virtual memory even though a swap monitor, such as the `dxsysinfo` utility, does not display a swap shortage. In some cases, a process could be killed automatically when it exceeds its allotted virtual memory.

See the discussion for the `vm_swap_eager` attribute in `sys_attrs_vm`(5) for a description of the effect of the swap allocation mode.

Use the following command to verify that your swap space is adequate:

```
# swapon -s
```

The data field titled `In-use space:` informs you if you are using most of your available swap space.

If you are not using all your available swap space but you are observing problems running large processes, you may need to assign more resources to the process. There are several kernel attributes in the `proc` subsystem that you can use to control the per-process virtual memory resources:

Stack limit	The `per-proc-stack-size` and `max-per-proc-stack-size` attributes.
Data limit	The `per-proc-data-size` and `max-per-proc-data-size` attributes.
Address space	The `max-per-proc-address-space` and `per-proc-address-space` attributes.

Consider the following options if you encounter problems that appear to be from a lack of memory:

- See the reference page that describes your command shell, such as `ksh`(1). Command shells have a `limit` or `ulimit` option that enables you to modify the virtual memory resources for a process.
- Use the `sysconfigdb` command or the Kernel Tuner GUI to modify the value of the per-process resource limits in the `/etc/sysconfigtab` file. Instructions for modifying kernel attributes are provided in Chapter 4.
- See the *System Configuration and Tuning* manual for information on tuning virtual memory (`vm`) subsystem attributes, such as `vm-maxvas`. See `sys_attrs`(5) for more information on the system attributes.

4

Configuring the Kernel

This chapter discusses kernel configuration during and after installation, and dynamic and static configuration. The following topics are discussed in this chapter:

- An overview of kernel configuration (Section 4.1)
- Pointers to other relevant documentation, in particular the individual reference pages that document all the attributes for every available kernel subsystem (Section 4.2)
- A description of kernel configuration at installation (Section 4.3)
- Information on determining when to configure your kernel (Section 4.4)
- A discussion on dynamic system configuration, including the Kernel Tuner graphical user interface, `/usr/bin/X11/dxkerneltuner` (Section 4.5)
- A discussion on static system configuration (Section 4.6)
- A description of the configuration files (Section 4.7)

4.1 Overview

The operating system kernel is a memory-resident executable image that handles all the system services – hardware interrupts, memory management, interprocess communication, process scheduling – and makes all other work on the operating system possible. In addition to the code that supports these core services, the kernel contains a number of subsystems.

A subsystem is a kernel module that extends the kernel beyond the core kernel services. File systems, network protocol families, and physical and pseudodevice drivers are all examples of supported subsystems. Some subsystems are required in the kernel, and others are optional. You configure your kernel by adding and removing these optional subsystems, either during installation or later when you need to change the system.

Another way to configure your kernel is by tuning certain values stored in it. For example, the kernel contains values that you can adjust for faster disk access. Modifying values to optimize disk access can improve your system's performance, however it can affect performance in other areas. Detailed information on system tuning and the interaction of attributes is included in the *System Configuration and Tuning* manual.

The system provides two methods of configuring your kernel: the dynamic method and the static method.

Dynamic method	You use commands to configure the kernel while it is running.
Static method	You modify system files and rebuild the kernel.

Modifying system files and rebuilding the kernel is often a difficult process, so use dynamic kernel configuration whenever possible. You cannot make all modifications dynamically, and dynamic changes may not be preserved across reboots.

4.2 Related Documentation and Utilities

The following sources provide information on system attributes, configuration tools, and utilities, as well as detailed reference information on configuration options. These sources, which are discussed in the following sections, are in the form of manuals, reference pages, and online help.

4.2.1 Manuals

The following manuals in the Tru64 UNIX operating system documentation set discuss system configuration.

- The *Installation Guide* and *Installation Guide — Advanced Topics* manuals provide information about initial kernel configuration during installation.
- The *Network Administration: Connections* and *Network Administration: Services* manuals provide information on configuring the network.
- The *System Configuration and Tuning* manual provides detailed information on system configuration and tuning.

4.2.2 Reference Pages

The reference pages listed here provide information on system attributes and utilities.

`sys_attrs`(5)	Contains information about system attributes and provides a pointer to several `sys_attrs*` reference pages that cover individual kernel subsystems such as `streams` or `socket`. Several subsystems have no configurable attributes and are not listed here.

Note

See the appropriate reference page and the *System Configuration and Tuning* manual before changing the value of any attribute.

`sys_attrs_vm`(5) Describes attributes for subsystems that are mandatory when the kernel is built. These subsystems include: Configuration Manager (`cm`), Generic Kernel (`generic`), Interprocess Communication (`ipc`), Process (`proc`), Virtual File System (`vfs`), and Virtual Memory (`vm`).

`sys_attrs_advfs`(5) Describes the attributes for the Advanced File System (`advfs`) kernel subsystem.

`sys_attrs_atm`(5) Describes attributes for Asynchronous Transfer Mode (ATM) kernel subsystems: Base ATM support (`atm`), ATM Forum Integrated Layer Management Interface (`atmilmi3x`), Classical IP services (`atmip`), ATM Forum signaling and Integrated Layer Management Interface support (`atmuni`), ATM Forum LAN Emulation (`lane`), and ATM Forum signaling (`uni3x`).

`sys_attrs_*`(5) Describe the attributes for single subsystems. The following table lists these reference pages, but reference pages for kernel subsystems that have not tunable attributes are omitted.

Reference Page	Description
`sys_attrs_ace`(5)	Serial Device (ace) kernel subsystem
`sys_attrs_alt`(5)	alt kernel subsystem, used by the Gigabit Ethernet adapters
`sys_attrs_autofs`(5)	AutoFS kernel subsystem
`sys_attrs_bparm`(5)	Boot Parameters kernel subsystem

Reference Page	Description
sys_attrs_bsd_tty(5)	BSD Terminal kernel subsystem
sys_attrs_cam(5)	SCSI CAM I/O kernel subsystem
sys_attrs_cm(5)	Configuration Manager kernel subsystem
sys_attrs_dli(5)	Data Link Interface kernel subsystem
sys_attrs_dlpi(5)	Data Link Provider Interface kernel subsystem
sys_attrs_ee(5)	ee kernel subsystem, used by the 10/100 MB/s Ethernet adapters
sys_attrs_eisa(5)	EISA Bus kernel subsystem
sys_attrs_fta(5)	fta kernel subsystem, used by the Fiber Distributed Data Interface (FDDI) adapters
sys_attrs_generic(5)	Generic kernel subsystem
sys_attrs_gpc_input(5)	GPC Input kernel subsystem
sys_attrs_io(5)	Input/Output kernel subsystem
sys_attrs_ipc(5)	Interprocess Communication kernel subsystem
sys_attrs_iptunnel(5)	Internet Protocol Tunneling kernel subsystem
sys_attrs_ipv6(5)	Internet Protocol Version 6 kernel subsystem
sys_attrs_isa(5)	ISA Bus kernel subsystem
sys_attrs_kevm(5)	Kernel Event Manager kernel subsystem
sys_attrs_lag(5)	Link Aggregation Group kernel subsystem
sys_attrs_lfa(5)	lfa kernel subsystem
sys_attrs_lsm(5)	Logical Storage Manager kernel subsystem
sys_attrs_net(5)	Network kernel subsystem
sys_attrs_netrain(5)	Redundant Array of Independent Network adapters (NetRAIN) kernel subsystem
sys_attrs_pci(5)	PCI kernel subsystem
sys_attrs_ppp(5)	Point-to-Point Protocol kernel subsystem
sys_attrs_presto(5)	PrestoServe kernel subsystem
sys_attrs_proc(5)	Process kernel subsystem
sys_attrs_psm(5)	Process Set Manager kernel subsystem
sys_attrs_pts(5)	Pseudoterminal kernel subsystem
sys_attrs_pwrmgr(5)	Power Manager (pwrmgr) kernel subsystem
sys_attrs_rt(5)	Realtime kernel subsystem

Reference Page	Description
sys_attrs_scc(5)	Serial driver kernel subsystem
sys_attrs_scc_input(5)	Serial driver keyboard driver kernel subsystem
sys_attrs_sec(5)	Security kernel subsystem
sys_attrs_snmp_info(5)	Simple Network Management Protocol kernel subsystem
sys_attrs_socket(5)	Socket kernel subsystem
sys_attrs_streams(5)	STREAMS kernel subsystem
sys_attrs_tc(5)	TURBOchannel Bus kernel subsystem
sys_attrs_tu(5)	tu kernel subsystem, used by the 10/100 Mb/s Ethernet adapters
sys_attrs_ufs(5)	UNIX File System kernel subsystem
sys_attrs_uipc(5)	AF_UNIX interprocess communication kernel subsystem
sys_attrs_vfs(5)	Virtual File System kernel subsystem

doconfig(8)	Describes the utility that you use to build the kernel with the settings specified in the current system configuration files.
kopt(8)	Describes a utility that enables you to select kernel options.
sysconfig(8), sysconfigtab(4), and sysconfigdb(8)	Describe the command line utility and database used to maintain the kernel subsystem configuration and to modify or display kernel subsystem attributes. The sysconfigtab command documents the file format of the configuration database; use the sysconfigdb utility to manage this configuration database.
sysconfigdb(8) and stanza(4)	Describe the command line utility used to manage the subsystem configuration database. The stanza command documents the format of a configuration stanza file. This is a file fragment that is built into the configuration database when you run sysconfigdb.

`autosysconfig`(8)	Describes a utility used to maintain the list of dynamic kernel subsystems that are configured automatically.
`cfgmgr`(8)	Describes a server that the `sysconfig` and other utilities use to manage kernel subsystems. See the `kloadsrv`(8), which documents the kernel load server.
`dxkerneltuner`(8)	Describes the Kernel Tuner graphical user interface, which enables you to modify or display kernel subsystem attributes.
`sys_check`(8)	Describes the `sys_check` utility, which examines various system attributes and makes recommendations for their appropriate settings. See Chapter 3 for more information.

4.2.3 Online Help

The Kernel Tuner graphical user interface offers its own online help.

4.3 System Configuration at Installation Time

When you install the operating system, the installation program initially copies a kernel image to the root partition of your system disk. This kernel image, known as the generic kernel, supports all processors and hardware options that are available for use with the current version of the operating system. In this way, the installation program ensures that you can boot your system regardless of its configuration. The file for the generic kernel is `/genvmunix`.

Toward the end of the installation, after all the subsets you selected have been written to disk and verified, the installation program calls the `/usr/sbin/doconfig` program. When it runs, the `/usr/sbin/doconfig` program calls another program, known as the `sizer` program. The `sizer` program determines what hardware and software options are installed on your system and builds a target configuration file specific to your system. (The configuration file is the system file that controls what hardware and software support is linked into the kernel.) The `/usr/sbin/doconfig` program then builds your custom `/vmunix` kernel from this target configuration file. This kernel is built using the default values for all subsystem attributes.

Unlike the generic kernel copied to the system at installation time, the target kernel is tailored to your system. Only the hardware and software options available on your system are compiled into the target kernel. As a result, the target kernel is much smaller and more efficient than the generic kernel.

When the installation is complete, the target kernel resides either in the root partition of your system disk or in memory, depending upon how your system was built. (See Section 4.6 for information about the different ways in which you can build a kernel.) If the appropriate console boot variables are set, your system always boots the target kernel automatically. For information about setting and using console boot variables, see Chapter 2 and the Owner's Manual for your system.

4.4 Deciding When and How to Reconfigure Your Kernel

After your target kernel is built and started by the installation procedure, you can use it without modifications, unless one of the following occurs:

- You decide to add new subsystems to the kernel, for example by installing new devices or to use additional options such as Asynchronous Transfer Mode (ATM).
- You decide to remove subsystems from the kernel, for example by removing a device or a feature such as the Logical Storage Manager (LSM).
- You decide to change the default attribute values in the kernel because system performance is not acceptable (perhaps because you are running an intensive application). Examples of such intensive applications may be internet web servers or databases. System tuning requires that you fully understand the affect of changing kernel attributes so that you do not create an unusable kernel or degrade system performance.

  For example, you may decide to run the `sys_check` utility as part of your normal system monitoring operations. Based on its analysis of system use, the report generated by `sys_check` may suggest new values for kernel attributes or the loading of additional subsystems. However, you should see the *System Configuration and Tuning* manual for information on potential affects on other aspects of system performance before you modify an attribute's value.

Most devices are recognized automatically by the system and configured into the kernel at boot time; see *Hardware Management* for information on adding devices. However, some devices, such as third-party disk drives, older types of drives, or products such as scanners and PCMCIA cards must be added manually. For these devices, you must reconfigure your kernel, either dynamically or statically, when one of these situations occurs. The method you use to reconfigure your kernel depends upon the support provided by the subsystem or subsystem attributes.

Some kernel subsystems, such as the envmon environmental monitoring subsystem, are dynamically loadable, meaning that you can add the subsystem to or remove the subsystem from the kernel without rebuilding the kernel. Often, subsystems that are dynamically loadable also allow you to dynamically configure the value of their attributes. Therefore, you can tune the performance of these subsystems without rebuilding the kernel. To determine whether an attribute is dynamically configurable, use the -m with the sysconfig and search for the dynamic identifier as follows:

```
# sysconfig -m | grep dynamic
lat: dynamic
envmon: dynamic
hwautoconfig: dynamic
```

If you decide to add or remove these subsystems from the kernel or configure the value of their attributes, use the procedures described in Section 4.5.

Some subsystems, such as required subsystems, are not dynamically loadable. However, these subsystems may allow you to configure the value of attributes dynamically. If so, you can configure the value of these subsystem attributes without rebuilding the kernel.

You can dynamically configure attributes using the following methods:

- You can configure the value of attributes in the running kernel using the sysconfig -r command. Only a few kernel subsystems support this run-time configuration.
- You can use the Kernel Tuner (dxkerneltuner) graphical user interface, which performs most of the same display and set functions as sysconfig. Launch this utility from the command line as follows:

  ```
  # /usr/bin/X11/dxkerneltuner
  ```

  Alternatively, open the Application Manager from the CDE front panel and select the Monitoring/Tuning folder. When the folder is opened, select the Kernel Tuner icon. See the dxkerneltuner(8) and the application's online help for more information on using the Kernel Tuner.

  The Kernel Tuner GUI displays all the available kernel subsystems in the main window. Select a subsystem to display the subsystem attributes, their current values, and the maximum and minimum values. If any attribute is modifiable, it is displayed with a text entry field where you enter a revised value for the attribute.
- You can configure the value of attributes in the dynamic subsystem database, /etc/sysconfigtab. When you want to run a kernel that contains the new attribute values, you reboot your system specifying the new kernel.

If you decide to configure attributes of these subsystems, use the procedures described in Section 4.5.8. It is recommended that you do not manually

edit system files such as /etc/sysconfigtab. Instead, use a command or utility such as dxkerneltuner to make any changes.

If you purchase a device driver or other kernel subsystem from a third-party company, that product may be dynamically loadable or allow you to dynamically configure attribute values. For information about dynamically configuring your kernel when working with products from other vendors, see the documentation for the product and Section 4.5.

If the subsystem you want to add, remove, or configure does not support dynamic configuration, you must use the static configuration method. Also, you must use this method to configure system parameters that do not support dynamic configuration. For information about the static configuration method, see Section 4.6.

4.5 Dynamic System Configuration

When you need to load, unload, or modify a dynamic subsystem, you use the /sbin/sysconfig command. This command has the following syntax:

/sbin/sysconfig [-h *hostname*] [-i *index*
[-v | -c | -d | -m | -o | -q | -Q | -r | -s | -u]] [*subsystem-name*]
[*attribute-list* | *opcode*]

You must be the superuser to load and unload subsystems, and you must know the name of the subsystem you want to manage. Determine the name of a subsystem by looking in the documentation that accompanies the subsystem or in the directories into which the subsystem is installed. Subsystems are installed in either the /subsys directory or the /var/subsys directory. When a subsystem is installed, a file named *subsystem-name*.mod appears in one of those two directories. You use that subsystem name as input to the /sbin/sysconfig command. The following sections describe the commands that you use to manage subsystems.

You can load and unload subsystems on a local system or a remote system. For information about adding and removing subsystems on remote systems, see Section 4.5.7.

If you are writing a loadable device driver or other loadable subsystem, see the device driver documentation and the *Programmer's Guide*. The device driver documentation describes the tasks performed by the system when you install a loadable device driver. These manuals also describe how to write and package loadable device drivers. The *Programmer's Guide* provides general information about creating subsystems that are dynamically configurable and discusses the framework that supports dynamic configuration of subsystems and attributes.

4.5.1 Configuring Subsystems

To configure (load) a subsystem, enter the `sysconfig` command using the `-c` option. Use this command whether you are configuring a newly installed subsystem or one that was removed using the `/sbin/sysconfig -u -u` (unconfigure) command option. For example, to configure the environmental monitoring `envmon` subsystem, enter the following command:

```
# /sbin/sysconfig -c envmon
```

4.5.2 Listing the Configured Subsystems

Subsystems can be known to the kernel, but not available for use. To determine which subsystems are available for use, use the `/sbin/sysconfig –s` command. This command displays the state of all subsystems. Subsystems can have the following states:

- Loaded and configured (available for use)
- Loaded and unconfigured (not available for use but still loaded)

  This state applies only to static subsystems, which you can unconfigure, but you cannot unload.
- Unloaded (not available for use)

  This state applies only to loadable subsystems, which are automatically unloaded when you unconfigure them.

If you use the `/etc/sysconfig -s` command without specifying a subsystem name, a list of all the configured subsystems is displayed. For example:

```
# /sbin/sysconfig -s
cm: loaded and configured
hs: loaded and configured
ksm: loaded and configured
generic: loaded and configured
io: loaded and configured
ipc: loaded and configured
proc: loaded and configured
sec: loaded and configured
socket: loaded and configured
rt: loaded and configured
advfs: loaded and configured
.
.
.
envmon: unloaded
```

This list (which is truncated) includes both statically linked subsystems and dynamically loaded subsystems.

To get information about the state of a single subsystem, include the name of the subsystem on the command line:

```
# /sbin/sysconfig -s lsm
lsm: unloaded
```

4.5.3 Determining the Subsystem Type

You can determine whether a subsystem is dynamically loadable or static by using the `/sbin/sysconfig -m` command, as shown:

```
# /sbin/sysconfig -m kinfo lat
kinfo: static
lat: dynamic
```

The output from this command indicates that the `kinfo` subsystem is static, meaning that you must rebuild the kernel to add or remove that subsystem from the kernel. The `lat` subsystem is dynamic, meaning that you can use the `sysconfig -c` command to configure the subsystem and the `sysconfig -u` command to unconfigure it.

4.5.4 Unloading a Subsystem

To unconfigure (and possibly unload) a subsystem, use the `/sbin/sysconfig -u` command, as shown:

```
# /sbin/sysconfig -u hwautoconfig
```

If you frequently configure and unconfigure device drivers you may notice that the device special files associated with a particular device driver differ from time to time. This behavior is normal. When you configure a device driver using the `/sbin/sysconfig` command, the system creates device special files. If you unload that device driver and load another one that uses the same `cdev` or `bdev` major numbers, the system removes the device special files for the unloaded device driver. Therefore, it must create new device special files the next time you configure the device. See the *Hardware Management* manual and `dsfmgr`(8) for more information on device special files.

4.5.5 Maintaining the List of Automatically Configured Subsystems

The system determines which subsystems to configure into the kernel at system reboot time by verifying the list of automatically configured subsystems. The system configures each subsystem on the list, using the `sysconfig -c` command at each system reboot.

You maintain the list of automatically configured subsystems by using the `/sbin/init.d/autosysconfig` command.

This command has the following syntax:

/sbin/init.d/autosysconfig [list] [add *subsystem-name*] [delete *subsystem-name*]

Use the `/sbin/init.d/autosysconfig list` command to see a list of the loadable subsystems that the system automatically configures at each reboot.

To add a subsystem to the list, use the `/sbin/init.d/autosysconfig add` command. For example to add the `lat` subsystem, enter the following command:

```
# /sbin/init.d/autosysconfig add lat
```

If you unload a subsystem that is on the automatically configured subsystem list, you should remove that subsystem from the list. Otherwise, the system configures the subsystem back into the kernel at the next system reboot. To remove the subsystem from the automatically configured subsystems list, use the `/sbin/init.d/autosysconfig delete` command. For example, to delete the `lat` subsystem, enter the following command:

```
# /sbin/init.d/autosysconfig delete lat
```

4.5.6 Managing Subsystem Attributes

To improve the performance or behavior of a subsystem, or of the system as a whole, you may modify the values of subsystem attributes. You can make such modifications using `sysconfig`, `sysconfigdb`, or the Kernel Tuner GUI. Under certain circumstances, such as recovering a crashed system, you may need to use the debugger `dbx` to examine and change the attributes in a damaged kernel. See the *Kernel Debugging* manual for information on this procedure.

If you modify an attribute at run time, the modification persists only during the current run session. If you shut down and reboot the system, the modification is lost. To modify subsystem attributes so that changes persist across reboots, you must store the attribute's modified value in the `/etc/sysconfigtab` database, as described in Section 4.5.8. The persistence of a modified attribute value depends on what command or utility option you use, according to the following guidelines:

- For permanent modifications that persist across reboots, use `sysconfigdb` (or `dbx`) at the command line. Alternatively, use the Kernel Tuner GUI, specifying and saving the change using the Boot Time Value field.
- For temporary modifications that do not persist across reboots, use `sysconfig -r` at the command line. Alternatively, use the Kernel Tuner graphical user interface, specifying a change to the current value of an attribute.

Note

In previous releases of the operating system, the /etc/sysconfigtab file was documented as a system file that you could modify with a text editor, such as vi. In recent releases, maintenance of the subsystem stanzas has become important for update installations and for the kernel to recognize changes. To maintain the correct structure of /etc/sysconfigtab, use only the sysconfigdb command or the Kernel Tuner graphical user interface to make changes.

See sysconfig(8), sysconfigdb(8), sysconfigtab(4), and dxkerneltuner(8) for information.

The system parameters that are set in the system configuration file define the system tables, and should be viewed as establishing default values in the kernel. You can override these values by using the /sbin/sysconfig command or by storing a value in the /etc/sysconfigtab database. For more information about the configuration file (and the param.c file), see Section 4.6.

You can manage dynamic subsystem attributes either locally or remotely. For information on how to use the /sbin/sysconfig command remotely, see Section 4.5.7.

4.5.6.1 Determining the Current Value of Subsystem Attributes

Use the /sbin/sysconfig -q command or dxkerneltuner to determine the value assigned to subsystem attributes. When you enter the /sbin/sysconfig -q command, the subsystem you specify on the command line must be loaded and configured. For information about getting a list of the loaded and configured subsystems, see Section 4.5.2.

The following example shows how to use this command to determine which attributes belong to the generic subsystem:

```
# /sbin/sysconfig -q generic
generic:
booted_kernel = vmunix
booted_args = vmunix
lockmode = 0
lockdebug = 0
locktimeout = 15
max_lock_per_thread = 16
lockmaxcycles = 0
rt_preempt_opt = 0
cpu_enable_mask = 0x1
binlog_buffer_size = 0
```

```
msgbuf_size = 32768
dump_sp_threshold = 4096
use_faulty_fpe_traps = 0
partial_dump = 1
make_partial_dumps = 1
compressed_dump = 1
make_compressed_dumps = 1
expected_dump_compression = 500
expect_1000b_to_compress_to = 500
dump_to_memory = 0
dump_allow_full_to_memory = 0
leave_dumps_in_memory = 0
dump_user_pte_pages = 0
live_dump_zero_suppress = 1
live_dump_dir_name = /var/adm/crash
include_user_ptes_in_dumps = 0
lite_system = 0
physio_max_coalescing = 65536
kmem_percent = 25
kmemreserve_percent = 0
kmem_debug = 0
kmem_debug_size_mask = 0
kmem_protected_size = 0
kmem_protected_lowat = 1000
kmem_protected_hiwat = 0
kmem_protected_kmempercent = 75
kmem_audit_count = 1024
kmemhighwater_16 = 4
.
.
.
kmemhighwater_12k = 4
old_obreak = 1
user_cfg_pt = 45000
memstr_buf_size = 0
memstr_start_addr = 0
memstr_end_addr = 0
memlimit = 0
insecure_bind = 0
memberid = 0
memberseq = 0
clu_configured = 0
clu_active_member = 0
old_vers_high = 0
old_vers_low = 0
act_vers_high = 1441151880873377792
act_vers_low = 15044
new_vers_high = 1441151880873377792
new_vers_low = 15044
versw_switch = 0
```

```
versw_transition = 0
rolls_ver_lookup = 0
login_name_max = 12
enable_async_printf = 1
```

(This display output has been truncated.)

4.5.6.2 Identifying Run-time Configurable Subsystem Attributes

You can identify which of a subsystem's attributes are configurable at run time using the `/sbin/sysconfig -Q` command:

```
# /sbin/sysconfig -Q vfs max-vnodes
vfs:
max-vnodes -    type=INT op=CRQ min_val=0 max_val=1717986918
```

This example shows using the `-Q` option to get information about the `max-vnodes` attribute of the `vfs` subsystem. The `max-vnodes` attribute has the `integer` datatype, a minimum value of zero (0), and a maximum value of 1717986918. The `op` field indicates the operations that you can perform on the `max-vnodes` attribute. The following list describes the values that can appear in this field:

C — You can modify the attribute when the subsystem is loaded initially.

R — You can modify the attribute while the subsystem is running.

Q — You can query the attribute for information.

4.5.6.3 Modifying Attribute Values at Run Time

You can modify the value of an attribute at run time using the `/sbin/sysconfig -r` command, `dxkerneltuner` or the source level debugger `dbx`. The modification you make persists until the next time the system is rebooted. When the system reboots, any changes made with the `/sbin/sysconfig -r` command are lost because the new value is not stored. The `-r` option to the `/sbin/sysconfig` command is useful for testing a new subsystem attribute value. If the new value causes the system to perform as expected, you can store it in the subsystem attribute database as described in Section 4.5.8. See `dbx`(1) and the *System Configuration and Tuning* manual for more information.

When you use the `/sbin/sysconfig -r` command you specify the attribute, its new value, and the subsystem name on the command line. For example, to modify the `dump-sp-threshold` attribute for the `generic` subsystem, enter a command similar to the following:

```
# /sbin/sysconfig -r generic dump-sp-threshold=20480
```

To modify the value of more than one attribute at a time, include a list on the /sbin/sysconfig command line. For example, to modify the dump-sp-threshold attribute and the locktimeout attribute, enter a command similar to the following:

```
# /sbin/sysconfig -r generic dump-sp-threshold=20480 \
locktimeout=20
```

You do not include a comma between the two attribute specifications.

To make the attribute value permanent, you must add it to the /etc/sysconfigtab file using the appropriate method, for example, by specifying it as a boot time value using the Kernel Tuner graphical user interface.

4.5.7 Managing Subsystems and Attributes Remotely

You can use the /sbin/sysconfig -h command to administer configurable subsystems and dynamic subsystem attributes remotely on a local area network (LAN). This allows you to administer several systems from a single machine.

Each system you want to administer remotely must have an /etc/cfgmgr.auth file that contains the full domain name of the local system. The name in the /etc/cfgmgr.auth file should be identical to the name in either the /etc/hosts file or in the Berkeley Internet Domain (BIND) or Network Information Service (NIS) hosts databases, if you are using BIND or NIS. You must create the /etc/cfgmgr.auth file; it is not on your system by default. The following shows an example cfgmgr.auth file:

```
salmon.zk3.dec.com
trout.zk3.dec.com
bluefish.zk3.dec.com
```

To manage subsystems and attributes on remote systems, you include the -h option and a host name with the /sbin/sysconfig command. For example, to load the environmental monitoring lat subsystem on a remote host named MYSYS, enter the following command:

```
# /sbin/sysconfig -h MYSYS -c lat
```

In this example, a lat.mod file must exist in either the /subsys directory or the /var/subsys directory on the remote system before you can load the specified subsystem. If the loadable subsystem subset is kitted correctly, the *subsystem-name*.mod file is installed on the remote system when you use the setld command to install the loadable subsystem.

4.5.8 Managing the Subsystem Attributes Database

Information about dynamically configurable subsystem attributes is stored in the /etc/sysconfigtab database. This database records the values assigned to subsystem attributes each time the system is rebooted or a subsystem is configured. No attributes are set automatically in this database. You must be the superuser to modify the /etc/sysconfigtab database and you must use the sysconfigdb command line utility or Kernel Tuner graphical user interface to make the change. See the Kernel Tuner's online help for more information on using that method.

Note

The /etc/sysconfigtab database may contain stanza entries from a configurable subsystem's stanza.loadable file. This file and the entry in the /etc/sysconfigtab database are created automatically when you install certain configurable subsystems. You should not modify these entries in the database.

There are multiple numbered versions of the sysconfigtab.* file in the /etc directory, but only the /etc/sysconfigtab version is used during normal operations. The versions are present to support the dynamic linking of modules to create a /vmunix kernel. This feature is called bootlinking and is documented in the *Guide to Preparing Product Kits* manual. You may not be able to use bootlinking if you delete any copies of the sysconfigtab.* file.

To add, update, or remove entries in the database, you create a stanza-format file containing names and values for attributes you want to modify. See stanza(4)) for information about stanza-format files. For example, suppose you want to set the lockmode attribute in the generic subsystem to 1. To set this attribute, create a file named, for example, generic_attrs that has the following contents:

```
generic:
        lockmode = 1
```

After you create the stanza-format file, you use the /sbin/sysconfigdb command to update the /etc/sysconfigtab database. You name the stanza-format file on the command line using the -f. option. The sysconfigdb command reads the specified file and updates both the on-disk and in-memory copy of the database. However, the running kernel is not updated. Use the sysconfig -r command to update the running kernel, as described in Section 4.5.6.3.

The sysconfigdb command has the following syntax options:

/sbin/sysconfigdb {-s}

/sbin/sysconfigdb -t *outfile* [-f *infile* -a | -u *subsystem-name*]

/sbin/sysconfigdb -t *outfile* [-f *infile* -m | -r *subsystem-name*]

/sbin/sysconfigdb -t *outfile* [-f *infile* -d *subsystem-name*]

/sbin/sysconfigdb -t *outfile* [-f *infile* -l [*subsystem-name...*]]

The following sections explain how to use the `/sbin/sysconfigdb` command to manage entries in the `/etc/sysconfigtab` database.

4.5.8.1 Listing Attributes in the Database

To list the entries in the `/etc/sysconfigtab` database, use the `/sbin/sysconfigdb -l` command. If you specify a subsystem name on the command line, the attributes of that subsystem are listed. Otherwise, all attributes defined in the database are listed.

For example, to list the attribute settings for the `generic` subsystem, enter the following command:

```
# /sbin/sysconfigdb -l generic
generic:
        memberid = 0
        new_vers_high = 1441151880873377792
        new_vers_low = 15044
```

4.5.8.2 Adding Attributes to the Database

To add subsystem attributes to the `/etc/sysconfigtab` database, enter the `sysconfigdb -a` command.

For example, to add the entries stored in a file named `add_attrs` to the database, enter the following command:

```
# /sbin/sysconfigdb -a -f add_attrs generic
```

4.5.8.3 Merging New Definitions into Existing Database Entries

To merge new definitions for attributes into an existing entry in the `/etc/sysconfigtab` database, enter the `sysconfigdb –m` command.

The `sysconfigdb` command merges the new definitions into the existing database entry as follows:

- If an attribute name does not appear in the database, the definition for that attribute is added to the database.
- If an attribute name does appear, the attribute receives the value specified by the new definition.

- If an attribute appears in the database, but is not included among the new definitions, its definition is maintained in the database.

For example, assume that the following entry for the `generic` subsystem already exists in the `/etc/sysconfigtab` database:

```
generic:
   lockmode = 4
   dump-sp-threshold = 6000
```

You then create a file named `merge_attrs` for updating this entry, which contains the following information:

```
generic:
  lockmode = 0
  lockmaxcycles = 4294967295
```

To merge the information in the `merge_attrs` file into the `/etc/sysconfigtab` database, enter the following command:

```
# /sbin/sysconfigdb -m -f merge_attrs generic
```

After the command finishes, the entry for the `generic` subsystem in the database appears as follows:

```
generic:
   lockmode = 0
   lockmaxcycles = 4294967295
   dump-sp-threshold = 6000
```

You can merge definitions for more than one subsystem into the `/etc/sysconfigtab` database with a single `sysconfigdb -m` command. For example, the `merge_attrs` file could contain new definitions for attributes in the `lsm` and `generic` subsystems. If you include more than one subsystem name in the `merge_attrs` file, you omit the subsystem name from the command line, as shown:

```
# /sbin/sysconfigdb -m -f merge_attrs
```

4.5.8.4 Updating Attributes in the Database

Use the `/sbin/sysconfigdb -u` command to update a subsystem that is already in the `/etc/sysconfigtab` database. For example, suppose the `generic` subsystem is defined as follows in the `/etc/sysconfigtab` file:

```
generic:
   lockmode = 4
   dump-sp-threshold = 6000
```

Suppose that you create a file named `update_attrs` for updating this entry, which contains the following information:

```
generic:
  lockmode = 0
```

```
  lockmaxcycles = 4294967295
```

To update the attributes, you enter the sysconfigdb command, as follows:

```
# /sbin/sysconfigdb -u -f update_attrs generic
```

After the command finishes, the entry for the generic subsystem in the database appears as follows:

```
generic:
   lockmode = 0
   lockmaxcycles = 4294967295
```

4.5.8.5 Removing Attribute Definitions from the Database

To remove the definitions of selected attributes from the /etc/sysconfigtab database, enter the /sbin/sysconfigdb -r command. The -r option specifies that you want to remove the attribute definitions stored in a file from the database.

For example, suppose the generic subsystem is defined as follows in the /etc/sysconfigtab database:

```
generic:
  lockmode = 4
  dump-sp-threshold = 6000
```

To remove the definition of the dump-sp-threshold attribute, first create a file named remove_attrs that contains the following information:

```
generic:
   dump-sp-threshold = 6000
```

Then, enter the following command:

```
# /sbin/sysconfigdb -r -f remove_attrs generic
```

After the command finishes, the entry for the generic subsystem in the database appears as follows:

```
generic:
   lockmode = 4
```

The /sbin/sysconfigdb command removes only identical entries. In other words, the entries must have the same attribute name and value to be removed.

You can remove definitions of more than one attribute and for attributes with a single sysconfigdb -r command. For example, the remove_attrs file could contain attribute definitions that you want to remove for the lsm and generic subsystems. If you include more than one subsystem in the remove_attrs file, you omit the subsystem name from the command line, as shown:

```
# /sbin/sysconfigdb -r -f remove_attrs
```

4.5.8.6 Deleting Subsystem Entries from the Database

To delete the definition of a subsystem from the `/etc/sysconfigtab` database enter the `/sbin/sysconfigdb -d` command.

For example, to delete the `generic` subsystem entry in the database, enter the following command:

```
# /sbin/sysconfigdb -d generic
```

The `generic` subsystem receives its default values the next time it is configured.

4.6 Static System Configuration

Static system configuration refers to the commands and files used to build and boot a new kernel and its static subsystems. The subsystems are viewed as static because they are linked directly into the kernel at build time. The steps you take to build a statically linked kernel vary depending upon why you want to modify the kernel.

If you modify the kernel to add a device driver, you follow these general steps:

- Install the device driver.
- If necessary, edit the target configuration file.

  In some cases, the device driver provides a Subset Control Program (SCP) that executes during the installation procedure and registers the driver in the necessary system configuration files. In this case, you need not edit the target configuration file yourself.

  If the device driver does not provide an SCP, you must edit the target configuration file yourself.
- Build a new kernel.

  If your device driver includes an SCP, build a new kernel by running the `/usr/sbin/doconfig` program as described in Section 4.6.3. If you need to edit the target configuration file before you build a new kernel, see Section 4.6.1.
- Shut down and reboot your system.

If you modify the kernel to add support for certain kernel options, you can build the new kernel by running the `/usr/sbin/doconfig` program and choosing the kernel option from a menu displayed during processing. You then shut down and reboot your system.

To determine which kernel options you can configure in this way, enter the `/usr/sbin/kopt` command. The command displays a list of kernel options and prompts you for kernel options selections. To exit from the `/usr/sbin/kopt` command without choosing options, press the Return key.

For information about running the /usr/sbin/doconfig program to add kernel options using a menu, see Section 4.6.2.

If you build a new static kernel for any other reason, you must modify one or more system files as part of rebuilding the kernel. The system files you modify depend upon the change you want to make to the kernel:

- You modify the target configuration file to make changes to keywords that, for example, define the kernel you want to build, define devices, or define pseudodevices. Also, you can edit this file to change the value of system parameters. For details about the contents of the target configuration file, see Section 4.7.
- You remove certain static subsystems from the kernel by removing (or commenting out) their entry from a file in the /usr/sys/conf directory. For information about this file, see Section 4.7.2.

For information about running the /usr/sbin/doconfig program to build a kernel after editing system files, see Section 4.6.3.

For examples of adding and configuring devices, see the *Hardware Management* manual.

4.6.1 Building the Kernel to Add Support for a New Device

When you add a new device to the system and the device installation includes no SCP, you must edit the target configuration file to allow the operating system to support the new device. You include the device definition keyword in the target configuration file. Because the operating system supports many devices, determining which keyword to add to your target configuration file can be difficult.

The following procedure explains how to determine which device definition keyword to add to your target configuration file. It also explains how to rebuild the kernel after you have edited the target configuration file. The procedure assumes that you do not know the appropriate keyword to add. In some cases, you may be able to determine the appropriate keyword by looking at documentation supplied with the hardware or with a new version of the operating system. Another source of this information is an existing configuration file on another system that already has the device connected to it. If you know what keyword you need to add to your system, use a utility to add that keyword to your target configuration file and rebuild the kernel as described in Section 4.6.3.

Caution

This procedure is risky and you should ensure that you have a copy of your custom /vmunix kernel file, a copy of the generic

kernel /genvmunix, and copies of the current configuration files. You may need the copies to revert to your previous configuration.

If you are unsure of the keyword you need to add to the target configuration file for your system, connect the new device to the system as directed in the hardware manual and use the following procedure:

1. Make certain you have a copy of the generic kernel, /genvmunix; you need to boot it later in the procedure. If the /genvmunix file does not exist on your system, or the generic kernel fails to recognize the device you are adding, copy the generic kernel from the software distribution media. See the *Installation Guide* for information on the location of the generic kernel on the distribution CD-ROM.

   In rare cases, you may need to rebuild the generic kernel. However, you cannot rebuild the generic kernel if you have installed any layered applications or a third-party device driver. In this case, if the original /genvmunix is corrupt or has been deleted, and you have no distribution media you should contact your technical support organization and obtain a copy of the generic kernel /genvmunix.

   To verify whether layered applications have been installed, verify the contents of the usr/sys/conf directory for a file named .product.list.

   To rebuild the generic kernel, you must have installed all the required and optional kernel subsets. You can get a list of the kernel build subsets, including information about whether or not they are installed, by issuing the following command:

   ```
   # /usr/sbin/setld -i | grep Kernel Build
   ```

   After all kernel subsets are installed, enter the following command:

   ```
   # doconfig -c GENERIC
   ```

   The -c option specifies that you want to build a kernel using an existing configuration file, in this case the GENERIC configuration file. For more information about building a kernel from an existing configuration file, see Section 4.6.3.

   After the generic kernel is running and recognizes the new device, continue with step 5. When the build ends, consider using the strip command to reduce the size of the kernel. See strip(1) for more information.

2. Log in as root or become the superuser and set your default directory to the /usr/sys/conf directory.

3. Save a copy of the existing /vmunix file. If possible, save the file in the root (/) directory, as follows:

```
# cp /vmunix /vmunix.save
```

If there are disk space constraints, you can save the kernel file in a file system other than root. For example:

```
# cp /vmunix /usr/vmunix.save
```

4. Shut down and halt the system as follows:

```
# shutdown -h now
```

5. At the console prompt, boot the generic kernel, `/genvmunix`. The generic kernel contains support for all valid devices, so if you boot it during the process of adding a new device to your target kernel, the generic kernel already knows the new device.

   To boot the generic kernel, enter the following command:

```
>>> boot -fi "genvmunix"
```

6. At the single-user mode prompt, verify and mount local file systems by issuing the following command, unless you are using the Logical Storage Manager software (LSM):

```
# /sbin/bcheckrc
```

   If you are using the Logical Storage Manager (LSM) software, verify the local file systems and start LSM by issuing the following command:

```
# /sbin/lsmbstartup
```

7. Run the `sizer` program to size your system hardware and create a new target configuration file that includes the new device:

```
# sizer -n MYSYS
```

   The `sizer -n` command creates a new target configuration file for your system that includes the appropriate device definition keyword for the new device; this process is similar to the process that occurs at system installation time. See Section 4.3 for more information. The `sizer` program stores the new target configuration file in the `/tmp` directory.

8. Compare the new target configuration file created by `sizer` with the existing target configuration file for your system:

```
# diff /tmp/MYSYS MYSYS
```

   Examine the differences between these files until you find the new device definition keyword. (The two files may differ in other ways if you have customized your existing configuration file, such as by specifying a nondefault value for the `maxusers` option.)

9. Use the text editor of your choice to add the new device definition keyword to your existing configuration file (in this case, MYSYS). Adding the new keyword allows your existing configuration file to support the new device, without losing any changes you made to that file in the past.

Note

If you add or remove communications devices from your configuration file, you must edit the /etc/inittab file and the /etc/securettys file to match your new configuration; that is, to match the /dev/tty*nn* special device files. See inittab(4) and securettys(4) for more information.

10. Build a new kernel by issuing the following /usr/sbin/doconfig command:

```
# /usr/sbin/doconfig -c MYSYS

*** KERNEL CONFIGURATION AND BUILD PROCEDURE ***

Saving /usr/sys/conf/MYSYS as /usr/sys/conf/MYSYS.bck
```

Answer the following prompt to indicate that you do not want to edit the configuration file:

```
Do you want to edit the configuration file? (y/n) [n]: n
*** PERFORMING KERNEL BUILD ***
.
.
.
The new kernel is /usr/sys/MYSYS/vmunix
```

11. When the kernel configuration and build process completes without errors, copy the new vmunix file to /vmunix. On a system named MYSYS, enter the following command:

```
# cp /usr/sys/MYSYS/vmunix /vmunix
```

Always use copy (cp) instead of move (mv) to preserve the context dependent symbolic link (CDSL). See Chapter 6 for more information on CDSLs.

12. Reboot the system as follows:

```
# /usr/sbin/shutdown -r now
```

If the new /vmunix file fails to boot, boot using the kernel you saved at the beginning of the procedure. To use the saved kernel, follow these steps:

1. Verify all local file systems by using the fsck –p command as follows:

```
# fsck -p
```

2. Write-enable the root file system by using the mount -u command as follows:

```
# mount -u /
```

3. If necessary, mount the file system where the /vmunix.save file is stored. For example, if you copied the /vmunix file to the /usr file system, enter the following command:

   ```
   # mount /usr
   ```

4. Restore the saved copy. For example, if you saved your running kernel in the /vmunix.save file, enter the following command:

   ```
   # mv /vmunix.save /vmunix
   ```

5. Shut down and reboot the system, as follows:

   ```
   # shutdown -r now
   ```

After your system is running again, you can modify the target configuration file as needed and rebuild the kernel starting at step 3.

4.6.2 Building the Kernel to Add Selected Kernel Options

If you invoke the /usr/sbin/doconfig program without using options, you are given the opportunity to modify the kernel using a menu. To modify the kernel using a menu, follow these steps:

1. Log in as root or become the superuser and set your default directory to the /usr/sys/conf directory.
2. Save a copy of the existing /vmunix file. If possible, save the file in the root (/) directory, as follows:

   ```
   # cp /vmunix /vmunix.save
   ```

   If there are disk space constraints, you can save the kernel file in a file system other than root. For example:

   ```
   # cp /vmunix /usr/vmunix.save
   ```

3. Run the /usr/sbin/doconfig program using no options, as follows:

   ```
   # /usr/sbin/doconfig

   *** KERNEL CONFIGURATION AND BUILD PROCEDURE ***

   Saving /usr/sys/conf/MYSYS as /usr/sys/conf/MYSYS.bck
   ```

4. Enter the name of the configuration file at the following prompt:

   ```
   Enter a name for the kernel configuration \
   file. [MYSYS]: MYSYS
   ```

   The kernel configuration processes convert the system name to uppercase when determining what name to supply as the default configuration file name. For example, on a system named mysys, the default configuration file is named MYSYS.

If the configuration file name you specify does not currently exist, the /usr/sbin/doconfig program builds one with that name. Continue this process by selecting the kernel options in step 10.

5. If the configuration file name you specify exists, answer the following prompt to indicate that you want to overwrite it:

```
A configuration file with the name MYSYS already exists.
Do you want to replace it? (y/n) [n]: y
```

6. Select kernel options from a menu similar to the following one:

```
*** KERNEL OPTION SELECTION ***
------------------------------------------------------------

  Selection  Kernel Option
------------------------------------------------------------
     1     System V Devices
     2     NTP V3 Kernel Phase Lock Loop (NTP_TIME)
     3     Kernel Breakpoint Debugger (KDEBUG)
     4     Packetfilter driver (PACKETFILTER)
     5     Point-to-Point Protocol (PPP)
     6     STREAMS pckt module (PCKT)
     7     Data Link Bridge (DLPI V2.0 Service Class 1)
     8     X/Open® Transport Interface (XTISO, TIMOD, TIRDWR)
     9     ISO 9660 Compact Disc File System (CDFS)
     10    Audit Subsystem
     11    Alpha CPU performance/profiler (/dev/pfcntr)
     12    ACL Subsystem
     13    Logical Storage Manager (LSM)
     14    ATM UNI 3.0/3.1 ILMI (ATMILMI3X)
     15    IP Switching over ATM (ATMIFMP)
     16    LAN Emulation over ATM (LANE)
     17     Classical IP over ATM (ATMIP)
     18    ATM UNI 3.0/3.1 Signalling for SVCs (UNI3X)
     19    Asynchronous Transfer Mode (ATM)
     20    All of the above
     21    None of the above
     22    Help
     23    Display all options again
------------------------------------------------------------

Enter the selection number for each kernel option you want.
For example, 1 3 [18]:
```

7. Answer the following prompt to indicate whether or not you want to edit the configuration file:

```
Do you want to edit the configuration file? (y/n) [n]:
```

You need not edit the configuration file unless you have changes other than adding one or more of the subsystems in the menu to the kernel.

If you choose to edit the configuration file, the /usr/sbin/doconfig program invokes the editor specified by the EDITOR environment variable.

For information about the configuration file, see Section 4.7

After you finish editing the configuration file, the /usr/sbin/doconfig program builds a new kernel.

8. When the kernel configuration and build process is completed without errors, move the new vmunix file to /vmunix. On a system named MYSYS, enter the following command:

   # **mv /usr/sys/MYSYS/vmunix /vmunix**

9. Reboot the system as follows:

   # **/usr/sbin/shutdown -r now**

If the new /vmunix file fails to boot, boot using the kernel you saved at the beginning of the procedure. To use the saved kernel, follow these steps:

1. Verify all local file systems by using the fsck -p command as follows:

   # **fsck -p**

2. Write-enable the root file system using the mount -u command as follows:

   # **mount -u /**

3. If necessary, mount the file system where the /vmunix.save file is stored. For example, if you copied the /vmunix file to the /usr file system, enter the following command:

   # **mount /usr**

4. Restore the saved copy. For example, if you saved your running kernel in the /vmunix.save file, enter the following command:

   # **mv /vmunix.save /vmunix**

5. Shut down and reboot the system, as follows:

   # **shutdown -r now**

After your system is running again, you can modify the target configuration file as needed and rebuild the kernel starting at step 3.

4.6.3 Building a Kernel After Modifying System Files

If you or an SCP modify system files, such as the target configuration file, you can rebuild your kernel using the /usr/sbin/doconfig -c command. The -c option allows you to name an existing configuration file, which the

/usr/sbin/doconfig program uses to build the kernel. To build a new kernel using an existing configuration file, follow these steps:

1. Log in as root or become the superuser and set your default directory to the /usr/sys/conf directory.
2. Save a copy of the existing /vmunix file. If possible, save the file in the root (/) directory, as follows:

   ```
   # cp /vmunix /vmunix.save
   ```

   If there are disk space constraints, you can save the kernel file in a file system other than root. For example:

   ```
   # cp /vmunix /usr/vmunix.save
   ```

3. Run the /usr/sbin/doconfig program specifying the name of the target configuration file with the -c option. For example on a system named MYSYS, enter the following command:

   ```
   # /usr/sbin/doconfig -c MYSYS

   *** KERNEL CONFIGURATION AND BUILD PROCEDURE ***

   Saving /usr/sys/conf/MYSYS as /usr/sys/conf/MYSYS.bck
   ```

4. Answer the following prompt to indicate whether or not you want to edit the configuration file:

   ```
   Do you want to edit the configuration file? (y/n) [n]:
   ```

   If you modified the configuration file before you started this procedure, indicate that you do not want to edit the configuration file.

   If you choose to edit the configuration file, the /usr/sbin/doconfig program invokes the editor specified by the EDITOR environment variable.

   For information about the configuration file, see Section 4.7.

   After you finish editing the configuration file, the /usr/sbin/doconfig program builds a new kernel.

5. When the kernel configuration and build are completed without errors, move the new vmunix file to /vmunix. On a system named MYSYS, enter the following command:

   ```
   # mv /usr/sys/MYSYS/vmunix /vmunix
   ```

6. Reboot the system as follows:

   ```
   # /usr/sbin/shutdown -r now
   ```

If the new /vmunix file fails to boot, boot using the kernel you saved at the beginning of the procedure. To use the saved kernel, follow these steps:

1. Verify all local file systems by using the fsck –p command as follows:

```
# fsck -p
```

2. Write-enable the `root` file system using the `mount -u` command as follows:

```
# mount -u /
```

3. If necessary, mount the file system where the `/vmunix.save` file is stored. For example, if you copied the `/vmunix` file to the `/usr` file system, enter the following command:

```
# mount /usr
```

4. Restore the saved copy. For example, if you saved your running kernel in the `/vmunix.save` file, enter the following command:

```
# mv /vmunix.save /vmunix
```

5. Shut down and reboot the system, as follows:

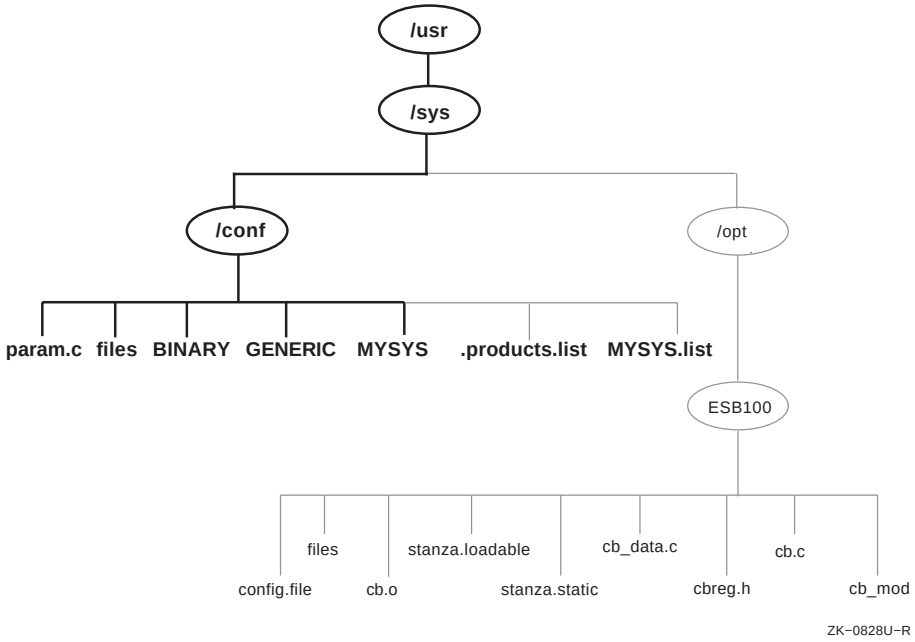
```
# shutdown -r now
```

After your system is running again, you can modify the target configuration file as needed and rebuild the kernel starting at step 3.

4.7 Configuration Files

To build and run a working kernel, the system depends on the presence of specific directories under the `/usr/sys` directory. Figure 4–1 shows the directory structure of the system configuration files. The dotted lines indicate optional directories and files for third-party static subsystems.

Figure 4–1: Configuration Files Directory Hierarchy



As shown in Figure 4–1, the `/usr/sys/conf` directory contains files that define the kernel configuration for the generic and target kernels. These files represent the configuration of the static portion of the kernel. When you work with the system files to reconfigure the kernel, you are interested primarily in five files:

- `/usr/sys/conf/`*`MYSYS`*, where *MYSYS* is the system name.
- `/usr/sys/conf/GENERIC`
- `/usr/sys/conf/.product.list`
- `/usr/sys/conf/`*`NAME`*`.list`
- `/usr/sys/conf/param.c`

The following sections provide more information about these files.

4.7.1 Configuration Files in /usr/sys/conf

The `/usr/sys/conf` directory contains two important system configuration files: the target configuration file and the GENERIC configuration file.

4.7.1.1 The Target Configuration File

The target configuration file, `/usr/sys/conf/`*`NAME`*, is a text file that defines the components that the system builds into your kernel. By

convention, the *NAME* portion of the pathname is the name of your system in capital letters. For example, a system named MYSYS is described by a file named /usr/sys/conf/MYSYS. Each system has a target configuration file built for it by the sizer program during system installation. You modify the target configuration file when you want to change one of the following keyword definitions:

- Global keywords that, if you are managing more than one system, are often defined the same across systems
- System definition keywords that describe the kernel you want to build for a particular system
- Device definition keywords that describe the devices connected to a particular system
- callout keyword definitions that allow you to run shell command subprocesses during kernel configuration
- options keyword definitions that specify software to be compiled into the system
- makeoptions keyword definitions that are passed to the compiler, assembler, and linker when building the kernel
- pseudodevice keyword definitions that describe pseudodevices used on the system

4.7.1.2 The GENERIC Configuration File

The /usr/sys/conf/GENERIC configuration file is the configuration file that describes the generic kernel. The generic kernel supports all valid devices and is useful when you are adding a new device to the system. Also, you can use the generic kernel as a backup kernel should your target kernel be corrupted in some way.

Avoid deleting the /genvmunix file, which contains the generic kernel. If you accidentally delete the generic kernel, you can rebuild it by using the doconfig -c GENERIC --c GENERIC command. For more information about building a kernel using an existing configuration file, see Section 4.6.3.

Note

Never delete the /usr/sys/conf/GENERIC file.

4.7.2 Extensions to the Target Configuration File

The /usr/sys/conf directory contains two optional configuration files that describe extensions to the target configuration file. These are the /usr/sys/conf/.product.list file and the /usr/sys/conf/*NAME* file.

These files store information about static kernel subsystems, sometimes called kernel layered products.

When you install a static subsystem, its SCP normally edits the `/usr/sys/conf/.product.list` file and adds an entry for the subsystem. After the SCP has completed, run the `/usr/sbin/doconfig` program to configure the new subsystem into the kernel.

The `/usr/sbin/doconfig` program creates the *NAME*`.list` file. The *NAME* variable is the same as the target configuration file, and by convention is your system name in capital letters. For example, the *NAME*`.list` file for a system named MYSYS is `MYSYS.list`.

If you need to modify your system because of a third-party layered product (for example, to remove a layered product from the kernel being built), make the necessary modifications to the *NAME*`.list` file and build a new kernel.

Each entry in the *NAME*`.list` file consists of six fields separated by a colon (`:`). The following example is part of a *NAME*`.list` file and shows an entry for a static kernel subsystem that has been loaded into the `/usr/sys/opt/ESB100` directory:

```
/usr/sys/opt/ESB100:UNXDASH100:920310100739:DASH Systems:controlsys:100
    1          2      3       4      5    6
```

The fields in this entry contain the following information:

1. The full pathname where the system configuration tools find extensions to input data. This location can contain files such as:
   - Product-specific configuration files
   - The `config.file` file fragment (contains keywords related only to the product)
   - The `files` file fragment (contains information about the location of the product's source code, when the product should be loaded into the kernel, and whether source or binary code is provided)
   - The `stanza.static` file (contains information about a static driver's major number requirements and the names and minor numbers of the device special files)
   - Object files
   - Source code files
2. The `setld` subset identifier.
3. The date and time that the product is ready for distribution.
4. The name of the company that provided the subsystem.
5. The product name.

6 The `setld` 3-digit product version code.

The order of the line entries in the *NAME*`.list` file reflects the order in which the entries are processed.

The `/usr/sbin/doconfig` program creates the *NAME*`.list` file by copying the `.product.list` file, if it exists. When you use the `/usr/sbin/doconfig -c` command option, `/usr/sbin/doconfig` uses the existing *NAME*`.list` file. If the `.product.list` file changes (for example, a new kernel layered product was installed) and the `-c` option is used, either delete the *NAME*`.list` file or manually edit it before invoking `/usr/sbin/doconfig` to propagate the change in the `.product.list` file to the *NAME*`.list` file.

You can create the file by copying the `.product.list` file to the *NAME*`.list` file. Then you can edit the *NAME*`.list` file and either delete the lines that you do not want built into the kernel or comment them out by putting a number sign (`#`) as the first character in each line that you do not want.

Note

Never edit the `.product.list` file.

See the device driver documentation for more information on the *NAME*`.list` and `.product.list` files.

4.7.3 The param.c File

The `param.c` file contains default values for a number of system parameters. Do not modify these parameters unless instructed to do so in a document or by your technical support organization.

The precedence order in which attribute values are read and used is as follows:

1. The run-time value of the attribute.
2. The value recorded in the `/etc/sysconfigtab` file.
3. The value recorded in the `/usr/sys/conf/`*SYSTEM_NAME* file.
4. The value recorded in the `/usr/sys/conf/param.c` file.

In some cases, a parameter in the `param.c` file also exists in your target configuration file. In this case, a value specified in the configuration file overrides the value specified in the `param.c` file. Therefore, if you modify the value of a system parameter in the `param.c` file, be sure to remove the corresponding entry from the target configuration file.

4.7.4 System Configuration File Entries

The system configuration file contains the following keyword definitions:

- Global keyword definitions
- System definition keywords
- Device definition keywords
- `callout` keyword definitions
- `options` keyword definitions
- `makeoptions` keyword definitions
- `pseudo-device` keyword definitions

Avoid performing manual tuning or custom configuration options in this file. See the *System Configuration and Tuning* manual for information on configuring a kernel and tuning it.

5

Administering Disks

This chapter discusses system administration tasks related to the administration of disks, including:

- A discussion on partitioning disks using a graphical user interface (Section 5.1)
- A description on partitioning disks manually (Section 5.2)
- A discussion on copying disks (Section 5.3)

See the *AdvFS Administration* manual for information on administering AdvFS.

5.1 Partitioning Disks Using the Disk Configuration Utility

The Disk Configuration graphical user interface (GUI), `diskconfig`, enables you to perform the following tasks:

- Display attribute information for existing disks
- Modify disk configuration attributes
- Administer disk partitions
- Create AdvFS and UFS file systems on a disk partition
- Administer disk aliases

See `diskconfig`(8) for information on invoking the Disk Configuration GUI (`diskconfig`). An online help volume describes how you use the GUI.

Invoke the Disk Configuration GUI using either of these methods:

- Enter `/usr/sbin/diskconfig` at the system prompt.
- Use the following procedure:

  1. Select the SysMan Applications pop-up menu from the CDE Front Panel.
  2. Select Configuration. The SysMan Configuration folder opens.
  3. Select the Disk icon from the SysMan Configuration folder.

Caution

Disk Configuration displays appropriate warnings when you attempt to change partition sizes. However, you should plan the changes in advance to ensure that you do not overwrite any required data. Back up any data partitions before attempting this task.

A window titled `Disk Configuration on` *`hostname`* is displayed. This is the main window for the Disk Configuration application, and lists each disk on the system, and gives the following information for each disk:

- The disk basename, such as `dsk10`. See the *Hardware Management* manual for information on disk names.
- The device model, such as `RZ1CB-CA`.
- The physical location of the device, specifying Bus, Target and LUN (logical unit number). See *Hardware Management* manual for information on the device location.

Select a device in the list then select `Configure...` to configure a specific disk; an alternate method is double clicking on the device in the list. Two windows, `Configure Partitions` and `Partition Table`, open.

Disk Configuration: Configure Partitions: *`device name device type`*

This window allows you to set the start address, end address, size, and usage options for the disk partitions.

Disk Configuration: Partition Table: *`device name device type`*

This window displays the current partitions, their start addresses, their end addresses, and their sizes.

See the online help for more information on these windows.

After making partition adjustments, use the SysMan Menu options to mount any newly created file systems as follows:

1. Add the new file system to the `/etc/fstab` file.
2. Invoke the SysMan Menu. See Chapter 1 for further information.
3. Select `File Systems`. Additional options are displayed.
4. Select `General File System Utilities`. Additional options are displayed.
5. Select `Mount File Systems`. The Mount File Systems main window opens.

6. Select `Mount an AdvFS/UFS file system`, then select `Next>`. The Steps dialog box opens.
7. Select `Next>`. The Select File System dialog box opens.
8. Select the file system in the List of File Systems, then enter `Next>`. The Mount dialog box opens.
9. Enter a mount point, such as `/usr/newusers` in the Mount Directory field.

   Alternatively, use `Browse...` to find and select a directory for the mount point.
10. Select the Access Mode.
11. Select `Next>`. The Summary dialog box opens.
12. Select `Finish`.
13. Select `Exit` to close the SysMan Menu.

Your new file system is now accessible.

5.1.1 Configure Partitions Window

This window provides the following information and options:

- A graphical representation of the disk partitions, in a horizontal bar-chart format. The currently-highlighted partition is a different color, and the details of that partition are displayed in the Selected Partition box. You can use the bar chart handles (or flags) to change the partition sizes. Position the cursor as follows:
  - On the center handle to change both adjacent partitions
  - On the top flag to move up the start of the right hand partition
  - On the bottom flag to move down the end of the left hand partition

  Press MB1 and drag the mouse to move the handles.
- A pull-down menu that enables you to toggle the sizing information between megabytes, bytes, and blocks.
- A statistics box, which displays disk information such as the device name, the total size of the disk, and usage information. This box enables you to assign or edit the disk label, and create an alias name for the device.
- The Selected Partition box, which displays dynamic sizes for the selected partition. These sizes are updated as you change the partitions by using the bar-chart. You can type the partition sizes directly into these windows to override the current settings. This box also enables you to select the file system for the partition and, if using AdvFS, the domain name and fileset name.

- The Disk Attributes... option.

  This button displays some of the physical attributes of the device.
- The Partition Table... option, which is described in the following section.

5.1.2 Partition Table Window

This window displays a bar-chart of the current partitions in use, their sizes, and the file system in use. You can toggle between the current partition sizes, the default table for this device, and the original (starting table) when this session started. If you make errors on a manual partition change, you can use this window to reset the partition table.

5.2 Manually Partitioning Disks

The following sections provide the information you need to change the partition scheme of your disks. In general, you allocate disk space during the initial installation or when adding disks to your configuration. Usually, you do not have to alter partitions; however, there are cases when it is necessary to change the partitions on your disks to accommodate changes and to improve system performance.

5.2.1 Utilities

These commands allow you to perform several disk maintenance tasks manually:

`disklabel`	Use this command to install, examine, or modify the label on a disk drive or pack. The disk label contains information about the disk, such as type, physical parameters, and partitioning. See `disklabel`(4) for more information on the `/etc/disktab` file.
`newfs`	Use this command to create a new UFS file system on the specified device. Do not use the `newfs` command to create Advanced File System (AdvFS) domains; use the `mkfdmn` command instead.
`mkfdmn` and `mkfset`	Use these commands to create Advanced File System (AdvFS) domains and filesets. See `mkfdmn`(8) for more information.

5.2.2 Using the disklabel Utility

The disk label provides detailed information about the geometry of the disk and the partitions into which the disk is divided. As root, you can change

the label with the `disklabel` command. See `disklabel`(8) for information on command options.

There are two copies of a disk label, one located on the disk and one located in system memory. Because it is faster to access system memory than to perform I/O, when the system boots, it copies the disk label into memory. Use the `disklabel -r` command to access the label on the disk directly instead of going through the in-memory label.

Caution

Before you change disk partitions, back up all the file systems if there is any data on the disk. Changing a partition overwrites the data on the old file system, destroying the data.

When changing partitions, remember that:

- You cannot change the offset, which is the beginning sector, or shrink any partition on a mounted file system or on a file system that has an open file descriptor.
- If you need a single partition on the entire disk, use partition `c`.
- Unless it is mounted, you must specify the raw device for partition `a`, which begins at the start of the disk (sector 0), when you change the label. If partition `a` is mounted, use partition `c` to change the label. Partition `c` must begin at sector 0.

Caution

If partition `a` is mounted and you attempt to edit the disk label using device partition `a`, you cannot change the label. Furthermore, you do not receive any error messages indicating that the label is not written.

Before changing the size of a disk partition, review the current partition setup by viewing the disk label. The `disklabel` command allows you to view the partition sizes. The bottom, top, and size of the partitions are in 512-byte sectors.

To review the current disk partition setup, use the following `disklabel` command:

`/sbin/disklabel -r` *device*

Specify the device with its directory name (`/dev`) followed by the raw device name, drive number, and partition `a` or `c`. You can specify the disk unit and number, such as `dsk1`.

An example of using the `disklabel` command to view a disk label follows:

```
# disklabel –r /dev/rdisk/dsk3a
type: SCSI
disk: rz26
label:
flags:
bytes/sector: 512
sectors/track: 57
tracks/cylinder: 14
sectors/cylinder: 798
cylinders: 2570
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0           # milliseconds
track-to-track seek: 0  # milliseconds
drivedata: 0

8 partitions:
#       size offset   fstype [fsize bsize cpg]
 a:  131072      0   4.2BSD   1024  8192  16  # (Cyl.    0 - 164*)
 b:  262144  131072   unused   1024  8192      # (Cyl.  164*- 492*)
 c: 2050860       0   unused   1024  8192      # (Cyl.    0 - 2569)
 d:  552548  393216   unused   1024  8192      # (Cyl.  492*- 1185*)
 e:  552548  945764   unused   1024  8192      # (Cyl. 1185*- 1877*)
 f:  552548 1498312   unused   1024  8192      # (Cyl. 1877*- 2569*)
 g:  819200  393216   unused   1024  8192      # (Cyl.  492*- 1519*)
 h:  838444 1212416   4.2BSD   1024  8192 16   # (Cyl. 1519*- 2569*)
```

Take care when you change partitions because you can overwrite data on the file systems or make the system inefficient. If the partition label becomes corrupted while you are changing the partition sizes, you can return to the default partition label by using the `disklabel` command with the `-w` option, as follows:

```
# disklabel -r -w /dev/rdisk/dsk1a rz26
```

The `disklabel` command allows you to change the partition label of an individual disk without rebuilding the kernel and rebooting the system. Use the following procedure:

1. Display disk space information about the file systems by using the `df` command.
2. View the `/etc/fstab` file to determine if any file systems are designated as swap space.
3. Examine the disk's label by using the `disklabel` command with the `-r` option. (See `rz`(7), `ra`(7), and `disktab`(4) for information on the default disk partitions.)
4. Back up the file systems.

5. Unmount the file systems on the disk whose label you want to change.
6. Calculate the new partition parameters. You can increase or decrease the size of a partition. Also, you can cause partitions to overlap.
7. Edit the disk label by using the `disklabel` command with the `-e` option to change the partition parameters, as follows:

   ```
   # /sbin/disklabel -e disk
   ```

   An editor, either the `vi` editor or that specified by the EDITOR environment variable, is invoked so you can edit the disk label, which is in the format displayed with the `disklabel -r` command.

   The `-r` option writes the label directly to the disk and updates the system's in-memory copy, if possible. The *disk* parameter specifies the unmounted disk (for example, `dsk0` or `/dev/rdisk/dsk0a`).

   After you quit the editor and save the changes, the following prompt is displayed:

   ```
   write new label? [?]:
   ```

   Enter `y` to write the new label or `n` to discard the changes.
8. Use the `disklabel` command with the `-r` option to view the new disk label.

5.2.3 Examining for Overlapping Partitions with the newfs Command

Commands to mount or create file systems, add a new swap device, and add disks to the Logical Storage Manager first verify whether the disk partition specified in the command already contains valid data, and whether it overlaps with a partition that is marked for use already. The `fstype` field of the disk label enables you to determine when a partition or an overlapping partition is in use.

If the partition is not in use, the command continues to execute. In addition to mounting or creating file systems, commands such as `mount`, `newfs`, `fsck`, `voldisk`, `mkfdmn`, `rmfdmn`, and `swapon` also modify the disk label, so that the `fstype` field specifies partition usage. For example, when you add a disk partition to an AdvFS domain, the `fstype` field is set to `AdvFS`.

If the partition is not available, these commands return an error message and ask if you want to continue, as shown in the following example:

```
# newfs /dev/disk/dsk8c
WARNING: disklabel reports that basename,partition currently
is being used as "4.2BSD" data. Do you want to
continue with the operation and possibly destroy
existing data? (y/n) [n]
```

Applications, as well as operating system commands, can modify the `fstype` of the disk label, to indicate that a partition is in use. See `check_usage`(3) and `set_usage`(3) for more information.

5.3 Copying Disks

You can use the `dd` command to copy a complete disk or a disk partition; that is, you can produce a physical copy of the data on the disk or disk partition.

Note

Because the `dd` command is not meant for copying multiple files, copy a disk or a partition only to a disk that you are using as a data disk, or to a disk that does not contain a file system. Use the `dump` and `restore` commands, as described in Chapter 9, to copy disks or partitions that contain a UFS file system. Use the `vdump` and `vrestore` commands, as described in the *AdvFS Administration* manual, to copy disks or partitions that contain an AdvFS fileset.

The operating system protects the first block of a disk with a valid disk label because this is where the disk label is stored. As a result, if you copy a partition to a partition on a target disk that contains a valid disk label, you must decide whether you want to keep the existing disk label on that target disk.

If you want to maintain the disk label on the target disk, use the `dd` command with the `skip` and `seek` options to move past the protected disk label area on the target disk. The target disk must be the same size as or larger than the original disk.

To determine if the target disk has a label, use the following `disklabel` command:

```
# /sbin/disklabel -r target_disk
```

You must specify the target device directory name (`/dev`) followed by the raw device name, drive number, and partition `c`. If the disk does not contain a label, the following message is displayed:

```
Bad pack magic number (label is damaged, or pack is unlabeled)
```

The following example shows a disk that already contains a label:

```
# disklabel -r /dev/rdisk/dsk1c
type: SCSI
disk: rz26
label:
flags:
```

```
bytes/sector: 512
sectors/track: 57
tracks/cylinder: 14
sectors/cylinder: 798
cylinders: 2570
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0           # milliseconds
track-to-track seek: 0  # milliseconds
drivedata: 0

8 partitions:
#      size  offset  fstype [fsize bsize  cpg]
 a:  131072       0  unused 1024 8192 # (Cyl.    0 - 164*)
 b:  262144  131072  unused 1024 8192 # (Cyl.  164*- 492*)
 c: 2050860       0  unused 1024 8192 # (Cyl.    0 - 2569)
 d:  552548  393216  unused 1024 8192 # (Cyl.  492*- 1185*)
 e:  552548  945764  unused 1024 8192 # (Cyl. 1185*- 1877*)
 f:  552548 1498312  unused 1024 8192 # (Cyl. 1877*- 2569*)
 g:  819200  393216  unused 1024 8192 # (Cyl.  492*- 1519*)
 h:  838444 1212416  unused 1024 8192 # (Cyl. 1519*- 2569*)
```

If the target disk already contains a label and you do not want to keep the label, you must clear the label by using the `disklabel -z` command. For example:

```
# disklabel -z /dev/rdisk/dsk1c
```

To copy the original disk to the target disk and keep the target disk label, use the `dd` command, specifying the device directory name (`/dev`) followed by the raw device name, drive number, and the original and target disk partitions. For example:

```
# dd if=/dev/rdisk/dsk0c of=/dev/rdisk/dsk1c \
skip=16 seek=16 bs=512k
```

6

Administering UNIX File Systems (UFS)

This chapter introduces file systems and the basic system administration tasks related to file systems. Several file systems are supported, but the Advanced File System (AdvFS) and UNIX File System (UFS) are the principal file systems used by applications and the components of the UNIX operating system. If your system was delivered with the operating system already installed, the AdvFS is configured as the default file system. See the *AdvFS Administration* manual for information on administering AdvFS.

This chapter discusses system administration tasks related to the following file system topics:

- An introduction to the available file systems (Section 6.1)
- A discussion on (Context-Dependent Symbolic Links) (CDSLs), which facilitate the joining of systems into clusters (Section 6.2)
- A discussion on creating UFS file systems manually (Section 6.3)
- A description of how to create UFS file systems using the SysMan Menu tasks (Section 6.4)
- A discussion on how to control UFS file system resources by assigning quotas to users (Section 6.5)
- Pointers to methods of backing up UFS file systems (Section 6.6)
- A description of the features for monitoring and tuning file systems (Section 6.7)
- Information for troubleshooting UFS file system problems (Section 6.8)

There are several other sources of information about system administration tasks and file systems. This chapter directs you to those sources when appropriate.

6.1 Introduction to File Systems

If you installed the operating system yourself, you may have chosen to create one or more UFS file systems. Even if your system arrived configured for AdvFS, you still can create UFS file systems. Both file systems can coexist on a system and many administrators choose to use the familiar UFS file system on system disks or in instances where the advanced features of AdvFS are not required.

The operating system supports current versions of several file systems, including:

- Advanced File System (AdvFS). This file system has its own documentation and advanced interfaces. See the *AdvFS Administration* manual and `advfs`(4) for more information. There are advanced administrative utilities available for AdvFS. When these utilities are available, there is a launch icon named Advanced File System in the CDE Application Manager – Storage_Management folder. See the *AdvFS Administration* manual for information on setting up and using the advanced administrative utilities.

  Basic AdvFS utilities are provided as SysMan Menu tasks. See Chapter 1 for information on accessing these tasks. There is online help for the utilities provided by SysMan.
- UNIX File System (UFS), documented in this chapter. See `ufs_fsck`(8), `sys_attrs_ufs`(5), and `tunefs`(8) for information on attributes and utilities.
- ISO 9660 Compact Disk File System (CDFS). See `cdfs`(4) for more information.
- Memory File System (mfs). See `newfs`(8) for more information.
- File on File Mounting file system (ffm). See `ffm`(4) for more information.

You may need to see the following volumes:

- The *Logical Storage Manager* manual for information about using the Logical Storage Manager (LSM) with both the AdvFS and UFS file systems.
- The *AdvFS Administration* manual for information on converting file systems from UFS to AdvFS, and from AdvFS to UFS.
- The *System Configuration and Tuning* manual for information on advanced UFS file system tuning.

The rest of this section, and the following sections, introduce concepts that are important in the context of creating and administering file systems. The information is not essential for basic file system creation and administration, but it may be useful if you plan to perform advanced operations or perform troubleshooting tasks.

The following list provides a brief overview of the topics, with detailed information in the sections that follow:

Directory Hierarchy

Any file system, whether local or remotely mounted, is part of the total directory hierarchy of a system or cluster. It can be considered as a tree, growing from the root file system (/) and branching as additional directories are added to the basic system hierarchy. When you create a

UFS file system, such as /usr/usrs/projects, you add it as a new branch on the hierarchy, under the existing /usr/usrs branch.

Disk Partitions

The common form of file system storage on all systems is the hard disk. The administration of such devices is described in *Hardware Management* manual. A disk is divided into logical partitions, which may be the whole disk (partition c) or parts of the disk, such as partitions a through h. Depending on the size of the disk, the partitions vary in size, and are usually expressed in megabytes (MB). When you initially create a file system, you create it on a disk partition and thus assign a finite amount of size (disk space) to that file system. Increasing the size of a UFS file system may involve moving it to a bigger partition or disk.

File System Structures

A file system has an on-disk data structure that describes the layout of data on the physical media. You may need to know this structure to troubleshoot the file system or perform advanced operations such as tuning. For most common operations, you do not need to know this information in detail. Reference information is provided in the following sections.

Directories and File Types

The various directory and file types are displayed in the output of common commands that you use. Reference information is provided so that you can identify file types such as symbolic links or sockets. For more detailed information, see the appropriate reference page as follows:

Regular files	file(1)
Directories	ls(1) and dir(4)
Device Special Files	The *Hardware Management* manual
Sockets	socket(2), the *Network Administration: Connections* manual, and the *Network Programmer's Guide*
Pipes	pipe(2)
Symbolic links	link(1) and ln(1)

6.1.1 Directory Hierarchy for File Systems

The location of file systems is based on the UNIX directory hierarchy, beginning with a root (/) directory. The file systems that you create become usable (or active) when they are mounted on a mount point in the directory hierarchy. For example, during installation of the operating system, you may have created the `usr` file system (as UFS), which is then automatically mounted on root (/) and has a pathname of `/usr` in the hierarchy.

The standard system directory hierarchy is set up for efficient organization. It separates files by function and intended use. Effective use of file systems includes placing command files in directories that are in the normal search path as specified by a user's setup file, such as `.cshrc`, `.profile`, or `.login`. Some of the directories are actually symbolic links. See `hier`(5) for more information about the operating system's directory hierarchy, including the hierarchy of the X11 Windows System.

Mounting a file system makes it available for use. Use the `mount` command to attach (or mount) file systems to the file system hierarchy under the system root directory; use the `umount` command to detach (or unmount) them. When you mount a file system, you specify a location (the mount point under the system root directory) to which the file system attaches. See `mount`(8) for more information about mounting and unmounting file systems.

The root directory of a mounted file system is also its mount point. Only one system root directory can exist on a system, because it uses the root directory as its source for system initialization files. Consequently, all file systems that are local to an operating system are mounted under that system's root directory.

6.1.2 Disk Partitions

A disk consists of physical storage units called sectors. Each sector is usually 512 bytes. A sector is addressed by the logical block number (LBN), which is the basic unit of the disk's user-accessible data area that you can address. The first LBN is numbered 0, and the highest LBN is numbered one less than the number of LBNs in the user-accessible area of the disk.

Sectors are grouped together to form up to eight disk partitions. However, disks differ in the number and size of partitions. The `/etc/disktab` file contains a list of supported disks and the default partition sizes for the system. See `disktab`(4) for more information.

Disk partitions are logical divisions of a disk that allow you to organize files by putting them into separate areas of varying sizes. Partitions hold data in structures called file systems and can be used for system operations such as paging and swapping. File systems have a hierarchical structure of directories and files, as shown in `hier`(5).

Disk partitions have default sizes that depend on the type of disk and that can be altered by using the `disklabel` command or the `diskconfig` graphical user interface. Partitions are named `a` to `h`. While it is possible for you to make the allocated space for a partition overlap another partition, the default partitions are never overlapping, and a properly used disk must not have file systems on overlapping partitions.

Example 6–1 shows the default partitioning for a model RZ1DF-CB disk, using the following command:

```
# disklabel -r /dev/rdisk/dsk0a
```

Only the disk table part of the output is shown here. Also listed is an example of an RZ1DF-CB Disk and HSZ RAID disk, taken from `rz`(7).

Example 6–1: Default Partitions

```
(RZ1DF-CB Disk)

8 partitions:
#          size    offset    fstype  [fsize bsize   cpg] # NOTE: values not exact
  a:     262144         0    4.2BSD    1024  8192    16       # (Cyl.    0 - 78*)
  b:    1048576    262144      swap                          # (Cyl.   78*- 390*)
  c:   17773524         0    unused       0     0            # (Cyl.    0 - 5289*)
  d:    1048576   1310720      swap                          # (Cyl.  390*- 702*)
  e:    9664482   2359296     AdvFS                          # (Cyl.  702*- 3578*)
  f:    5749746  12023778    unused       0     0            # (Cyl. 3578*- 5289*)
  g:    1433600    524288    unused       0     0            # (Cyl.  156*- 582*)
  h:   15815636   1957888    unused       0     0            # (Cyl.  582*- 5289*)

HSZ10, HSZ40, HSZ50, HSZ70  (RAID) Partitions

       Disk      Start     Length

       dsk?a     0         131072
       dsk?b     131072    262144
       dsk?c     0         end of media
       dsk?d     0         0
       dsk?e     0         0
       dsk?f     0         0
       dsk?g     393216  end of media
       dsk?h     0         0
```

The disk label is located in block 0 (zero) in one of the first sectors of the disk. The disk label provides detailed information about the geometry of the disk and the partitions into which the disk is divided. The system disk driver and the boot program use the disk label information to recognize the drive, the disk partitions, and the file systems. Other information is used by the operating system to use the disk most efficiently and to locate important file system information.

The disk label description of each partition contains an identifier for the partition type (for example, standard file system, swap space, and so on). There are two copies of a disk label, one located on the disk and one located

in system memory. Because it is faster to access system memory than to perform I/O, when a system recognizes a disk, it copies the disk label into memory. The file system updates the in-memory copy of the label if it contains incomplete information about the file system. You can change the label with the `disklabel` command. See `disklabel`(8) for more information on the command line interface. See the *Hardware Management* manual for information on the disk configuration utility `diskconfig`.

6.1.3 UFS Version 4.0

The version of UFS that is currently provided is at revision 4.0. This version has the same on-disk data layout as UFS Version 3.0, as described in Section 6.1.4 but has larger capacities.

Version 4.0 supports 65533 hard links or subdirectories while Version 3.0 supports 32767 hard links or subdirectories. The actual number of directories is 65531 (64k) and 32765 (32k), because the empty directory already has two hard links to itself and to its parent directory. When you use the `ls -a` command, these links are displayed as `.` and `...` In the remainder of this section, the examples all refer to 32k subdirectories although the information also applies to files having 32k or more hard links.

There are some considerations and important restrictions that you should take into account, particularly when using both versions, as follows:

Using the `newfs` or `diskconfig` command to create file systems

When you create new file systems using `newfs` or `diskconfig` command, the new file systems always are created as Version 3.0 (32k subdirectories or hard links) to minimize any incompatibility problems.

Using the `fsck` command to verify file systems

When you use `fsck` to verify a dirty file system (such as one not unmounted normally, or perhaps after a system crash), the file system is marked as either Version 3.0 or Version 4.0, depending on the maximum number of subdirectories found. If the `fsck` command finds a directory with more than 32k subdirectories, the file system is marked as Version 4.0. Otherwise, if the `fsck` command does not find a directory with more than 32k hard links, the file system is marked as Version 3.0. A file system normally is converted to Version 4.0 as soon as the 32k subdirectory limit is exceeded by a user.

A new `fsck` option, `-B`, is added. This option enables you to convert Version 4.0 file systems back to Version 3.0. When you use this option, `fsck` make the conversion only if no directory in the file system has more than 32k subdirectories and no file has more than 32k hard links.

The following important restrictions apply when using both Version 3.0 and Version 4.0 of UFS on systems that are running previous versions of the operating system (such as V4.0F):

- Do not run previous versions of `fsck` using the `-p` or `-y` options on a Version 4.0 file system unless you are certain that there are no directories that have more than 32k subdirectories. If you attempt to do this, any directories that have more than 32k subdirectories are deleted permanently from the file system.
- Do not list directories with more than 32k subdirectories in the root (`/`) and `/usr` partitions (or other UFS partitions) in the `/etc/fstab` file. At boot time `fsck -p` runs automatically on all file systems listed in `/etc/fstab`.

  As a protection against this, Version 4.0 creates a mismatch between the main superblock and alternate superblocks so that old versions of `fsck -p` cannot be run on a Version 4.0 file system. The first time you attempt to run the old version of `fsck -p` on a Version 4.0 file system that has more than 32k subdirectories, it fails because of a superblock mismatch with alternate superblocks. When you are prompted to specify an alternate superblock, always respond `n`. Even if you enter `y` in error, the Version 4.0 file system remains untouched, providing you do not enter `y` when the following prompt is displayed:

  ```
  CLEAR? [yn]
  ```

  At this time, you can correct the `FREE BLK COUNT` and the `UPDATE STANDARD SUPERBLOCK` if required. However, the second time you run `fsck -p` on a Version 4.0 file system, this mismatch protection does not exist. Any directories with more than 32k subdirectories are deleted permanently.

Automatic conversion from Version 3.0 to Version 4.0

As there are no on-disk data layout differences between the two releases of UFS, you can mount any legacy Version 3.0 file systems on the latest release of the operating system. If you attempt to create more than 32k hard links on a Version 3.0 file system, it is converted automatically to Version 4.0. The following example system message is displayed during conversion:

```
Marking /dev/disk/dsk023 as Tru64 UNIX UFS v.4
```

Manually converting file systems from Version 3.0 to Version 4.0

If you want to share or mount a Version 4.0 file system that does not have more than 32k subdirectories, you can mount it on a system that is running a previous version of the operating system that supports

only Version 3.0, such as Tru64 UNIX Version 4.0F. However, you must first convert the file system from Version 4.0 as follows:

- On the system that supports Version 3.0, use the `fsck` command on the file system partition, as shown in the following example:

  # **fsck /dev/rrz03**

- On the system that supports Version 4.0, use the `fsck` command on the file system partition, as shown in the following example:

  # **fsck -B /dev/disk/dsk34d**

6.1.4 File System Structures: UFS

A UFS file system has four major parts:

- Boot block
- Superblock
- Inode blocks
- Data blocks

These are described in the following sections.

The structure of the AdvFS file system is discussed in the *AdvFS Administration* manual.

6.1.4.1 Boot Block

The first block of every file system (block 0) is reserved for a boot, or initialization program.

6.1.4.2 Superblock

Block 1 of every file system is called the superblock and contains the following information:

- Total size of the file system (in blocks)
- Number of blocks reserved for inodes
- Name of the file system
- Device identification
- Date of the last superblock update
- Head of the free-block list, which contains all the free blocks (the blocks available for allocation) in the file system

When new blocks are allocated to a file, they are obtained from the free-block list. When a file is deleted, its blocks are returned to the free-block list.

- List of free inodes, which is the partial listing of inodes available to be allocated to newly created files

6.1.4.3 Inode Blocks

A group of blocks follows the superblock. Each of these blocks contains a number of inodes. Each inode has an associated inumber. An inode describes an individual file in the file system. There is one inode for each file in the file system. File systems are subject to a limit on the number of inodes, which in turn controls the maximum number of files that a file system can contain. The maximum number of inodes depends on the size of the file system.

The first inode (inode 1) on each file system is unnamed and unused. The second inode (inode 2) must correspond to the root directory for the file system. All other files in the file system are under the file system's root directory. After inode 2, you can assign any inode to any file. You can also assign any data block to any file. The inodes and blocks are not allocated in any particular order.

If an inode is assigned to a file, the inode can contain the following information:

- File type

  The possible types are regular, device, named pipes, socket, and symbolic link files.

- File owner

  The inode contains the user and group identification numbers that are associated with the owner of the file.

- Protection information

  Protection information specifies read, write, and execute access for the file owner, members of the group associated with the file, and others. The protection information also includes other mode information specified by the `chmod` command.

- Link count

  A directory entry (link) consists of a name and the inumber (inode number) that represents the file. The link count indicates the number of directory entries that refer to the file. A file is deleted if the link count is zero; the file's inode is returned to the list of free inodes, and its associated data blocks are returned to the free-block list.

- Size of the file in bytes

- Last file access date
- Last file modification date
- Last inode modification date
- Pointers to data blocks

  These pointers indicate the actual location of the data blocks on the physical disk.

6.1.4.4 Data Blocks

Data blocks contain user data or system files.

6.1.5 Directories and File Types

The operating system views files as bit streams, allowing you to define and handle on-disk data, named pipes, UNIX domain sockets, and terminals as files. This object-type transparency provides a simple mechanism for defining and working with a wide variety of storage and communication facilities. The operating system handles the various levels of abstraction as it organizes and manages its internal activities.

While you notice only the external interface, you should understand the various file types recognized by the system. The system supports the following file types:

- Regular files contain data in the form of a program, a text file, or source code, for example.
- Directories are a type of regular file and contain the names of files or other directories.
- Character and block device special files identify physical and pseudodevices on the system.
- UNIX domain socket files provide a connection between network processes. The `socket` system call creates socket files.
- Named pipes are device files. Processes use named pipes to communicate with each other.
- Linked files point to target files or directories. A linked file contains the name of the target file. A symbolically linked file and its target file can be located on the same file system or on different file systems. A file with a hard link and its target file must be located on the same file system.

6.1.6 Device Special Files

Device special files represent physical devices, pseudodevices, and named pipes. The `/dev` directory contains device special files. Device special files

serve as the link between the system and the device drivers. Each device special file corresponds to a physical device (for example, a disk, tape, printer, or terminal) or a pseudodevice (for example, a network interface, a named pipe, or a UNIX domain socket). The driver handles all read and write operations and follows the required protocols for the device.

There are three types of device special files:

- Block device special files

  Block device special files are used for devices whose driver handles I/O in large blocks and where the kernel handles I/O buffering. Physical devices such as disks are defined as block device files. An example of the block device special files in the `/dev` directory follows:

  ```
  brw------- 1 root system 8, 1 Jan 19 11:20 /dev/disk/dsk0a
  brw------- 1 root system 8, 1 Jan 19 10:09 /dev/disk/dsk0b
  ```

- Character device special files

  Character device special files are used for devices whose drivers handle their own I/O buffering. Disk, terminal, pseudoterminal, and tape drivers typically are defined as character device files. An example of the character device special files in the `/dev` directory follows:

  ```
  crw-rw-rw- 1 root system 7,   0 Jan 31 16:02 /dev/ptyp0
  crw-rw-rw- 1 root system 7,   1 Jan 31 16:00 /dev/ptyp1
  crw-rw-rw- 1 root system 9,1026 Jan 11 14:20 /dev/rtape/tap_01
  ```

  Another case of a character device special file is the raw disk device, for example:

  ```
  crw-rw-rw- 1 root system 7,   0 Jan 10 11:19 /dev/rdisk/dsk0a
  ```

- Socket device files

  The printer daemon (`lpd`) and error logging daemon (`syslogd`) use the socket device files. An example of the socket device files in the `/dev` directory follows:

  ```
  srw-rw-rw- 1 root  system     0 Jan 22 03:40 /dev/log
  srwxrwxrwx 1 root  system     0 Jan 22 03:41 /dev/printer
  ```

For detailed information on device special files and their naming conventions, see the *Hardware Management* manual.

6.2 Context-Dependent Symbolic Links and Clusters

This section describes context-dependent symbolic links (CDSLs), a feature of the directory hierarchy that supports joining systems into clusters. CDSLs impose certain requirements on the file system and directory hierarchy of all systems, even those that are not members of a cluster. You should be aware of these requirements as follows:

- The root (/), /var, and /usr file systems each have a /cluster subdirectory that is not used on a single system, but must not be deleted or the system cannot be added into a cluster at some future time.
- When systems are joined into clusters, they are designated as members of the cluster. There is a unique pathname to any file, including an identifier that is unique to the member system (member-specific). These pathnames are called context-dependent symbolic links (CDSLs). As the name implies, CDSLs are symbolic links with a variable element in the pathname. The variable element is different for each cluster member and provides the context when it is resolved by an application or command.
- Some important system files reside in target directories that have unique CDSLs pointing to the target location. This design ensures that shared (cluster-wide) files are kept separate from unshared (member-specific) files.
- Update installations may fail if CDSLs are moved or destroyed.

See hier(5) for a description of the directory structure.

CDSLs enable systems joined together as members of a cluster to have a global namespace for all files and directories they need to share. CDSLs allow base components and layered applications to be cluster aware. Shared files and directories work equally well on a cluster and a single system and file system administration tools work identically both on a single system and in a cluster.

If CDSLs are important to you because your systems may become cluster members at some future date, you should read the following sections. If you encounter errors that refer to missing CDSLs (such as a failed update installation) you may need to maintain, verify, or repair CDSLs as described in the following sections.

6.2.1 Related Documentation

The following documents contain information about CDSLs:

- The *Installation Guide* contains information about update installations. See installupdate(8) for information on the update installation process.

  The TruCluster documentation describes the process of adding a system to a cluster and further explains how CDSLs are used on a running cluster; this documentation is not part of the base documentation set.
- The local(4), ls(1), ln(1), and hier(5) reference pages provide references and information on commands.

The cdslinvchk(8) reference page contains a discussion of the /usr/sbin/cdslinvchk script that you use to produce an inventory of all CDSLs on a single system when the system is installed or updated.

6.2.2 Description of CDSLs

Individual systems can be connected into clusters that appear as one system to users. A single system in a cluster is called a member. (See the TruCluster documentation for a description of a Tru64 UNIX cluster.) To facilitate clustering, file systems must have a structure and identifying pathname that allows certain files to be unique to the individual cluster member and contain member-specific information.

Other files may need to be shared by all members of a cluster. The CDSL pathname allows the different systems in a cluster to share the same file hierarchy. Users and applications can use traditional pathnames to access files and directories whether they are shared or member-specific.

For example, if two systems are standalone or simply connected by a network link, each has an /etc/passwd file that contains information about its authorized users. When two systems are members of a cluster, they share a common /etc/passwd file that contains information about the authorized users for both systems.

Other shared files are:

- Any configuration files and directories that are site-specific rather than system-specific, such as /etc/timezone or /etc/group
- Files and directories that contain no customized information, such as /bin or /usr/bin
- Any device special files for disk and tape devices that are available cluster-wide

Some files always must be member-specific; that is, not shared. The file/etc/rc.config is an example of a member-specific file while rc.config.common is a shared file. These files contain configuration information that either applies only to the individual system or to all members of a cluster. CDSLs allow clustered systems to share files and to maintain the identity of member-specific files. Other categories of member-specific files are:

- Certain directories, such as /var/adm/crash. These directories contain files that are created by applications, utilities, or daemons that only apply to the individual cluster member.
- Some device special files located in /dev and /devices.
- Configuration files that reference member-specific device special files, such as /etc/securettys.

- Processor-specific files used during booting or configuration such as /vmunix and /etc/sysconfigtab.

When a system is not connected to a cluster the pathnames are still present, although they are transparent to users. You must be aware of the cluster file naming conventions, and must preserve the file structure. If a CDSL is accidentally removed, you may need to recreate it.

6.2.2.1 Structure of a CDSL

CDSLs are simply the symbolic links described in ln(1). The links contain a variable that identifies each system that is a cluster member. This variable is resolved at run time into a target. A CDSL is structured as follows:

```
/etc/rc.config -> /cluster/members/{memb}/etc/rc.config
```

Before support for clusters was introduced, the pathname for this file was /etc/rc.config. This file is now linked through a CDSL to a member-specific target, and the structure of the link can be interpreted as follows:

- The /cluster directory resides in the root directory and contains paths to the files that are either shared or (as in this example) member-specific.
- The /cluster/members/directory contains a directory for the local member identifier, member0, and a link to the variable path element {memb}. The directory /cluster/members/member0 contains member-specific system directories such as devices and etc.
- The {memb} variable path element is used to identify individual members of a cluster. At run time, this variable is resolved to be member, appended with the value of the sysconfigtab variable generic:memberid. The default value for this variable is zero, and the value is unique for each member of a cluster.

The file /.local.. in root is a link to cluster/members/{memb} and defines the system-specific files. Any system-specific file can be referenced or created through the /.local.. path. A file created as /.local../etc/[filename] is not accessible through the path /etc/[filename] because /etc is a shared directory. The file is only accessible through /.local../etc/[filename] and /cluster/members/{memb}/etc/[filename].

When a single system is not clustered with other systems, the variable generic:memberid is set to zero automatically. An example of a typical CDSL on a single system is:

/cluster/members/{memb}/etc/rc.config

This CDSL is resolved to:

/cluster/members/member0/etc/rc.config

When a system is clustered with two other systems and the variable `generic:memberid` is set to three, the same CDSL is resolved to:

/cluster/members/member3/etc/rc.config

When running in a cluster, a file that is member-specific can be referenced in the following three ways:

- From your specific system in a member-specific or shared format, for example: `/var/adm/crash/crash-data.5`
- From your specific system in a member-specific format only, for example: `/.local../var/adm/crash/crash-data.5`
- From any member of the cluster, for example: `/cluster/members/member0/var/adm/crash/crash-data.5`

Two special cases of CDSLs exist only for members of a cluster:

- Miniroot
- Special Unshared Directories:
  - `/dev -> /cluster/members/{memb}/dev`
  - `/tmp -> /cluster/members/{memb}/tmp`

See the TruCluster documentation for more information.

6.2.3 Maintaining CDSLs

Symbolically-linked files enjoy no special protection beyond the general user and file access mode protections afforded all files. CDSLs have no special protection either. On a single system, there are several situations that could cause it to fail when a CDSL has been broken:

- Whenever an update installation to the operating system is performed.

  On a system that is not in a cluster, you become aware of missing CDSLs only when you attempt to update the operating system using the update installation process, `installupdate`(8) and it fails. To prevent this problem, always run the `/usr/sbin/cdslinvchk` script before an update installation in order to obtain its report on the state of CDSLs on your system.

- When a user or application moves or removes a member-specific CDSL.

  Member-specific CDSLs can be removed accidentally with the `rm` or `mv` commands. To prevent this problem, avoid manual edits and file creations and use tools such as `vipw` (for editing `/etc/passwd`) to edit files. All system administration tools and utilities are aware of CDSLs and should be the preferred method for managing system files.

6.2.3.1 Verifying CDSL Inventory

Use the script `/usr/sbin/cdslinvchk` to verifying the CDSL inventory on a single system. Periodically, revise the inventory and examine the CDSLs against it. See `cdslinvchk`(8) for more information.

6.2.3.2 Creating CDSLs

If a CDSL is accidentally destroyed, or if a new CDSL must be created, the process for repairing or creating links is described in `ln`(1). For example, if the `/etc/rc.config` link is destroyed, you create it as follows:

- Verify the value of `{memb}`, as defined by the `sysconfigtab` variable `generic:memberid`
- Verify that the file exists, for example:

  ```
  # ls /cluster/members/members3/etc/rc.config
  ```

- For a `generic:memberid` of 3 , create a new link as follows:

  ```
  # cd /etc
  # ln -s /cluster/members/member3/rc.config
  ```

6.3 Creating UFS File Systems Manually

The basic file system configuration for your operating system is defined during installation, when your system's root file system is established. After installation, you can create file systems as your needs evolve. The following sections describe how you create file systems manually, at the command line. You must use command line operations on file systems when working at the console, when the system is in single-user mode and graphic utilities are unavailable.

For information on creating AdvFS file systems, see the *AdvFS Administration* manual.

6.3.1 Using newfs to Create a New File System

The typical procedure for creating a file system is as follows:

1. Identify the disk device and the raw disk partition that you want to use for the new partition, ensuring that the partition is correctly labeled and formatted and is not in use already. Use the command line interfaces `hwmgr` and `dsfmgr` to identify devices or to add new devices and create the device special files. This procedure is described in the *Hardware Management* manual. See `hwmgr`(8) and `dsfmgr`(8) for information on the command options.

   If required, use the `disklabel -p` command to read the current partition status of the disks. Examine the `/etc/fstab` file to ensure

that the partitions are not already allocated to file systems, or used as swap devices. See disklabel(8), and fstab(4) for more information.

2. Having identified which unused raw (character) disk partition you use, you can determine the special device file name for the partition. For example, partition g on disk 2 has a special device file named /dev/rdisk/dsk2g. See the *Hardware Management* manual for information on device special file names.
3. Use the newfs command to create a file system on the target partition. See newfs(8) for more information.
4. Create a mount point directory, and use the mount command to mount the new file system, making it available for use. If you want the mount to persist across reboots, add a mount command to the /etc/fstab file. If you want to export the file system, add it to the /etc/exports file; see mount(8) for more information.
5. Use the chmod command to verify and adjust any access control restrictions; see chmod(1) for more information.

These steps are described in more detail in the remainder of this section.

The newfs command formats a disk partition and creates a UFS file system. Using the information in the disk label or the default values specified in the /etc/disktab file, the newfs command builds a file system on the specified disk partition. You can use newfs command options to specify the disk geometry.

Note

Changing the default disk geometry values may make it impossible for the fsck program to find the alternate superblocks if the standard superblock is lost.

The newfs command has the following syntax:

/sbin/newfs [–N] [*newfs_options*] *special_device* [*disk_type*]

You must specify the unmounted, raw device (for example, /dev/rdisk/dsk0a).

See newfs(8) for information on the command options specific to file systems. This reference page also provides information on the mfs command, and describes how you create a memory file system (mfs).

The following example shows the creation of a new file system:

1. Determine the target disk and partition. For most systems, your local administrative log book tells you which disk devices are attached to a

system and which partitions are assigned. However, you may be faced with administering a system that could be in an unknown state; that is, devices may have been removed or added. Use the following commands and utilities to assist you in identifying a target disk and partition:

a. Examine the contents of the `/dev/disk` directory. Each known disk device has a set of device special files for the partition layout. For example, `/dev/disk/dsk1a` to `/dev/disk/dsk1h` tells you that there is a device named `dsk1`.

b. Devices may be available on the system, but without any device special files. Use the `hwmgr` command to examine all devices that are known to the system physically and visible on a bus. For example:

```
# hwmgr -view devices -category disk
 HWID:            DSF Name    Model       Location
-------------------------------------------------------
   15:  /dev/disk/floppy0c    3.5in         fdi0-unit-0
   17:     /dev/disk/dsk0c RZ1DF-CB  bus-0-targ-0-lun-0
   19:                     RZ1DF-CB  bus-0-targ-1-lun-0
   19:   /dev/disk/cdrom0c    RRD47  bus-0-targ-4-lun-0
```

If a device is found for which no device special files exist, you can create the device special files using the `dsfmgr` utility.

Note

Normally, device special files are created automatically when a new disk device is added to the system. You only need to create them manually under the circumstances described in the *Hardware Management* manual.

c. Having identified a device, use the `disklabel` command to determine what partitions may be in use as follows:

```
# disklabel -r /dev/rdisk/dsk0a
```

```
8 partitions:
#       size  offset fstype [fsize bsize cpg] #NOTE: values not
                                                    exact
 a:   262144       0 4.2BSD 1024 8192  16  #(Cyl.   0 -78*)
 b:  1048576  262144   swap                 #(Cyl.  78*-390*)
 c: 17773524       0 unused    0    0      #(Cyl.   0 -5289*)
 d:  1048576 1310720   swap                 #(Cyl. 390*-702*)
 e:  9664482 2359296  AdvFS                 #(Cyl. 702*-3578*)
 f:  5749746 12023778 unused    0    0      #(Cyl.3578*-5289*)
 g:  1433600  524288 unused    0    0      #(Cyl. 156*-582*)
 h: 15815636 1957888 unused    0    0      #(Cyl. 582*-5289*)
```

2. From the `disklabel` command output, it appears that there are several unused partitions. However, you cannot use the `c` partition because it overlaps with the other partitions. Unless a custom disklabel has been

created on the disk, only three possible tables of standard partitions are available for use, as shown in Table 6–1.

Table 6–1: Disk Partition Tables

Partition Table	Description
c	The entire disk is labeled as a single partition. Therefore, other partitions overlap c so you cannot use it.
a b g h	The disk is divided into four partitions. Partition a can be used as a boot partition. Partitions c, d, e, and f overlap so you cannot use them.
a b d e f	The disk is divided into five partitions. Partition a can be used as a boot partition. Partitions c, g, and h overlap so you cannot use them.

The disk listed in the output from the `disklabel` command in step 1.c already uses partitions a, b, d, and e. Therefore it is labeled for five partitions, and the f partition is unused and available for use by the new file system.

Note

If a custom disk label has been applied to the disk and partitions are extended, you may not be able to use a partition even if it is designated as unused. In this case, the `newfs` command is not able to create the file system and returns an error message.

3. Use the `newfs` command to create a file system on the target partition, as follows:

```
# newfs /dev/rdisk/dsk0f

Warning: 2574 sector(s) in last cylinder unallocated
/dev/rdisk/dsk0f:  5749746 sectors in 1712 cylinders of \
20 tracks, 168 sectors
   2807.5MB in 107 cyl groups (16 c/g, 26.25MB/g, 6336 i/g)
super-block backups (for fsck -b #) at:
 32, 53968, 107904, 161840, 215776, 269712, 323648,
377584, 431520, 485456, 539392, 593328, 647264, 701200,
755136, 809072, 863008, 916944, 970880, 1024816, 1078752,
1132688, 1186624, 1240560,
.
.
.
```

The command output provides information on the size of the new file system and lists the super-block backups that are used by the file system checking utility `fsck`. See `fsck`(8) for more information.

4. Mount the file system as described in the following sections.

6.3.2 Making File Systems Accessible to Users

You attach a file system to the file system hierarchy using the `mount` command, which makes the file system available for use. The `mount` command attaches the file system to an existing directory, which becomes the mount point for the file system.

Note

The operating system does not support 4-Kb block-size file systems. The default block size for file systems is 8 kilobytes. To access the data on a disk that has 4-Kb block-size file systems, you must back up the disk to either a tape or a disk that has 8-Kb block-size file systems.

When you boot the system, file systems that are defined in the `/etc/fstab` file are mounted. The `/etc/fstab` file contains entries that specify the device and partition where the file system is located, the mount point, and more information about the file system, such as file system type. If you are in single-user mode, the root file system is mounted read only.

If you should encounter a "dirty file system" error message when you try to mount the file system, you need to run the `fsck` command on that file system.

To change a file system's mount status, use the `mount` command with the `-u` option. This is useful if you try to reboot and the `/etc/fstab` file is unavailable.

If you try to reboot and the `/etc/fstab` file is corrupted, use a command similar to the following:

```
# mount -u /dev/disk/dsk0a /
```

The `/dev/disk/dsk0a` device is the root file system.

6.3.3 Using the /etc/fstab File

Either AdvFS or UFS can be the root file system, although AdvFS is used by default if you do not specify UFS during installation. If your system was supplied with a factory-installed operating system, the root file system is AdvFS. The operating system supports only one root file system from

which it accesses the executable kernel (/vmunix) and other binaries and files that it needs to boot and initialize. The root file system is mounted at boot time and cannot be unmounted. Other file systems must be mounted, and the /etc/fstab file tells a booting system what file systems to mount and where to mount them.

The /etc/fstab file contains descriptive information about file systems and is read by commands such as the mount command. When you boot the system, the /etc/fstab file is read and the file systems described in the file are mounted in the order that they appear in the file. A file system is described on a single line; information on each line is separated by tabs or spaces.

The order of entries in the /etc/fstab file is important because the mount and umount commands read and act on the file entries in the order that they appear.

You must be root user to edit the /etc/fstab file. When you complete changes to the file and want to immediately apply the changes, use the mount -a command. Otherwise, any changes you make to the file become effective only when you reboot the system.

The following is an example of an /etc/fstab file:

```
/dev/disk/dsk2a   /                 ufs     rw      1    1
/dev/disk/dsk0g   /usr              ufs     rw      1    2
/dev/disk/dsk2g   /var              ufs     rw      1    2
/usr/man@tuscon   /usr/man          nfs     rw,bg   0    0
proj_dmn#testing  /projects/testing  advfs   rw      0    0
    1               2                3      4       5    6
```

Each line contains an entry and the information is separated either by tabs or spaces. An /etc/fstab file entry has the following information:

1 Specifies the block special device or remote file system to be mounted. For UFS, the special file name is the block special file name, not the character special file name. For AdvFS, the special file name is a combination of the name of the file domain, a number sign (#), and the fileset name.

2 Specifies the mount point for the file system or remote directory (for example, /usr/man) or /projects/testing.

3 Specifies the type of file system, as follows:

cdfs	Specifies an ISO 9600 or HS formatted (CD–ROM) file system.
nfs	Specifies NFS.
procfs	Specifies a /proc file system, which is used for debugging.

ufs	Specifies a UFS file system or a swap partition.
advfs	Specifies an AdvFS file system.

4 Describes the mount options associated with the partition. You can specify a list of options separated by commas. Usually, you specify the mount type and any additional options appropriate to the file system type, as follows:

ro	Specifies that the file system is mounted with read-only access.
rw	Specifies that the file system is mounted with read-write access.
userquota groupquota	Specifies that the file system is processed automatically by the quotacheck command and that file system quotas are enabled with the quotaon command. By default, user and group quotas for a file system are contained in the quota.user and quota.group files, which are located in the directory specified by the mount point. For example, the quotas for the file system on which /usr is mounted are located in the /usr directory. You also can specify another file name and location. For example: userquota=/var/quotas/tmp.user
xx	Specifies that the file system entry should be ignored.

5 Used by the dump command to determine which UFS file systems should be backed up. If you specify the value 1, the file system is backed up. If you do not specify a value or if you specify 0 (zero), the file system is not backed up.

6 This is the pass number and is used to control parallelism in the fsck (UFS) and quotacheck (UFS and AdvFS) utilities when processing all the entries in the /etc/fstab file. You can use this field to avoid saturating the system with too much I/O to the same I/O subsystem by controlling the sequence of file system verification during startup.

If you do not specify a pass number or if you specify 0 (zero), the file system is not verified. All entries with a pass number of 1 are processed one at a time (no parallelism). For the root file system, always specify 1. Entries with a pass number of 2 or greater are processed in parallel

based on the pass number assigned (with some exceptions). All entries with a pass number of 2 are processed before pass number 3, pass number 3 are processed before 4, and so on. The exceptions are multiple UFS file systems on separate partitions of the same disk or multiple AdvFS filesets in the same domain. These are processed one after the other if they all have the same pass number. All other file systems with the same pass number are processed in parallel.

See `fstab`(4) for more information about its fields and options.

Swap partitions are configured in the `/etc/sysconfigtab` file as shown in the following example:

```
swapdevice=/dev/disk/dsk0b,/dev/disk/dsk0d
vm-swap-eager=1
```

See the *Hardware Management* manual, Chapter 12, and `swapon`(8) for more information on swapping and swap partitions.

6.3.4 Mounting the UFS File System Manually

You use the `mount` command to make a file system available for use. Unless you add the file system to the `/etc/fstab` file, the mount is temporary and does not exist after you reboot the system.

The `mount` command supports the UFS, AdvFS, NFS, CDFS, and `/proc` file system types.

The following `mount` command syntax is for all file systems:

mount [- adflruv] [- o *option*] [- t *type*] [*file_system*] [*mount_point*]

For AdvFS, the file system argument has the following form:

domain#fileset

Specify the file system and the mount point, which is the directory on which you want to mount the file system. The directory already must exist on your system. If you are mounting a remote file system, use one of the following syntaxes to specify the file system:

```
host:remote_directory
remote_directory@host
```

The following command lists the currently mounted file systems and the file system options.

```
# mount -l
/dev/disk/dsk2a on / type ufs (rw,exec,suid,dev,nosync,noquota)
/dev/disk/dsk0g on /usr type ufs (rw,exec,suid,dev,nosync,noquota)
/dev/disk/dsk2g on /var type ufs (rw,exec,suid,dev,nosync,noquota)
/dev/disk/dsk3c on /usr/users type ufs (rw,exec,suid,dev,nosync,noquota)
/usr/share/man@tuscon on /usr/share/man type nfs (rw,exec,suid,dev,
nosync,noquota,hard,intr,ac,cto,noconn,wsize=8192,rsize=8192,
timeo=10,retrans=10,acregmin=3,acregmax=60,acdirmin=30,acdirmax=60)
```

```
proj_dmn#testing on /alpha_src type advfs (rw,exec,suid,dev,nosync,noquota)
```

The following command mounts the `/usr/homer` file system located on host `acton` on the local `/homer` mount point with read-write access:

```
# mount -t nfs -o rw acton:/usr/homer /homer
```

See `mount`(8) for more information on general options and options specific to a file system type.

6.3.5 Unmounting the UFS File System Manually

Use the `umount` command to unmount a file system. You must unmount a file system if you want to do the following:

- Verify a file system by using the `fsck` command.
- Change partitions by using the `disklabel` command. (Take care with this operation. Changing partitions can destroy existing file systems on the disk.)

The `umount` command has the following syntax:

umount [- afv] [- h *host*] [- t *type*] [*mount_point*]

If any user process (including a `cd` command) is in effect within the file system, you cannot unmount the file system. If the file system is in use when the command is invoked, the system returns the following error message and does not unmount the file system:

```
mount device busy
```

You cannot unmount the root file system with the `umount` command.

6.3.6 Extending the UFS File System

You can increase the capacity of a UFS file system up to the storage available on a single disk or Logical Storage Manager (LSM) volume. The process of increasing the capacity (or size) of a UFS file system is called extending the file system.

When the file system is on line (mounted) you can extend it by using `mount` command options. If preferred, you can perform this operation when the file system is off line (dismounted) by using the `extendfs` command. You can use this procedure as either a temporary or permanent solution if the system notifies you that a file system is full. File systems can be extended as frequently as required, up to the physical limits of the storage device.

You cannot reverse this procedure. The only way you can return a file system to its original volume is to back up the file system using the `dump` or a back up utility, and then restore the file system to an appropriately sized disk partition.

The prerequisites for extending a file system are as follows:

Identify file systems

Use the `more` command on the `/etc/fstab` file to identify file systems and the partitions on which they reside as follows:

```
# /usr/bin/more /etc/fstab
/dev/disk/dsk0a on / type ufs (rw)
/proc on /proc type procfs (rw)
.
.
.
/dev/disk/dsk15e on /databases type ufs (rw)
/dev/disk/dsk4g on /projects type ufs (rw)
```

Verify the file system back up status

This procedure is nondestructive, and designed so that you can perform it quickly when needed. However, you may want to back up important data files.

Determine the available disk storage capacity

A UFS file system exists on a single disk partition or LSM volume. To extend the file system, increase the disk space as follows:

- If LSM is not in use, increase the size of the disk partition by decreasing the size of an unused adjacent partition.
- If LSM is in use, use LSM commands to extend the volume. See the *Logical Storage Manager* manual for information. The examples in this section refer to UFS file systems where LSM is not in use.

For example, you have a disk that is currently used as follows:

- The `a` partition is in use as a 500 MB tertiary swap partition.
- The `b` partition is in use as a 2 GB UFS file system dedicated to user files.
- The `g` and `h` partitions are unused and total 6 GB in additional disk storage capacity.

In the preceding example, you can extend the UFS file system on the `b` partition by an additional 6 GB. You need not take the entire 6 GB in a single extension; you can stage the extension to conserve disk space.

If you do not have adequate disk capacity to extend the file system, back it up and restore it to as new disk volume as described in Chapter 9.

Reset the partition size

Use the `disklabel` command to reset the size of the partition on which the file system resides. The following procedure describes the use of the `disklabel` command.

Note

You cannot use the graphical disk configuration utility, `diskconfig`, to perform this operation because the `diskconfig` utility does not bypass the partition verification and disallows reconfiguration of any partitions that are in use.

1. Save the current disk label to a file so that you can edit the partitions. For example:

   ```
   # /sbin/disklabel -r /dev/disk/dsk4 > d4label
   ```

2. Edit the saved label to increase the capacity of the partition in use, decreasing the unused partition by an identical amount.

   ```
   .
   .
   .
   b: 10192000 1048576 4.2BSD 1024 8192 16 # (Cyl.  312*- 2750*)
   g:  7104147 8669377 unused 1024 8192    # (Cyl. 2580*- 5289*)
   ```

   For example, to increase the size of partition `g` by 3,000,000 blocks, change the label as follows:

   ```
   .
   .
   .
   b: 13192000 1048576 4.2BSD 1024 8192 16 # (Cyl.  312*- 2750*)
   g:  4104147 8669377 unused 1024 8192    # (Cyl. 2580*- 5289*)
   ```

   Save the disklabel file and exit from the editor.

3. Write the label to the raw disk, specifying the edited file as follows:

   ```
   # /sbin/disklabel -R /dev/rdisk/dsk4 d4label
   ```

See `disklabel`(8) for more information.

After you create additional disk space, extend the file systems by using one of the methods described in the following sections.

6.3.6.1 Extending a Dismounted File System

Use the `extendfs` command to extend a file system that is off line and in use. You can extend the entire partition on a single operation, or extend the file

system in stages. Use the extendfs command in either of the two following ways. These procedures assume that you completed the prerequisite step to increase the disk partition size, as described in Section 6.3.6.

- To extend the file system to the entire partition, use a command similar to the following:

```
# /sbin/extendfs dsk4
Warning: 1324 sector(s) in last cylinder unallocated
/dev/rdisk/dsk4h: 9057236 sectors in 2696 cylinders of 20
  tracks, 168 sectors
      4422.5MB in 169 cyl groups (16 c/g, 26.25MB/g, 6336 i/g)
super-block backups (for fsck -b #) at:
 32, 53968, 107904, 161840, 215776, 269712, 323648, 377584,
.
.
.
```

  The output from this command is similar to the output from the /sbin/newfs command, used for creating new UFS file systems. See newfs(8) for more information.

- To extend the file system to use only part of the available partition space, use the -s option as follows:

```
# /sbin/extendfs -s 500000
```

  This example takes only 500,000 blocks of the available partition space, saving the remainder for future extensions. As shown in the preceding example, the output from the /sbin/extendfs command is similar to the output from the /sbin/newfs command.

6.3.6.2 Extending a Mounted File System

Use the mount command to extend a file system that is on line (mounted) and in use. This procedure assumes that you completed the prerequisite steps to identify the mounted partition and increase the disk partition size, as described in Section 6.3.6.

The syntax of the mount command is described in Section 6.3.4. To extend a file system, you use the extend option with the -o option and specify the mount point as follows:

```
# /sbin/mount -u -o extend /projects
extending file system, please wait.
```

The mount command does not display any completion or error output for this operation. The length of time for completion depends on the size of the partition, and may take several minutes to complete. You should therefore verify the operation using the df command as follows:

```
# /usr/bin/df /projects
```

The output from this command confirms if the operation was a success. If the operation was not a success, verify that the disk partition is correct and the mount point exists.

6.4 Administering UFS File Systems Using SysMan Menu

In addition to the manual method of file system creation and administration, the operating system provides some graphical tools, and also some SysMan Menu tasks, which you can use in different user environments. See Chapter 1 for information on invoking and using SysMan. If you are using the Common Desktop Environment, other graphical utilities are available. Access these from the CDE Application Manager main folder as follows:

1. Select the Application Manager icon from the CDE front panel
2. Select the System_Admin icon from the Application Manager folder window
3. Select the Storage_Management icon from the Application Manager – System_Admin folder window

Depending on what options are installed and licensed on your system, the following icons may be available in this window:

Advanced File System	Select this icon to run the AdvFS graphical user interfaces. See the *AdvFS Administration* manual for more information. See `dtadvfs`(8) for information on launching the Advfs graphical interface from the command line.
Bootable Tape	Select this icon to invoke the SysMan Menu Bootable Tape Creation interface. Use this interface to create a bootable system image on tape. This image contains a standalone kernel and copies of selected file systems that you specify during creation. You can recover the image using the `btextract` utility. See Chapter 9 for information on using the bootable tape interfaces. See `btcreate`(8), `btextract`(8), and `bttape`(8) for more information. The `bttape` command is used to launch the

	bootable tape graphical user interface from a command line or script.
File System Management	Select this icon to invoke the SysMan Storage utilities described in this section.
Logical Storage Manager (LSM)	Select this icon to invoke the LSM graphical user interface. Logical Storage Management enables you to create virtual disk volumes that appear as a single device to the system and any applications. See the *Logical Storage Manager* manual for more information and `lsm`(8) for a list of LSM commands. To invoke this interface from the command line, use the `lsmsa` or `dxlsm` command. See `lsmsa`(8) for more information; the `dxlsm` is scheduled for retirement in a later release of the operating system.
Prestoserve I/O Accelerator	Select this icon to invoke the Prestoserve graphical utilities. Prestoserve stores synchronous disk writes in nonvolatile memory instead of writing them to disk. Then, the stored data is written to disk asynchronously as needed or when the machine is halted. See the *Guide to Prestoserve* manual for more information and `presto`(8) for information on the command line interface. To invoke this interface from the command line, use the `dxpresto` command. See `dxpresto`(8) for more information.

The following sections describe the UFS file system utilities in the SysMan Menu.

6.4.1 File System Tasks in the SysMan Menu

The SysMan Menu contains a main menu option titled Storage. When expanded, these options appear as follows:

```
- Storage
  - File Systems Management Utilities
     - General File System Utilities
            | Dismount a File System
            | Display Currently Mounted File Systems
            | Mount File Systems
            | Share Local Directory (/etc/exports)
            | Mount Network Directory (/etc/fstab)
       - Advanced File System (AdvFS) Utilities
            | Manage an AdvFS Domain
            | Manage an AdvFS File
            | Defragment an AdvFS Domain
            | Create a New AdvFS Domain
            | Create a New AdvFS Fileset
            | Recover Files from an AdvFS Domain
            | Repair an AdvFS Domain
        - UNIX File System (UFS) Utilities
            | Create a New UFS File System
        - Logical Storage Manager (LSM) Utilities
            | Initialize the Logical Storage Manager (LSM)
            | Logical Storage Manager
```

Each option provides a step-by-step interface to perform basic file system administrative tasks. See Chapter 1 for information on invoking and using the SysMan Menu. You can launch the file system utilities from the SysMan Station also. For example, if you are using the SysMan Station to display the Mounted_Filesystems view, you can press MB3 to do the following:

- Launch any available Storage options, such as Dismount to unmount a mounted file system.
- Display properties of file systems such as the mount point or space used.

The SysMan Station Physical_Filesystems view provides a graphical view of file systems mapped to physical devices and enables you to perform tasks such as make AdvFS filesets on an existing domain. See Chapter 1 for information on invoking and using the SysMan Station. See the online help for information on using its file system options.

The following SysMan Menu Storage options are documented in other books:

Advanced File System (AdvFS) Utilities

See the *AdvFS Administration* manual.

Logical Storage Manager (LSM) Utilities

See the *Logical Storage Manager* manual.

The following sections describe the General File System Utilities and the UNIX File System (UFS) Utilities file system tasks available from the SysMan Menu. The typical procedure for creating a file system is exactly as described in Section 6.3, although the SysMan Menu tasks are not organized in the same sequence. These tasks are general-purpose utilities that you can use any time to create and administer file systems.

6.4.2 Using SysMan to Dismount a File System

To dismount a file system you need to specify its mount point, device special file name, or AdvFS domain name. You can obtain this information by using the more command to display the contents of the /etc/fstab file, or by using the SysMan Menu Storage option Display Currently Mounted File Systems described in Section 6.4.3. See mount(8) and umount(8) for the command line options.

The Dismount a File System option is available under the SysMan Menu Storage options. Expand the menu and select General File System Utilities if it is not displayed. When you select this option, a window titled Dismount a file system is displayed, prompting you to complete either of the following fields. You do not need to complete both fields:

1. Mount point:

   Enter the mount point on which the file system is currently mounted, such as /mnt.

2. File system name:

   Enter the device special file name for the mounted partition, such as /dev/disk/dsk0f, or an AdvFS domain name such as accounting_domain#act.

Select Apply to dismount the file system and continue dismounting other file systems, or select OK to dismount the file system and exit.

6.4.3 Using SysMan to Display Mounted File Systems

The option to display mounted file systems is available under the SysMan Menu Storage options. Expand the menu and select General File System Utilities – Display Currently Mounted File Systems. When you select this option, a window titled Currently Mounted File Systems is displayed, containing a list of the file systems similar to the following:

```
/dev/disk/dsk0a         /
/proc                   /proc
usr_domain#usr          /usr
usr_domain#var          /var
19serv:/share/19serv/tools/tools /tmp_mnt/19serv/tools
```

The following information is provided in the window:

File System — This can be one of the following:

- The special device file name from the `/dev/*` directories that maps to the mounted device partition. The pathname `/dev/disk/dsk0a` indicates partition `a` of disk `0`. See the *Hardware Management* manual for information on device names and device special files.
- An NFS (Network File System) mounted file share, possibly mounted using the `automount` or `autofs` utilities, which automatically mount exported networked file systems when a local user accesses (imports) them. See the *Network Administration: Services* manual for information on NFS, `automount` and `autofs`. An NFS mount typically lists the exporting host system name, followed by the exported directory as follows:

```
19serv:/share/19serv/tools/tools   /tmp_mnt/19serv/tools
```

  `19serv:` is the host name identifier followed by a colon

  `/share/19serv/tools/tools` is the pathname to the exported directory

  `/tmp_mnt/19serv/tools` is the temporary mount point that is created by NFS automatically.

- An AdvFS domain name such as `usr_domain#var`. See the *AdvFS Administration* manual or `advfs`(4) for information on domains.
- A descriptive name, such as `file-on-file mount`, which would point to a service mount point such as `/usr/net/servers/lanman/.ctrlpipe`

Mount Point — The directory on which the file system is mounted, such as `/usr` or `/accounting_files`.

The list can be extensive, depending on the number of currently mounted file systems. The list can provide information on current file-on-file mounts that

may not be visible in the /etc/fstab file. Files in the /etc/fstab file that are not currently mounted are not included in this list.

The following options are available from the Currently Mounted File Systems window:

Details...

Use this option to display detailed file system data, otherwise known as the properties of the file system. You can obtain the following data from this option:

```
File system name: /dev/disk/dsk0a
Mount point:      /
File system size: 132 MBytes
Space used:       82 MBytes
Space available:  35 MBytes
Space used %:     70%
```

Dismount...

Use this option to dismount a selected file system, You are prompted to confirm the dismount request. You cannot dismount the file system if it is currently in use or even if a user has run the cd command to change directory to the file system that you want to dismount. Use the wall command if you want to ask users to stop using the file system.

Reload

Use this option to refresh the Currently Mounted File Systems list and update any dismounted file systems.

If you mount file systems using the command line, or if NFS mounts are established, these newly mounted systems are not displayed until you exit the utility and invoke it again.

OK

Select OK to exit the Currently Mounted File Systems window and return to the SysMan Menu.

6.4.4 Using SysMan to Mount File Systems

The operation of mounting a file system has the following prerequisites:

- The file system must be listed in the /etc/fstab file.
- The mount point must exist. If not, use the mkdir command to create a mount point. See mkdir(1) for more information.
- The file system must be created on a disk partition, and the disk must be on line. See Section 6.4.7 for information on creating UNIX File Systems (UFS) using the SysMan Menu. See Section 6.3.1 for information on

manually creating file systems using the newfs command. See newfs(8) for more information. Information on creating AdvFS file systems is located in the *AdvFS Administration* manual.

The diskconfig graphical utility provides a way to customize disk partitions and write a file system on the partition in a single operation. See the *Hardware Management* manual for information on the diskconfig command, and see diskconfig(8) for more information on launching the utility. You can launch this utility from the SysMan Menu, from the SysMan Station, or from the CDE Application Manager also.

- The device is visible to the system.

  Normally, the availability of disk devices is managed automatically by the system. However, if you have just added a device dynamically, while the system is still running, it may not yet be visible to the system and you may have to tell the system to find the device and bring it on line.

  Use the hwmgr command to do this, and to verify the status of disk devices and partitions for existing disks (if necessary). See hwmgr(8) for more information. See the *Hardware Management* manual for information on administering devices.

  Normally, the device special files for a disk partition, such as /dev/disk/dsk5g, are created automatically and maintained by the system. However if you do not find the device special file, you may need to create it.

  See the *Hardware Management* manual for information on the dsfmgr command, and see dsfmgr(8) for information on command options such as dsfmgr -s, which lists the device special files for each device (Dev Node).

The option to mount a file system is available under the SysMan Menu Storage options. Expand the menu and select General File System Utilities – Mount File Systems to display the Mount Operation window. This interface provides an alternative to the mount command, described in mount(8). This utility operates only on the file systems currently listed in the /etc/fstab file. You can obtain information on the mounted file systems using the Display Mounted Filesystems SysMan Menu option, described in Section 6.4.3.

The Mount Operation window provides the following four exclusive selectable options:

1. Mount a specific file system.

   Select this option to mount a single specific file system. The File System name and Mount Point window are displayed, prompting you to complete either of the following fields:

Mount point:	Type the mount point directory from the `/etc/fstab` file, such as `/cdrom`.
File system name:	Type a device special file name, such as `/dev/disk/cdrom0c`. Alternatively, type an AdvFS domain name, such as `usr_domain#usr`.

The File System Mounting Options window are displayed next. This window is common to several of the mounting operations, and is described at the end of this list.

2. Mount all file systems listed in `/etc/fstab`.

   Use this option to mount all file systems currently listed in the `/etc/fstab` file. Using the option assumes that all the specified partitions or domains are on line, and all the mount points have been created.

   The File System Mounting Options window are displayed next. This window is common to several of the mounting operations, and is described at the end of this list.

3. As above, but only those of a specified type.

   Use this option to mount all file systems of a specified type listed in the `/etc/fstab` file. Using the option assumes that all the specified partitions or domains are on line, and all the mount points have been created.

   You specify the file system type in the File System Mounting Options window, which is displayed next. This window is common to several of the mounting operations, and is described at the end of this list. For example, you can choose to include only AdvFS file systems.

4. Mount all file systems not of the selected type.

   Use this option to exclude from the mount operation, all file systems of a specified type listed in the `/etc/fstab` file. Using the option assumes that all the specified partitions or domains are on line, and all the mount points have been created.

   You specify the file system type to be excluded in the File System Mounting Options window, which is displayed next. This window is common to several of the mounting operations, and is described at the end of this list. For example, you can choose to exclude only UFS file systems.

The File System Mounting Options window is common to several of the preceding list of mount options, and enables you to specify additional optional characteristics for the mount operation. Some options may not be

available, depending on the type of mount operation that you are attempting. The following options are available from this window:

Access Mode	Select the type of access that you want to enable:	
	Read/Write	Select this option to permit authorized users to read from and write to files in the file system.
	Read only	Select this option to permit authorized users only to read from files in the file system, or to mount read-only media such as a CD-ROM volume.
File system type	From the menu, select one of the following options:	
	Unspecified	Select this option to allow any file system specification.
	AdvFS	Select this option to specify an Advanced File System type. See the *AdvFS Administration* manual or `advfs`(4) for more information.
	UFS	Select this option to specify a UNIX File System type. See Section 6.1.4 for a description of this file system.
	NFS	Select this option to specify a Networked File System. See the *Network Administration: Services* manual and `nfs`(4) for more information.
	CDFS	Select this option to specify a Compact Disk Read Only Memory File System. See `cdfs`(4) for more information.

	Other	Select this option to enter your own file system choice in the Other file system type: field described in the next item.
Other file system type	Type the designation for the file system such as `mfs` for the memory file system (ram disk). See `mount`(8) for more information on supported file systems, and see the individual file system reference pages, such as `newfs`(8) for the memory file system.	
Advanced Mount options	Type any advanced mount options that you want for the file system. For example, the `dirty` option, which allows a file system to be mounted even if it was not dismounted cleanly, such as after a system crash. See `mount`(8) for more information on the various options.	

After you have entered the options you want, select `Finish` to process the mount operation and return to the SysMan Menu options. Select `Back` to return to the Mount Operation window and process new mount operations, or select `Cancel` to terminate the mount operation.

If data in any field is incomplete or incorrect, you are prompted to correct it before the mount operation can proceed.

6.4.5 Using SysMan to Share a Local Directory

File sharing involves adding file systems to the `/etc/exports` file so that users of other host systems can mount the shared directories by means of NFS (Network File System). If the Advanced Server for UNIX (ASU) is installed and running, you may have further options to share file systems with PC clients. See the ASU *Concepts and Planning Guide*.

You may need to enable network access to your system for remote hosts to mount the shared directories, such as by adding the hosts to the `/etc/hosts` file, setting up NFS, and running `dxhosts`. See the *Network Administration: Services* manual for information on configuring your system to allow incoming connections to shared file systems.

You can manage shared file systems using the `dxfileshare` graphical user interface, which you can launch from the command line or from the CDE Application Manager – DailyAdmin folder. See the File Sharing option in that folder. Online help is available for this interface. See `dxfileshare`(8) for more information on invoking the interface.

The only prerequisite for shared file systems is that you should have created disk file systems that are suitable for sharing as described in Section 6.3.1 (manual method) or Section 6.4.7 (using SysMan Menu options) already. You specify the shared file system by its directory pathname, such as `/usr/users/share`.

The file system sharing option is available under the SysMan Menu Storage branch as follows:

```
-Storage
   - File Systems Management Utilities
     - General File System Utilities
         | Share Local Directory (/etc/exports)
```

6.4.5.1 Sharing a File System

Follow these steps to share an existing file system:

1. In the window titled Share Local Directory on *hostname.xxx.yyy.xxx*, any existing shares are listed in the first box, identified by the directory pathname.

   Select `Add...` to add a directory to the list. A window titled `Share Local Directory: Add Local Directory` opens.

2. Enter the directory path name, such as `/usr/users/share/tools`, in the field labeled `Share This Directory:`.

3. Choose whether to share the directory with read/write access or read-only access. `Read/Write` is selected by default.

4. Choose whether to share the directory with all qualified hosts (remote systems) or only with named hosts as follows:

`All`	Select this to designate all hosts.
`Selected`	Select this to designate a list of certain hosts.

5. If you selected `Selected` in the previous step, enter the host name and address, such as `dplhst.xxx.yyy.com`. The host must be known to your local host, either through the `/etc/hosts` file or through a domain name server (DNS). See the *Network Administration: Services* manual for more information.

6. Select `OK` to validate the data and close the dialog box and return to the `Share Local Directory on` *host name* window.

   All changes are deferred until you select `OK` in this window.

   When you select `OK`, the directories are made available for sharing.

6.4.5.2 Removing a Shared File System

To remove a share, use the same utility as follows:

1. Delete hosts from the access list.
2. Modify access to shared file systems by changing the read/write permissions or removing selected hosts from the access list.
3. Delete shared file systems from the shared list to prevent any access.

6.4.6 Using SysMan to Mount a Network File System

You can mount shared file systems that are shared (exported) by other hosts using the Network File System (NFS). Your local system (host) must be configured to import NFS-shared file systems, including authorized network access to remote hosts. Remote systems (hosts) must be configured to share or export file systems by specifying your system in their `/etc/exports` files. You can mount NFS-shared file systems in several ways:

Temporarily, where the mount does not persist across a reboot.

> A mount point is created and the file system is connected for the current session. If the system is shut down for any reason, the mount point persists, but the file system connection is lost and is not reestablished when the system is booted.

Permanently, by specifying the shared NFS file systems in your local `/etc/fstab` file.

> For example, your `/etc/fstab` file already may have one or more NFS file system entries similar to the following:
>
> ```
> /usr/lib/toolbox@ntsv /usr/lib/toolbox nfs rw,bg,soft,nosuid 0 0
> ```
>
> See Section 6.3.3 for a description of the structure of an `/etc/fstab` file.

Automatically on request from a user, using the NFS `automount` utility.

> See the *Network Administration: Services* manual and `automount`(8) for information on using this option. Using `automount` enables local users to mount any file systems that are shared with (exported to) your local system transparently. You do not need to respond to mount requests from users constantly.

The information in this section enables you to add more NFS shares permanently to your `/etc/fstab` file or to create temporary imports of shared file systems.

See Section 6.4.5 for a description of the process of sharing (exporting) file systems using the SysMan Menu options.

You can manage shared file systems using the File Sharing graphical user interface (GUI), which you can launch from the command line, or from the CDE Application Manager – DailyAdmin folder. See the File Sharing option in that folder. Online help is available for this interface. See `dxfileshare`(8) for more information on invoking the interface.

6.4.6.1 Mounting a Shared Network File System

The option to mount NFS file systems is available under the SysMan Menu Storage options. Expand the menu and select `General File System Utilities – Mount Network Directory (/etc/fstab)`. Follow these steps to mount a shared file system:

1. In the window titled Mount Network Directory on *hostname*, there is a list of existing available NFS shared file systems listed in the `/etc/fstab` file, which provides you with the following information:

Directory and Host	The name of the host, and the directory it is exporting to your local system.
Mounted On	The local mount point on which the shared file system is mounted. This is a directory pathname, such as `/tools/bin/imaging`.
Options	The access options for the directory, which can be as follows:

Read/Write	Allows users to both read data from and write data to the shared file system. This may depend on the access conditions set by the exporting host.
Read-Only	Allows users only to read data from the shared file system.

Reboot	Indicates whether the mount is reestablished if the system is shut down for any reason, and can be as follows:

true	Permanent; the entry is in the local `/etc/fstab` file and the mount persists across reboots.
false	Temporary; the entry is not in the local `/etc/fstab` file and the mount does not persist.

6.4.6.2 Adding a Network Directory

Select `Add...` to add a file system to the list of NFS-shared directories. A window titled Mount Network Directory: Add Network Directory is displayed.

When you use this option, file systems are mounted with the options `hard` (retries until a response is received) and `bg` (background mount) by default. See `mount`(8) for more information on these options.

6.4.7 Using SysMan to Create a UFS File System

Creating a UFS file system manually using the `newfs` command is described in Section 6.3.1 and the same prerequisites and sources of data apply to the process of creating a file system with the SysMan Menu options, except that you are limited to standard disk partitions. If you want to use custom partitions, use the `diskconfig` utility as described in the *Hardware Management* manual.

Obtain the following items of data before proceeding:

- Information about where the file system is to be stored, specified by either of the following:
  - The device special file name of the disk partition on which the file system is to be created, such as `/dev/disk/dsk13h` for the `h` partition on disk 13.
  - If the Logical Storage Manager application is in use, an LSM volume name. See the *Logical Storage Manager* manual for more information.
- The disk model, such as RZ1DF-CB. You can obtain such information using the `hwmgr` command as follows:

```
# hwmgr -view devices
```

Alternatively, use the SysMan Station Hardware View, select the disk, press MB3 and choose Properties... from the pop-up menu to view details of the device. The /etc/disktab file is a source of information on disk models. See disktab(4) for information on the /etc/disktab file structure.

- Determine whether you need any particular options for the file system, such as block size or optimization. See newfs(8) for a complete list of options. You can display the options from within the SysMan Menu utility.

The option to create a new UFS file system is available under the SysMan Menu Storage options. Expand the menu and select UNIX File System (UFS) Utilities – Create a New UFS File System. A window titled Create a new UFS File System is displayed next.

Follow these steps to create a file system:

1. Enter the name of the disk partition or the LSM volume that you selected to store the file system in the field labeled Partition or LSM Volume.
2. Enter the name of the disk model, such as HB00931B93, in the field named Disk type.
3. Enter any option flags, such as -b 64 for a 64 kilobyte block size, in the Advanced newfs options field.

   If you are unsure what options to use, clear all fields and select Apply. This displays a newfs information window, containing a list of flag options.

Select OK to create the file system and exit to the SysMan Menu or select Apply to create the file system and continue creating more file systems. To terminate the operation, select Cancel.

Use the SysMan Menu option Mount File Systems described in Section 6.4.4 to mount the newly created file systems.

6.5 Managing Quotas

The following sections describe user and group quotas for UFS. AdvFS also supports fileset quotas, which limit the amount of space a fileset can have. For information about AdvFS fileset quotas, see the *AdvFS Administration* manual, which also has AdvFS-specific information about user and group quotas.

As a system administrator, you establish usage limits for user accounts and for groups by setting quotas for the file systems they use. Thus, user and group quotas are known as file system quotas. The file system quotas are

known as disk quotas because, when established, they limit the number of disk blocks used by a user account or a group of users.

You set quotas for user accounts and groups by file system. For example, a user account can be a member of several groups on a file system and also a member of other groups on other file systems. The file system quota for a user account is for a user account's files on that file system. A user account's quota is exceeded when the number of blocks (or inodes) used on that file system are exceeded.

Like user account quotas, a group's quota is exceeded when the number of blocks (or inodes) used on a particular file system is exceeded. However, the group blocks or inodes used only count toward a group's quota when the files that are produced are assigned the group ID (GID) for the group. Files that are written by the members of the group that are not assigned the GID of the group do not count toward the group quota.

Note

Quota commands display block sizes of 1024-bytes instead of the more common 512–byte size.

You can apply quotas to file systems to establish a limit on the number of blocks and inodes (or files) that a user account or a group of users can allocate. You can set a separate quota for each user or group of users on each file system. You may want to set quotas on file systems that contain home directories, such as `/usr/users`, because the sizes of these file systems can increase more significantly than other file systems. You should avoid setting quotas on the `/tmp` file system.

6.5.1 Hard and Soft Quota Limits

File system quotas can have both soft and hard quota limits. When a hard limit is reached, no more disk space allocations or file creations that would exceed the limit are allowed. A hard limit is one more unit (such as one more block, file, or inode) than would be allowed when the quota limit is active.

The quota is up to, but not including the limit. For example, if a hard limit of 10,000 disk blocks is set for each user account in a file system, an account reaches the hard limit when 9,999 disk blocks have been allocated. For a maximum of 10,000 complete blocks for the user account, the hard limit should be set to 10,001.

The soft limit may be reached for a period of time (called the grace period). If the soft limit is reached for an amount of time that exceeds the grace period, no more disk space allocations or file creations are allowed until enough disk

space is freed or enough files are deleted to bring the disk space usage or number of files below the soft limit.

As an administrator, you should set the grace period large enough for users to finish current work and then delete files to get their quotas down below the limits you have set.

Caution

With both hard and soft limits, it is possible for a file to be partially written if the quota limit is reached when the write occurs. This can result in the loss of data unless the file is saved elsewhere or the process is stopped.

For example, if you are editing a file and exceed a quota limit, do not abort the editor or write the file because data may be lost. Instead, escape from the editor you are using, remove the files, and return to the session. You can write the file to another file system, remove files from the file system whose quota you reached, and then move the file back to that file system.

6.5.2 Activating File System Quotas

To activate file system quotas on UFS, perform the following steps.

1. Configure the system to include the file system quota subsystem by editing the `/sys/conf/`*`NAME`* system configuration file to include the following line:

   ```
   options       QUOTA
   ```

2. Edit the `/etc/fstab` file and change the fourth field of the file system's entry to read `rw`, `userquota`, and `groupquota`. See `fstab`(4) for more information.
3. Use the `quotacheck` command to create a quota file where the quota subsystem stores current allocations and quota limits. See `quotacheck`(8) for command information.
4. Use the `edquota` command to activate the quota editor and create a quota entry for each user.

   For each user or group you specify, `edquota` creates a temporary ASCII file that you edit with any text editor. Edit the file to include entries for each file system with quotas enforced, the soft and hard limits for blocks and inodes (or files), and the grace period.

   If you specify more than one user name or group name in the `edquota` command line, the edits affect each user or group. You can use

prototypes that allow you to set up quotas for groups of users quickly, as described in Section 6.5.3.

5. Use the quotaon command to activate the quota system. See quotaon(8) for more information.
6. To verify and enable file system quotas during system startup, use the following command to set the file system quota configuration variable in the /etc/rc.config file:

```
# /usr/sbin/rcmgr set QUOTA_CONFIG yes
```

Note

Setting QUOTQ_CONFIG to yes causes the quotacheck command to be run against the UFS file systems during startup. The AdvFS design does not need this service. While it is not recommended, you can force quotacheck to be run against both UFS and AdvFS file systems during system startup using the following command:

```
# /usr/sbin/rcmgr set \
QUOTACHECK_CONFIG -a
```

To restore the default UFS-only quotacheck behavior, use the following command:

```
# /usr/sbin/rcmgr set \
QUOTACHECK_CONFIG ""
```

If you want to turn off quotas, use the quotaoff command. Also, the umount command turns off quotas before it unmounts a file system. See quotaoff(8) for more information.

6.5.3 Setting File System Quotas for User Accounts

To set a file system quota for a user, you can create a quota prototype or you can use an existing quota prototype and replicate it for the user. A quota prototype is an equivalent of an existing user's quotas to a prototype file, which is then used to generate identical user quotas for other users. Use the edquota command to create prototypes. If you do not have a quota prototype, create one by following these steps:

1. Log in as root and use the edquota command with the following syntax:

   edquota *proto-user users*

   For example, to set up a quota prototype named large for user eddie, enter the following command:

   ```
   # edquota large eddie
   ```

The program creates the large quota prototype for user eddie. You must use a real login name for the *users* argument.

2. Edit the quota file opened by the edquota program to set quotas for each file system that user eddie can access.

To use an existing quota prototype for a user:

1. Enter the edquota command with the following syntax:

   edquota –**p** *proto-user users*

   For example, to set a file system quota for the user marcy, using the large prototype, enter:

   ```
   # edquota -p large marcy
   ```

2. Confirm that the quotas are what you want to set for user marcy. If not, edit the quota file and set new quotas for each file system that user marcy can access.

6.5.4 Verifying File System Quotas

If you are enforcing user file system quotas, you should verify your quota system periodically. You can use the quotacheck, quota, and repquota commands to compare the established limits with actual use.

The quotacheck command verifies that the actual block use is consistent with established limits. You should run the quotacheck command twice: when quotas are first enabled on a file system (UFS and AdvFS) and after each reboot (UFS only). The command gives more accurate information when there is no activity on the system.

The quota command displays the actual block use for each user in a file system. Only the root user can execute the quota command.

The repquota command displays the actual disk use and quotas for the specified file system. For each user, the current number of files and the amount of space used (in kilobytes) is displayed along with any quotas.

If you find it necessary to change the established quotas, use the edquota command, which allows you to set or change the limits for each user.

See quotacheck(8), quota(1), and repquota(8) for more information on file system quotas.

6.6 Backing Up and Restoring File Systems

The principal backup and restore utilities for both AdvFS and UFS are the vdump and the vrestore utilities. These utilities are used for local operations on both AdvFS and UFS file systems. The utilities are described

in vdump(8) and vrestore(8). For remote backup and restore operations on both AdvFS and UFS file systems, the utilities are rvdump and rvrestore.

For administrators who want to back up only UFS, the traditional utilities are described in dump(8) and restore(8).

Examples of backup and restore operations for AdvFS are described in the *AdvFS Administration* manual. Examples of backup and restore operations for UFS are described in Chapter 9, which also describes the process for creating a bootable tape. While this is not strictly a backup, it does provide a method of creating a bootable magnetic tape copy of the root file system and important system files from which you can boot the system and recover from a disaster such as a root disk crash.

Another archiving service is the Networker Save and Restore product, also described in Chapter 9.

6.7 Monitoring and Tuning File Systems

The following sections describe commands you use to display information about, and verify UFS file systems. They also include some basic information on file system tuning. For a more detailed discussion of tuning, see the *System Configuration and Tuning* manual.

6.7.1 Verifying UFS Consistency

The fsck program verifies UFS and performs some corrections to help ensure a reliable environment for file storage on disks. The fsck program can correct file system inconsistencies such as unreferenced inodes, missing blocks in the free list, or incorrect counts in the superblock.

File systems can become corrupted in many ways, such as improper shutdown procedures, hardware failures, power outages, and power surges. A file system can become corrupted if you physically write protect a mounted file system, take a mounted file system off line, or if you do not use the sync command before you shut the system down.

At boot time, the system runs fsck noninteractively, making any corrections that can be done safely. If it encounters an unexpected inconsistency, the fsck program exits, leaves the system in single-user mode, and displays a recommendation that you run the program manually, which allows you to respond yes or no to the prompts that fsck displays.

The command to invoke the fsck program has the following syntax:

/usr/sbin/fsck [*options* ...] [*file_system* ...]

If you do not specify a file system, all the file systems in the /etc/fstab file are verified. If you specify a file system, always use the raw device.

See fsck(8) for information about command options.

Note

To verify the root file system, you must be in single-user mode, and the file system must be mounted read only. To shut down the system to single-user mode use the shutdown command that is described in Chapter 2.

6.7.2 Monitoring File System Use of Disks

To ensure an adequate amount of free disk space, you should monitor the disk use of your configured file systems regularly. You can do this in any of the following ways:

- Verify available free space by using the df command
- Verify disk use by using the du command or the quot command
- Verify file system quotas (if imposed) by using the quota command

You can use the quota command only if you are the root user.

6.7.2.1 Examinng for Available Free Space

To ensure sufficient space for your configured file systems, you should use the df command regularly to display the amount of free disk space in all the mounted file systems. The df command displays statistics about the amount of free disk space on a specified file system or on a file system that contains a specified file.

The df command has the following syntax:

df [- eiknPt] [- F *fstype*] [*file*] [*file_system* ...]

Without arguments or options, the df command displays the amount of free disk space on all the mounted file systems. For each file system, the df command reports the file system's configured size in 512-byte blocks, unless you specify the -k option, which reports the size in kilobyte blocks. The command displays the total amount of space, the amount presently used, the amount presently available (free), the percentage used, and the directory on which the file system is mounted.

For AdvFS file domains, the df command displays disk space usage information for each fileset.

If you specify a device that has no file systems mounted on it, df displays the information for the root file system.

You can specify a file pathname to display the amount of available disk space on the file system that contains the file.

You cannot use the `df` command with the block or character special device name to find free space on an unmounted file system. Instead, use the `dumpfs` command.

See `df`(1) for more information.

The following example displays disk space information about all the mounted file systems:

```
# /sbin/df
Filesystem         512-blks   used  avail capacity Mounted on
/dev/disk/dsk2a      30686  21438   6178    77%   /
/dev/disk/dsk0g     549328 378778 115616    76%   /usr
/dev/disk/dsk2      101372   5376  85858     5%   /var
/dev/disk/dsk3      394796     12 355304     0%   /usr/users
/usr/share/man@tsts 557614 449234  52620    89%   /usr/share/man
domain#usr          838432 680320  158112   81%   /usr
```

Note

The `newfs` command reserves a percentage of the file system disk space for allocation and block layout. This can cause the `df` command to report that a file system is using more than 100 percent of its capacity. You can change this percentage by using the `tunefs` command with the `-minfree` flag.

6.7.2.2 Verifying Disk Use

If you determine that a file system has insufficient space available, examine how its space is being used. You can do this with the `du` command or the `quot` command.

The `du` command pinpoints disk space allocation by directory. With this information you can decide who is using the most space and who should free up disk space.

The `du` command has the following syntax:

/usr/bin/du [- aklrsx] [*directory* ... *filename* ...]

The `du` command displays the number of blocks contained in all directories (listed recursively) within each specified directory, file name, or (if none are specified) the current working directory. The block count includes the indirect blocks of each file in 1-kilobyte units, independent of the cluster size used by the system.

If you do not specify any options, an entry is generated only for each directory. See `du`(1) for more information on command options.

The following example displays a summary of blocks that all main subdirectories in the /usr/users directory use:

```
# /usr/bin/du –s /usr/users/*
440     /usr/users/barnam
43      /usr/users/broland
747     /usr/users/frome
6804    /usr/users/norse
11183   /usr/users/rubin
2274    /usr/users/somer
```

From this information, you can determine that user rubin is using the most disk space.

The following example displays the space that each file and subdirectory in the /usr/users/rubin/online directory uses:

```
# /usr/bin/du -a /usr/users/rubin/online
1 /usr/users/rubin/online/inof/license
2 /usr/users/rubin/online/inof
7 /usr/users/rubin/online/TOC_ft1
16 /usr/users/rubin/online/build
  .
  .
  .
251 /usr/users/rubin/online
```

As an alternative to the du command, you can use the ls -s command to obtain the size and usage of files. Do not use the ls -l command to obtain usage information; ls -l displays only file sizes.

You can use the quot command to list the number of blocks in the named file system currently owned by each user. You must be root user to use the quot command.

The quot command has the following syntax:

/usr/sbin/quot [-c] [-f] [-n] [*file_system*]

The following example displays the number of blocks used by each user and the number of files owned by each user in the /dev/disk/dsk0h file system:

```
# /usr/sbin/quot -f /dev/disk/dsk0h
```

The character device special file must be used to return the information for UFS files, because when the device is mounted the block special device file is busy.

See quot(8) for more information.

6.7.3 Improving UFS read Efficiency

To enhance the efficiency of UFS reads, use the `tunefs` command to change a file system's dynamic parameters, which affect layout policies.

The `tunefs` command has the following syntax:

tunefs [-a *maxc*] [-d *rotd*] [-e *maxb*] [-m *minf*] [-o *op t*] [*file_s*]

You can use the `tunefs` command on both mounted and unmounted file systems; however, changes are applied only if you use the command on unmounted file systems. If you specify the root file system, you must reboot to apply the changes.

You can use command options to specify the dynamic parameters that affect the disk partition layout policies. See `tunefs`(8) for more information on the command options and `sys_attrs_ufs`(5) for information on UFS subsystem attributes.

6.8 Troubleshooting File Systems

Use the following tools to help you resolve problems associated with UFS file systems:

- Using the UNIX Shell Option

  The UNIX Shell Option is an installation option for experienced administrators and is available during either a textual or graphical installation of the operating system. For example, you may be able to recover from a corrupted root file system using this option.

  See the *Installation Guide* for an introduction to this installation option and the *Installation Guide — Advanced Topics* manual for an explanation of the file-system related administration you can accomplish with it. Use the shell option for both AdvFS and UFS file system problems.

- Using the `/usr/field` directory and the `fsx` command

  The `/usr/field` directory contains programs related to the field maintenance of the operating system. You can use the programs in this directory to monitor and exercise components of the operating system and system hardware.

  The `fsx` utility exercises file systems. See `fsx`(8) for more information. Other programs in the directory, such as a tape exerciser (`tapex`) and a disk exerciser (`diskx`) may be useful when investigating file system problems.

- Use the `dumpfs` utility to display information on UFS file systems. See `dumpfs`(8) for more information.

- Use the event manager, EVM (the Event Manager) to filter and display events that are related to file system problems. This utility is useful for setting up preventative maintenance and monitoring of file systems and storage devices. See Chapter 13 for information.
- Use the SysMan Station and Insight Manager to provide graphical views of file systems and to monitor and troubleshoot file system problems, such as lack of disk space. See Chapter 1 for information.

7

Administering User Accounts and Groups

Assigning user accounts and organizing user accounts into related groups is the most common way that you provide system resources to users. This chapter describes these user account and group administration topics:

- A discussion of the utilities that you can use to administer accounts and groups, and the user environments in which you can use these utilities (Section 7.1)
- A quick start section, providing brief information on the utilities; you can use the online help to guide you through a task (Section 7.2)
- Information to help you understand general account and group concepts (including LDAP and NIS), and important data items such as the unique identifiers assigned to accounts and groups; this section also describes the contents of the system data files for passwords and groups and how to set the default characteristics of an account or group (Section 7.3)
- Specific instructions on using utilities to perform administrative tasks on user accounts such as adding, modifying, and deleting user accounts and the associated system resources (Section 7.4)
- Specific instructions on using utilities to perform administrative tasks on user groups (Section 7.5)
- Information on administering associated (synchronized) Windows NT domain and UNIX accounts (Section 7.6)

7.1 Account Administration Options and Restrictions

Depending on your local system configuration, the user environment, and your personal preferences, there are several methods and a number of different utilities that you can use to administer user accounts. The following sections introduce and describe these options and identify any restrictions or requirements for their use.

7.1.1 Administrative Utilities

The operating system provides several different utilities that you can use to administer accounts. Not all are described in detail in this chapter. However, the principles of use are the same for all utilities. See the online

help and reference pages for each utility for specific information on the options available.

The utilities are listed in Table 7–1. You must be root user on the UNIX system or the Windows NT domain administrator to use these utilities.

Table 7–1: Utilities for Administering Accounts and Groups

Utility	User Environment Description
SysMan Menu Accounts options: Manage local users and groups Manage NIS users and groups Manage LDAP users and groups	You can use the SysMan Menu from a wide variety of user environments (see Chapter 1). This utility provides limited administrative features, such as adding and deleting accounts and groups. It does not enable you to administer the default characteristics for UNIX accounts and groups if you have Advanced Server for UNIX (ASU) installed. It does not allow you to choose the creation or deletion of associated (synchronized) Windows NT domain accounts but does this automatically, depending on how the account defaults are configured (with `useradd` or `usermod`). The filter (search) features provided by SysMan Menu Accounts options make it the preferred method of managing a high volume of user accounts.
Account Manager (`dxaccounts`)	This is a graphical user interface that provides most user and group administrative options for both UNIX and Windows NT domain accounts. This is an X11–based tool, rather than a SysMan Menu tool. CDE (the default UNIX environment) is X11–compliant.

`useradd`
`usermod`
`userdel`
	These are command line tools that run on the UNIX system in the character cell environment; they provide you with access to all user account administrative tasks. You can use these commands to administer both UNIX accounts and associated (synchronized) Windows NT domain accounts. You can use these commands to configure the default account environment also.

`groupadd`
`groupmod`
`groupdel`
	These are command line tools that run on the UNIX system in the character cell environment; they provide you with access to all user group administrative tasks. You can use these commands to configure the default UNIX group environment.

Table 7–1: Utilities for Administering Accounts and Groups (cont.)

Utility	User Environment Description
Advanced Server for UNIX (ASU) User Manager for Domains	This Microsoft Windows NT-based application for a PC system enables you to administer Windows NT domain accounts. You can use this, and other ASU utilities, to set up the default account characteristics using the policy management options. You cannot configure the default UNIX account environment.
ASU `net` commands	Commands that can be entered at a UNIX system terminal or at the DOS prompt on a system running the Windows NT server. These commands replicate the behavior of the ASU User Manager for Domains utility.

You must install and configure the Advanced Server for UNIX (ASU) software to use the Microsoft Windows-based utilities. Using the ASU utilities is not explained in detail in this chapter, but is discussed only in the context of a UNIX server running the ASU software. See the ASU *Installation and Administration Guide* for more information on installing and using ASU.

7.1.2 Notes and Restrictions on Using the Utilities

The following restrictions apply when using account management utilities, or when certain system features are enabled:

- Characteristics of the default UNIX account configuration

  You can use only the UNIX command line utilities or Account Manager `dxaccounts` to configure the default UNIX account and group characteristics.

  See the ASU *Installation and Administration Guide* for more information on setting default values for PC accounts when ASU is in use.

- Enhanced (C2) security

  When enhanced security is enabled, it places restrictions on account creation and enables additional features such as:

  1. Enhanced password controls
  2. Options for enabling and disabling (or locking) accounts
  3. Options for deleting and retiring accounts

  See the *Security Administration* manual for more information.

- Network Information Services (NIS)

  NIS enables users to log in to any system on the local network that is running NIS. User data, such as account name and password is shared

between all NIS systems, and users use different commands, such as `yppasswd`, instead of `passwd` to change passwords.

When NIS is configured, you have two potential classes of users to manage:

local users and groups
NIS users and groups

Features in the user account administration utilities that support NIS are enabled only when NIS is running. See the *Network Administration: Services* manual for information on setting up the NIS environment.

- Lightweight Directory Access Protocol (LDAP)

  LDAP is similar in concept to NIS. You have one repository, the LDAP database, against which to authenticate.

  LDAP enables users to log into any system on the local network that is running LDAP. User data, such as the account name and password, is shared among all LDAP systems, and users use different commands; one such example is the use of `ypasswd` instead of `passwd` to change passwords.

  When LDAP is configured, you have three potential classes of users to manage:

  local users and groups
  NIS users and groups
  LDAP users and groups

  Features in the user account administration utilities that support LDAP are enabled only when LDAP is running.

  See the OpenLDAP documentation at **`www.OpenLDAP.org`** for more information on LDAP.

- Multiple instances of account management utilities

  When invoked, any account management utility creates a lock file, preventing other account management utilities (or two instances of the same utility) from accessing system files such as `/etc/passwd`. This lock file is located at `/etc/.AM_is_running`. Creation of the lock file prevents possible corruption of account data in the system files. Under certain circumstances, this lock file may not clear correctly and you must delete it manually. Before you remove a lock file, ensure that it does not relate to a legitimate instance of an account management tool by running the `ps -ef` command to check for instances of `AM_is_running`.

  Only the command line utilities and the SysMan Menu tools support LDAP; the Account Manager utility (`dxaccounts`) does not.

The SysMan Menu Accounts options are designed to use deferred completion. This means that any data that you enter is stored and not written to a file until you confirm it. Therefore, while you can invoke a SysMan Menu Accounts option while another instance of an account management utility is running, you cannot select `Apply` or `OK` to update the system file. When the other instance of an account management utility is closed, the lock file is removed and you can complete the transaction.

The Division of Privileges (DOP) and distributed administration features enables the root user to easily assign account management privileges to other users. However, only one account management utility can be used by one authorized user at any time.

7.1.3 Related Documentation

Additional documentation on administering accounts can be found in manuals, reference pages, and online help.

7.1.3.1 Manuals

The following lists refers to information on administering accounts in the Tru64 UNIX operating system documentation set.

- Chapter 6 provides information on file systems and user file space.
- The *Network Administration: Services* manual provides information on NIS user accounts.
- The *Security Administration* manual provides information on important security considerations when assigning resources to users. Information on account requirements for enhanced security and system auditing is provided in this volume.
- The *Common Desktop Environment: Advanced User's and System Administrator's Guide* provides information on configuring the CDE environment and setting up system default resources such as printers.
- The ASU documentation kit provides the *Concepts and Planning Guide*, *Installation and Administration Guide*, and *Release Notes*.

7.1.3.2 Reference Pages

Reference pages provide a definitive list of all options and switches supported by commands. The following pages are referenced in this chapter:

- The command line utilities are documented in `useradd`(8), `usermod`(8), `userdel`(8), `groupadd`(8), `groupmod`(8), and `groupdel`(8).
- The SysMan utilities are documented in `sysman`(8) and `sysman_cli`(8).

- Invoking the Account Manager (`dxaccounts`) is documented in `dxaccounts`(8).
- The system files are documented in `passwd`(4), `group`(4), `shells`(4), and `default`(4).
- Individual commands are documented in `passwd`(1), `vipw`(8), `grpck`(8), and `pwck`(8).

7.1.3.3 Online Help

The SysMan Menu Accounts options and Account Manager (`dxaccounts`) each provide online help that describe all the options and define appropriate data entries.

Some command line routines also provide text help for the command syntax. This help is invoked with the `-h` or `-help` command flag.

7.1.4 Related Utilities

The resources in the following list are also useful when administering accounts. These commands and utilities may be useful in correcting system problems when the graphical user environments are unavailable, such as after a system crash, or if you have access to only a character-cell terminal.

`vipw`

The `vipw` utility allows you to invoke a text editor to edit the password file manually. Avoid editing system files manually if possible; use one of the available utilities instead. You can use the `vipw` utility to edit the local password database, but you cannot use it to edit the NIS database, or use it on systems that have enhanced security.

The `vipw` utility enables you to edit the `passwd` file and at the same time locks the file to prevent others from modifying it. It also verifies the consistency of the password entry for root and does not allow a corrupted root password to be entered into the `passwd` file. You can also use the `vipw` utility to patch a corrupted `passwd` file when in standalone mode.

See `vipw`(8) for more information.

`who`

Provides a list of currently logged in users. See `who`(1) for more information.

`finger`

Displays user information from the password file. See `finger`(1) for more information.

csh, ksh, and sh	The csh, ksh, and sh commands invoke and interpret the C, Korn, and POSIX shells.
grpck	The grpck command enables you to verify the integrity of the group file.
pwck	The pwck utility enables you to verify the integrity of the passwd file.
quotaon	The quotaon command enables you to turn quota information on and off.
passwd, chfn, and chsh	The passwd, chfn, and chsh commands allow users to change their password file information; the passwd allows a user to change his or her password, the chfn allows the user to change his or her full name; the chsh allows a user to change the login shell.

7.2 Account Administration - Quick Start

The following sections provide you with brief instructions on invoking the account administration utilities so that you can create basic accounts quickly. For example, if you have just installed and configured the system as the root user, you may want to set up a nonprivileged user account under your own name using the default account settings. At a later time you can read Section 7.3 and other sections to understand how to configure the system defaults and use the advanced features of account and group administration utilities.

7.2.1 Creating Primary Accounts During System Setup

On the first root login after a full installation of the operating system, the System Setup utility is displayed automatically to guide you through the options for configuring your system. The Account Manager (dxaccounts) icon included in System Setup enables you to configure initial accounts. This icon invokes an X11–compliant graphical user interface (GUI) that you can run under the Common Desktop Environment (CDE) or other X-windowing environments. See Section 7.5.2 for full information on using the Account Manager. When the Advanced Server for UNIX (ASU) is installed and configured, you can use the Account Manager (dxaccounts) GUI to administer Windows NT domain accounts as described in Section 7.6.

7.2.2 Using the Account Manager (dxaccounts) GUI

The Account Manager (`dxaccounts`) provides features supported by the CDE environment, such as drag-and-drop and cut-and-paste, to quickly clone new accounts from existing accounts. You can invoke this GUI as follows:

- Use the following command from a terminal to invoke the GUI in any X11–compliant windowing environment:

  # **dxaccounts**

- In CDE, open the Application Manager or the SysMan Applications pop-up menu from the Front Panel. Choose Daily Administration, and click on the Account Manager icon.

The Account Manager GUI (`dxaccounts`) also provides options for administering Windows NT domain users when ASU is installed. These options are dimmed on the window if ASU is not installed and configured.

You can use the Account Manager GUI (`dxaccounts`) to configure default options for user accounts, such as the shell and the parent directory. See Section 7.4.2.6 for information.

7.2.3 Using the SysMan Menu Accounts Option

The SysMan Menu Accounts options provide the same functions as `dxaccounts`, but with limited support for the following features:

- Managing Windows NT domain accounts for PC clients
- Managing accounts under Enhanced (C2) security

Invoke the SysMan Menu Accounts options from the CDE Applications Manager, the CDE Front Panel (SysMan Applications menu), or from the command line as follows:

sysman accounts

The Accounts options also let you add and modify accounts in NIS (Network Information Service) and Lightweight Directory Access Protocol (LDAP) environments. You can add local users to any system without adding them to the NIS environment. See the *Network Administration: Services* manual for information on NIS.

To use the Accounts options from the SysMan Menu, invoke the SysMan Menu as described in Chapter 1 and expand the options as follows:

1. Choose the Accounts option to expand the menu. The following menu options are displayed:
   - `Manage local users`
   - `Manage local groups`

- `Manage NIS users`
- `Manage NIS groups`
- `Manage LDAP users`
- `Manage LDAP groups`

2. Move the pointer (or use the Tab key) to choose an option. Click on mouse button 1 (MB1) or the Enter key to invoke the utility.
3. The first window (or screen) of the utility opens, presenting you with the following options:

   Add...

   Use this option to create a new user account.

   Modify...

   Use this option to modify account details for an existing user account.

   Delete...

   Use this option to remove a user's account, and optionally to delete all their system resources.

   Filter...

   Use this option to filter (search) for a specific user or set of users. You can specify different search criteria such as the user's UID or account comment.

   Options...

   Use this option to define the number of accounts at which filtering starts automatically. You can choose which user data is included in listings of user accounts.

Detailed use of these utilities is described in Section 7.4.1, and in the online help.

7.2.4 Using the Command Line Utilities

The following command line utilities are available for administering accounts and groups:

`useradd`, `usermod`, and `userdel`	Use these commands to add, modify, and delete user accounts, respectively.

`groupadd`, `groupmod`, and `groupdel`	Use these commands to add, modify, and delete groups, respectively.
`adduser` and `addgroup`	These utilities, documented in `adduser`(8) and `addgroup`(8) are obsolete interactive scripts provided only for backwards compatibility. If you are still using these scripts, you should migrate to one of the newer utilities that provide support for any work environment, including character-cell terminals and Windows NT.

The command line utilities also provide options for administering Windows NT domain accounts when ASU is installed.

7.2.5 Advanced Server for UNIX

Advanced Server for UNIX (ASU) is a layered application that implements Windows NT Version 4.0 server services and functions on a server running the UNIX operating system. To other computers running Windows, the UNIX system appears to be a Windows NT Version 4.0 server. Through ASU, you can share UNIX file systems and printers as shares. By default, the client Windows user must have both a Windows NT domain account and a UNIX account in order to share UNIX resources. When ASU is running, the UNIX account administrative utilities that are described in this chapter can be used to perform certain account administrative tasks, such as creating new accounts.

ASU software is located on the *Associated Products Volume 2 CD-ROM* and provides two free connects. See the *Installation and Administration Guide* provided in the software kit.

7.3 Understanding User Accounts and Groups

The administration of user accounts and groups involves managing the contents of the system's password and group files. On standalone systems, the files you manage are `/etc/passwd`, which is documented in `passwd`(1), and `/etc/group`, which is documented in `group`(4).

On networked systems, typically, the Network Information Service (NIS) or Lightweight Directory Access Protocol (LDAP) is used for central account and group management. NIS and LDAP allow participating systems to share a common set of password and group files. See the *Network Administration: Services* manual and **`www.OpenLDAP.org`** for more information.

If enhanced (C2) security is enabled on your system, you need to administer more than the `/etc/passwd` file for security. For example, the protected

password database is used for security related information such as minimum password lengths and password expiration times. These tasks are documented in the *Security Administration* manual.

7.3.1 System Files

The following system files may be updated when you perform account administration tasks and should be backed up regularly:

`/etc/group`

The `/etc/group` file contains group data. Each row specifies one of the following: the group name; optional encrypted password; numerical group ID; and a list of all users who are members of the secondary group. For example:

```
system:*:0:root luis
daemon:*:1:daemon
uucp:*:2:uucp
mem:*:3:
kmem:*:3:root
bin:*:4:bin,adm
sec:*:5:
cron:*:14:
.
.
.
users:*:15:billP carsonK raviL annieO
sysadmin:*:16:
tape:*:17:
.
.
.
```

`/etc/passwd`

The `/etc/passwd` file consists of rows of one record (row) per user, containing seven fields of user data. See Section 7.3.3 for more information. Example entries are:

```
carsonK:6xl6duyF4JaEI:200:15:Kit Carson,3x192,1-6942,
:/usr/users/carsonK:/bin/sh
annieO:.murv3n1pg2Dg:200:15:Annie Olsen,3x782,1-6982,
:/usr/users/annieO:/bin/sh
```

The example lines are broken to fit the page, and appear as a single line in the file.

`/usr/skel`

The `/usr/skel` directory contains skeleton files for new accounts such as a `.login` file. Users can edit these files to customize their account to the local environment, by defining environment variables and default paths to programs or project files. The `/etc/shells` file provides a list of available command shells on the system.

Log files

The log files `/var/adm/wtmp` and `/var/adm/utmp`, and log files in the `/usr/var/adm/syslog.dated` directory provide information about account usage.

If enhanced security is in use, the following security files are relevant:

- `/etc/auth/system/default`
- `/tcb/files/auth.db`
- `/var/tcb/files/auth.db`

If NIS (Network Information Services) is in use, the following NIS files are relevant. Be sure to back up these files on the NIS master database:

- `/var/yp/src/group`
- `/var/yp/src/passwd`
- `/var/yp/src/prpasswd`

LDAP information is stored in the LDAP database; this should be backed up.

The following log files provide information about account use:

- `/var/adm/wtmp`
- `/var/adm/utmp`
- The log files in the `/usr/var/adm/syslog.dated` directory

7.3.2 Understanding Identifiers (UIDs and GIDs)

Each user account is recognized by a unique number called a user identifier (UID). The system also recognizes each user group by a unique number called a group identifier (GID). The system uses these numbers to track user file access permissions and group privileges and to collect user accounting statistics and information.

The maximum number of UIDs and GIDs is 4,294,967,294 (32 bits with 2 reserved values). The maximum number of users that can be logged on is determined by the available system resources, but is of course a much smaller figure. If you intend to use the full range of UIDs and GIDs,

be aware that some older utilities and applications do not support the maximum number and you should take the following precautions:

- If you not running the latest versions of your end-user applications, ensure that they support maximum UIDs and GIDs. For example, the widely used Kerberos Version 4.0 does not support UIDs and GIDs beyond a certain range. If you currently use Kerberos Version 4.0, consider upgrading to Kerberos Version 5.0. Similarly, If you use PATHWORKS, consider upgrading to ASU Version 4.0 or higher.
- The System V file system (S5FS) does not support the maximum range of UIDs and GIDs. Any file system `syscall` that specifies UIDs and GIDs greater than 65,535 returns an `EINVAL` error. Users assigned a UID or GID greater than 65,535 cannot create or own files on a System V file system. Consider using the UFS or AdvFS as a solution.
- The behavior of certain commands and utilities change when the maximum UID and GID range is increased. Compare these changes against any local use of these commands, such as in shell scripts:
  - The `ls -l` command does not display the disk block usage on quota files or sparse files. To display the actual disk block usage for any file, use the `ls -s` command.
  - The `cp` command incorrectly copies quota files or other sparse files. To correctly copy quota files or other sparse files, use the `dd` command with the `conv=sparse` parameter:

    ```
    # dd conv=sparse if=inputfile  of= outputfile
    ```

  - If you back up a UFS file system that contains quota files or other sparse files using the `vdump` utility and restore it using the `vrestore` utility, the quota files or other sparse files are restored as follows:
    - ☐ The first page of a file on disk is restored as a fully populated page; that is, empty nonallocated disk blocks are zero filled.
    - ☐ Any additional pages on disk are restored sparse.

7.3.3 Understanding the Password File

The `passwd` file for a standalone system identifies each user (including root) on your system. Each `passwd` file entry is a single line that contains seven fields. The fields are separated by colons and the last field ends with a newline character. The syntax of each entry and the meaning of each field is as follows:

username:*password*:*user_id*:*group_id*:*user_info*:*login_directory*:*login_shell*

username	The name for the user account. The *username* must be unique and consist of from one to eight alphanumeric characters.

password	You cannot enter a password directly. Enter an asterisk (*) in the `passwd` field to disable a login to that account. An empty password field allows anyone who knows the login name to log in to your system as that user.
user_id	The UID for this account. This number must be unique for each user on the system. Reserve the UID 0 for root. Assign each UID in ascending order beginning with 100. Lower numbers are used for pseudousers such as `bin` or `daemon`. (See also the `/usr/include/limits.h` file).
group_id	The GID for this account, which is an integer. See the *Technical Overview* for information on the limit. Reserve the GID 0 for the `system` group. Be sure to define the GID in the `group` file.
user_info (or GECOS data)	This field contains additional user information such as the full user name, office address, telephone extension, and home phone. The `finger` command reads the information in the *user_info* field. Users can change the contents of their *user_info* field with the `chfn` command. See `finger`(1) and `chfn`(1) for more information.
login_directory	The absolute pathname of the directory where the user account is located immediately after login. The `login` program assigns this pathname to the HOME environment variable. Users can change the value of the HOME variable, but if a user changes the value, then the home directory and the login directory are two different directories. Create the login directory after adding a user account to the `passwd` file. Typically the user's name is used as the name of the login directory. See `chown`(1), `mkdir`(1), `chmod`(1), and `chgrp`(1) for more information on creating a login directory.
login_shell	The absolute pathname of the program that starts after the user logs in. If you leave this field empty, the Bourne shell `/bin/sh` starts. See `sh`(1b) for information on the Bourne shell. Users can change

their login shell by using the chsh command. See chsh(1) for more information.

In windowing (graphical) user environments, utilities such as Account Manager (dxaccounts) can be used to perform all the operations provided by commands such as passwd and mkdir.

You only can set default characteristics for new accounts in some graphical utilities, while the command line utilities enable full access to setting and changing the default characteristics. See Section 7.4.2.6 for an explanation of how to do this with Account Manager (dxaccounts).

When the /etc/passwd file is very large, a performance degradation can occur. If the number of passwd entries exceeds 30,000, mkpasswd sometimes fails to create a hashed (ndbm) database. Because the purpose of this database is to allow for efficient (fast) searches for password file information, failure to build it causes commands that rely on it to do a linear search of /etc/passwd. This results in a serious performance degradation for those commands.

If you use the mkpasswd -s option to avoid this type of failure, a potential database or binary compatibility problem may arise. If an application that accesses the password database created by mkpasswd is built statically (nonshared), that application cannot read from or write to the password database correctly. This causes the application to fail either by generating incorrect results or by possibly dumping core.

Any statically linked application can be affected if it directly or indirectly calls any of the libc ndbm routines documented in ndbm(3) and then accesses the password database. To remedy this situation, you must relink the application. To avoid this compatibility problem, do not use the mkpasswd -s option.

Note

In an NIS environment you can add a user account to either the local passwd file or the NIS distributed passwd file. Accounts added to the local passwd file are visible only to the system to which they are added. Accounts added to the NIS distributed passwd file are visible to all NIS clients that have access to the distributed file. See nis_manual_setup(7) for more information on adding users in a distributed environment.

Similarly, LDAP users are also global.

7.3.4 Understanding the Group File

All users are members of at least one group. The `group` file identifies the group name for a user. There are two primary reasons to group user accounts:

- Several users work together on the same files and directories; grouping these users together simplifies file and directory access.
- Only certain users are permitted access to system files or directories; grouping them together simplifies the identification of privileged users.

The `group` file is used for the following purposes:

- To assign a name to a group identification number used in the `passwd` file
- To allow users to be members of more than one group by adding the user account to the corresponding group entries

Each entry in the `group` file is a single line that contains four fields. The fields are separated by colons, and the last field ends with a newline character. The syntax of each entry and the meaning of each field is as follows:

groupname: *password*: *group_id*: *user1* [*user2*,..., *userN*]

groupname	The name of the group defined by this entry. The *groupname* consists of from one to eight alphanumeric characters and must be unique.
password	Place an asterisk (*) in this field. Entries for this field are currently ignored.
group_id	The group identification number (GID) for this group, which is an integer. See the *Technical Overview* for information on the limits. Reserve the GID 0 for the system. The GID must be unique.
user	The user account belonging to this group, identified by the user name defined in the `passwd` file. If more than one user belongs to the group, the user accounts are separated by commas. The last user account ends with a newline character. A user can be a member of more than one group.

There is a limit to the number of groups that a user can be in, as documented in `group`(4). The maximum line length is LINE_MAX as defined in the `/usr/include/limits.h` file. User accounts should be divided into a number of manageable groups.

You can set defaults for certain GID values using the graphical or command line utilities. See Section 7.4.2.6 for an explanation of how to do this with Account Manager GUI (`dxaccounts`).

7.4 Administering User Accounts

The following sections describe how to:

- Administer user accounts using the SysMan Menu options. This method also allows you to add users in NIS (Network Information Service) and Lightweight Directory Access Protocol (LDAP) environments. Invoking the SysMan Menu and selecting the Manage Local Users option is described in Section 7.2.3.
- Administer local and NIS users and associated Windows NT domain accounts using the Account Manager GUI (`dxaccounts`). Invoking the Account Manager GUI is described in Section 7.2.2.

The process for using the `useradd` command line utility is similar and is documented in the reference pages but does not support NIS accounts. See the *Network Administration: Services* manual for information on NIS. The SysMan Menu Accounts options can be used from a terminal, X11, or Java client.

Note

Avoid using `adduser` because it does not provide all the available options and is not sensitive to security settings. To preserve the integrity of system files, avoid using manual methods of adding user accounts.

7.4.1 Using the SysMan Menu Accounts Options

The following sections describe how you create new accounts using SysMan Menu options. The following tasks are described:

- Gathering account information (Section 7.4.1.1)
- Setting account options, which apply to Local, NIS, and LDAP accounts (Section 7.4.1.2)
- Using filter options, which apply to Local, NIS and LDAP accounts, for searching accounts (Section 7.4.1.3)
- Creating or modifying local user accounts (Section 7.4.1.4)
- Deleting local user accounts (Section 7.4.1.5)
- Creating or modifying LDAP and NIS user accounts (Section 7.4.1.6)
- Deleting LDAP and NIS user accounts (Section 7.4.1.7)

For information on how you use the keyboard to enter information into fields on SysMan Menu utilities, invoke the online help.

7.4.1.1 Gathering Account Information

To prepare for administering accounts, gather the information on the worksheet provided in Table 7–2. If enhanced security is in use, the data items must comply with the minimum requirements (such as password length). See the *Security Administration* manual for more information.

See Section 7.3.3 for an explanation of the `passwd` file data items.

Table 7–2: Account Administration Worksheet

Field	Description	Data Item
User Name*		
Comments (gecos)	Full name	
	Location	
	Telephone	
User ID (UID)*	Can be assigned automatically	
Password*	Use mixed case or alphanumeric	
Primary Group*	Can be assigned automatically	
Secondary Groups		
Shell	Can be chosen	
Home Directory*	Can be created automatically	
Lock Account		
Local User		
NIS User		
Windows User	Shares needed	

* denotes a mandatory field

An example of typical user data is provided in Table 7–3.

Table 7–3: Account Administration Worksheet with Example Data

Field	Description	Data Item
User Name*		carsonK
Comments (gecos)	Full name	Kit Carson
	Location	Office 3T-34
	Telephone	4-5132
User ID (UID)*	Can be assigned automatically	Use next available
Password*	Use mixed case or alphanumeric	Use site specific initial password
Primary Group*	Can be assigned Automatically	Users
Secondary Groups		marsx, 25
Shell	Can be chosen	ksh
Home Directory*	Can be created automatically	/usr/marsx/carsonK
Lock Account		no
Local User		no
NIS User		yes
Windows User		yes, share \\maul\astools

* denotes a mandatory field

7.4.1.2 Setting Filter and Display Options

Use SysMan Manage local users Options... to configure filtering (described in Section 7.4.1.3) and display options. To set options, invoke the SysMan Menu and choose the Manage Local Users option as described in Section 7.2.3.

When you select Options... the SysMan Account Management: Program Options window opens and you can configure the following settings. Some option names are truncated here and appear as a descriptive line in the window:

On startup....

> Use this option to set a trigger value for the filter feature. The default setting is 200 user accounts.
>
> This feature is useful if you have many hundreds or thousands of user accounts. The more accounts that you have on your system, the

longer it takes any SysMan Accounts task to find and display all the accounts. Setting a trigger value causes the SysMan Accounts task to default to enter a filter (search) mode on startup. This enables you to choose a specific account or group of accounts and to greatly reduce the search and display time.

For example, if you set a figure of 300 user accounts, SysMan Accounts defaults to filter mode only when you have more than 300 accounts.

UserName

This checkbox enables display of the user's account name in all account listings.

Userid (UID)

This checkbox enables display of the user identifier (UID) in all account listings.

Comments

This checkbox enables display of any account comments (such as location and telephone number) in all account listings.

Selecting checkboxes affects your filter options. You can filter accounts based only on the data displayed.

7.4.1.3 Using Filter Options

If you have a large number of accounts you can use the Filter... option to quickly find a particular account or group of accounts. You can invoke the filter automatically, depending on the settings in Options... (described in Section 7.4.1.2). Automatic invocation enables you to avoid a delay while the Account Manager finds and loads all the user account data. You can filter both local and NIS accounts using this feature.

To use the search and filter option, invoke the SysMan Menu and choose the Manage Local Users option as described in Section 7.2.3. Select Filter... to open a dialog window titled: Manage Local Users: Show. Using this window, you can perform simple and advanced searches.

To perform a simple search:

Enter a filter (a search string) or a set of filters. All simple searches are based on account names entered as follows:

- An individual user name such as `s_kahn`
- A wildcard pattern, such as `*khan` or `?_khan`

- A comma-separated list of user names or wildcard patterns, such as `*khan, kim, donny_w, tom*`

Any accounts matching the filter specification are listed in the Manage Local Users window, with the original filter string identified at the top of the window.

To perform an advanced search:

Select Advanced to display the additional filter options. Activate a search option by selecting the checkbox.

The filter options are:

User name or filter...

Enter a filter as described for the Simple Search option.

User ID range...

Enter either a restricted range of UIDs, such as `1-100`, or an open-ended range, such as `100-`, to find all accounts with a UID greater than 100, or `-100` to find all accounts with a UID less than 100.

Pattern in "comments"...

Enter a search pattern to search on data entered in the Comment (GECOS data) field when the user's account was created.

This may be a telephone number, a physical location, or other user-specific information. You can use the asterisk (`*`) or question mark (?) wildcards to define a pattern. For example; *`*string*`*, such as: `*Sub*`.

LOCKED or UNLOCKED search criteria...

This option enables you to include (or exclude) locked or unlocked accounts. You can use this option to identify all currently locked accounts.

A warning dialog box opens if you do not clear the contents of the Simple Search before invoking an Advanced Search. If you see this warning dialog box, select `OK` to accept the Advanced Search. This action supersedes any search criteria that you specified in the Simple Search.

7.4.1.4 Creating or Modifying Local Accounts

To create a new account, invoke the SysMan Menu and choose the Manage local users option as described in Section 7.2.3. A table listing all the existing local user accounts is displayed.

The online help provides explanations for the fields, and defines valid data.

Use the following procedure to add a local user:

1. Select the Add... option to open the Manage Local Users: Add a User dialog box.
2. Complete the data fields using the information from the worksheet in Table 7–2.
3. If additional NIS options are required, select Options.... The Options dialog box opens. Select the appropriate NIS option, then select OK to return to the Add a User window.
4. Select OK to add the new user. If you have made an error, such as a mistyped password confirmation, the utility prompts you to correct it.
5. The Local Users window opens, showing a confirmation message. Select OK to return to the SysMan Menu.

To modify an existing account, invoke the SysMan Menu and choose the Manage local users option as described in Section 7.2.3. The table of local users is displayed, listing all the existing local user accounts.

The online help provides explanations for the fields, and defines valid data.

Use the following procedure to modify a user entry:

1. Scroll through the list of users and select the entry you want to modify.
2. Select Modify... to open the Account Manager: Modify a User window.
3. Change the contents of data fields as needed.
4. If additional NIS options are required, select Options.... The Options dialog box opens. Select the appropriate NIS option, then select OK to return to the Modify a User window.

   To add or modify more than one account, select Apply instead of OK. All changes are deferred until you select OK to exit.
5. Select OK to confirm the changes. If you have made an error, such as a mistyped password confirmation, the utility prompts you to correct it. password confirmations.
6. The Manage Local Users window opens, showing a confirmation message. Select OK to return to the SysMan Menu.

7.4.1.5 Deleting Local Accounts

Before deleting accounts consider the following:

- As an alternative to deletion, you can use `Modify...` to lock an account. You can transfer the account to another new user using `Modify...` to change some account details.
- You can invoke the `dxarchiver` utility before deleting the account to create a compressed archive file of the user's directories and files. See `dxarchiver`(8) for more information.

To delete an account, choose the Manager Local Users option as described in Section 7.2.3. The table of local users lists all the existing accounts. Use the following process to delete a user:

1. Scroll through the list of users and select the user account that you want to delete.
2. Select `Delete...` to open the Account Manager: Delete a User dialog box.
3. Optionally, select Delete User's Directory and Files if you want to remove the user's resources and recover the disk space.
4. Select `OK` to delete the account. The list of local users is updated immediately.

7.4.1.6 Creating or Modifying LDAP and NIS Accounts

To create a new LDAP or NIS account, invoke the SysMan Menu and select the Manage NIS Users option or Manage LDAP Users option as described in Section 7.2.3. The LDAP or NIS Users table lists all the existing local user accounts.

Use the following procedure to create an account for a local user:

1. Select `Add...` to open the Manage LDAP or NIS Users: Add a User window.
2. Complete the data fields using the information from the worksheet described in Table 7–2.
3. Select `OK` to add the new user. If you have made an error, such as a mistyped password confirmation, the utility prompts you to correct it. password confirmations.
4. The Manager LDAP or NIS Users window opens, showing a message confirming the successful addition. Select `OK` to return to the SysMan Menu.

To modify an existing account, invoke the SysMan Menu and choose the `Manage LDAP or NIS Users` option as described in Section 7.2.3. The

LDAP or NIS Users table lists all the existing local user accounts. Use the following procedure to modify a user entry:

1. Scroll through the list of LDAP or NIS users and select the user account that you want to modify.
2. Select `Modify...` to open the Manage LDAP or NIS Users: Modify a User dialog box.
3. Change the contents of data fields as required.
4. Select `OK` to confirm the changes. If you have made an error, such as a mistyped password confirmation, the utility prompts you to correct it. password confirmations.

   To add more than one account, select `Apply` instead of `OK`. All changes are deferred until you select `OK` to exit.
5. The Local Users window opens with a message confirming the successful addition. Select `OK` to return to the SysMan Menu.

The online help provides explanations for the fields, and defines valid data.

7.4.1.7 Deleting LDAP and NIS Accounts

To delete LDAP or NIS accounts, choose the Manage LDAP or NIS Users option as described in Section 7.2.3. The LDAP or NIS Users table lists all the existing accounts.

Use the following process to delete a user:

1. Scroll through the list of users and select the account that you want to delete.
2. Select `Delete...` to open the Manage LDAP or NIS Users: Delete a User dialog box.
3. Optionally, select Delete User's Directory and Files if you want to remove the user's resources and recover the disk space.
4. Select `OK` to delete the account. The list of LDAP or NIS users is updated immediately.

7.4.2 Using Account Manager (dxaccounts)

Invoke the Account Manager GUI (`dxaccounts`) as described in the quick start instructions in Section 7.2.2. The Account Manager on <host> window opens first. Use the following procedure to administer accounts, using the data gathered in the Table 7–2 worksheet.

Use the following procedures to add, modify and delete accounts when using the Account Manager GUI. The processes are identical for administering

NIS users, except that you also must be authorized to make changes to the NIS databases. (See the *Network Administration: Services* manual for more information on NIS.)

Most options require root privilege because they affect the user account databases. Options that do not affect the databases are available to all users. An example of such an option is Find, which you use to locate accounts.

When ASU is installed, additional options are displayed in the `dxaccounts` windows that enable you to administer accounts in Windows NT domains and create associated UNIX accounts simultaneously. See the *Installation and Administration Guide* for more information on ASU.

If Enhanced (C2) security is enabled, additional options enable you to retire and disable accounts according to the security settings in force. See the *Security Administration* manual for more information.

7.4.2.1 Adding and Modifying Accounts

You use the Account Manager on <host> window to add or modify user accounts as follows:

- Select `Add` to create a new account.
- To modify an existing account, double click on the user's icon. If there are many accounts, you use the options described in Section 7.4.2.3 to find accounts.
- You can copy (clone) a new account from an existing account, as described in Section 7.4.2.4.

Use the following procedure to add or modify accounts:

1. If the current view is not Local Users, pull down the View menu and select the Local Users option.
2. Select `Add` to open the Add/Modify Local User dialog box; select `Add`.

   (To modify an existing account, double click on the user's icon.)
3. Enter the new user name in the Username field.
4. Either select the next available UID, or enter a new UID.

   If you modify a user's UID with Account Manager, the ownership of the user's files and subdirectories does not change and, under certain circumstances, the home directory ownership may not change either. For example, if you change the UID of user johndoe from 200 to 201, the files and subdirectories under his home directory still belong to UID 200. Furthermore, if johndoe does not own his home directory, the ownership of that directory does not change either. To avoid this problem, use the `chown` command to change the directory and files, if applicable.

5. Use the pull-down menu to choose the primary group, or clear the text field and type a group name.

   If secondary groups are required, select `Secondary Groups...`. In the Secondary Groups window, double click on any required local or NIS (if available) groups.

6. Select the preferred shell from the pull-down menu.
7. The home directory is created at the default location of `/usr/users/<username>`. Enter an alternative path if required.
8. Select `Password...` to enter an initial password. Use a mixed case or alphanumeric string of length determined by local security settings.
9. Enter any user information (GECOS field data) in the comments fields.
10. You can check the following boxes:

    Automatically create the home directory

    > This creates the directory with the correct ownership and protections.

    Lock the account

    > This prevents any logins until you clear the field.

11. Select `OK` to create the account and return to the Account Manager main window. If you have made an error, the utility prompts you to correct it. password confirmations.

    The Current View is updated with an icon for the new user.

7.4.2.2 Deleting Accounts

Invoke the `dxaccounts` utility as described in Section 7.2.2. The Account Manager on <host> window is displayed first.

1. Double click the icon of the account that you want to delete. If there are many accounts, use the options described in Section 7.4.2.3 to find accounts.
2. Select `Delete`. The Delete Local UNIX User window opens. You can remove the user's files and directories at this time. (You may want to archive these. See the `dxarchiver` option.)
3. Select `OK` to confirm the deletion and return to the Account Manager on <host> window. This window is updated immediately, removing the deleted user account.

7.4.2.3 Finding and Selecting Accounts

The dxaccounts utility provides a useful search feature that you can use to locate user accounts. You can use this feature to choose groups of users to which you want to apply global changes, such as modifying the user shell or password.

Invoke the dxaccounts utility as described in Section 7.2.2. The Account Manager on <host> window opens first.

1. Select Find.
2. Enter a search string (a text string) in one of the fields and select OK.

The Find option enables you to locate and display all accounts where the data in the search field contains the search string. For example:

1. Enter the string ad in the Username field then select OK.
2. The Selected Users window opens, with a list of users who matched the search criteria.

   The matched users include adm, admin, adamK, and wadmanB. These user accounts are highlighted in the Current View.

After you select a group of user accounts, you can choose the modify (or delete) option to perform global operations on the selected users.

7.4.2.4 Copying Accounts

You can use existing accounts as templates to create new accounts, enabling you to clone the account properties. You can create an exact duplicate of one or more accounts using the following procedure:

1. Select the icon for an existing user account to highlight it, or use the mouse to select a group of accounts.
2. Select OK to copy the account.
3. Select Paste to create a clone account. The new icon label has the original name, appended with the string _copy*n*, where *n* represents the sequential number of the copy. You can make as many copies as required.
4. Choose each duplicate account in turn to rename it and to modify its properties as described in Section 7.4.2.1.
5. Make the minimum required modifications to the account as follows:

   a. Enter the new user name

   b. Change the UID or choose the next available UID

   c. Change the password

6. Select OK to add the modified account and return to the Account Manager on <host> window. This window is updated immediately with an icon for the new account.

You can use the same procedure to clone groups.

When copying user accounts using cut and paste or drag and drop, the Allow Duplicate UIDs option in the General Preferences dialog box is honored. For example, when making a copy of a user account that has a UID of 200, if the Allow Duplicate UIDs check box is off (the default), a unique UID is generated automatically for the resulting copy. If the Allow Duplicate UIDs check box is on, then the copy has an identical UID. The same rule applies to copying groups.

Using MB1 to drag and drop user accounts, groups, or templates results in a copy operation, not a move operation. This is different from the default CDE behavior, where using MB1 performs a drag and drop move operation and Shift-MB1 performs a copy operation. For example, if you use MB1 to drag a user account from the Local Users view and drop it in the NIS Users view, you create a copy of that user account in NIS. To avoid this problem, delete the original icon after the copy is complete.

7.4.2.5 Using the Password Option

The dxaccounts utility provides a password option enabling you to change or remove passwords for a single user or a group of users. Use this option as follows:

1. Choose the user or users whose passwords you want to change. The Find option may be useful in selecting groups of users.
2. From the Edit menu, choose Password.
3. In the New Password window, enter and confirm the new password.

   Select No Password to remove the current passwords; there are important system security implications when you choose this option.
4. Select OK to confirm the change and return to the Account Manager main window.

7.4.2.6 Account Manager (dxaccounts) General Options

The Account Manager GUI (dxaccounts) enables you to set defaults easily for newly created user accounts. Also, you can set account defaults using the command line (useradd) but you cannot use SysMan Menu Accounts options to set defaults. Use the following procedure to add or modify defaults:

1. From the Options menu, select General.... The General Options window opens, enabling you to set the following defaults:

Duplicates Policy

These options enable you to allow duplicate User Identifiers (UID) and Group Identifiers (GID).

ID Ranges Policy

These options enable you to control the minimum, next, and maximum UID and GID.

Default Primary Group

This option enables you to set the default primary group to a group other than `users`.

Default Primary Group

This option enables you to set the default home directory to a location other than `/usr/users`.

Default Shell for User

This option enables you to set the default login shell.

Default Primary Group

This option enables you to set the default skeleton directory path to a location other than `/usr/skel`.

Use Hashed Password Database

This option forces the creation of a hashed (encrypted) password database.

Require Password For New Accounts

This option forces the entry of a password each time an account is created.

Synchronize UNIX and Windows NT domain accounts

This option forces the automatic creation of an account when the UNIX account is created.

2. After you make the required changes, select `OK` to update the defaults and return to the Account Manager main window.

7.5 Administering Groups

The following sections describe how to administer groups:

- Using these SysMan Menu Accounts options:

  Manage local groups

  Manage NIS groups

  Manage LDAP groups

- Using the Account Manager GUI (`dxaccounts`).

You also can use the `groupadd`, `groupmod`, and `groupdel` commands to administer groups. See the documentation specified in Section 7.1.3 for more information on command line options.

Note

Avoid using the `addgroup` utility as it does not provide all the available options and is not sensitive to security settings.

To preserve system file integrity, avoid using manual methods of adding user accounts.

7.5.1 Using the SysMan Menu Accounts Group Options

The following sections describe how to administer groups using SysMan Menu options. The following tasks are described in this section:

- Creating a new local, LDAP, or NIS group
- Modifying an existing local, LDAP, or NIS group
- Deleting a local, LDAP, or NIS group

For information on how to use the keyboard to enter information into fields on SysMan Menu screens, invoke the online help.

7.5.1.1 Gathering Group Information

To prepare for administering groups, gather the information in the worksheet provided in Table 7–4. If enhanced security is in use, the data items must comply with the minimum requirements. See the *Security Administration* manual for more information.

See Section 7.3.4 for an explanation of the `group` file data items. In the SysMan Menu options, you can specify default values for NIS groups. See the *Network Administration: Services* manual for information on configuring NIS.

In Table 7–4 the data items marked O are optional. You must specify at least one user account.

Table 7–4: Group Administration Worksheet

Field	Description	Data Item
Group Name*		
Password*	Not currently used.	
Group Identifier (GID)*	If unused, the next number is assigned automatically	
User*		
User		
User		
User		
User		
User		

* denotes a mandatory field

7.5.1.2 Creating or Modifying Groups

To create a new group, invoke the SysMan Menu and choose the Manage local groups option as described in Section 7.2.3. The Local Groups table is displayed, listing all the existing local groups. The process for adding NIS groups is identical, except that you choose the Manage NIS groups option.

Use the following procedure to create a group:

1. Select `Add...` to open the Add a Group dialog box.
2. Complete the data fields using the information from the worksheet in Table 7–4.
3. Optionally, in the Members panel, highlight the names of users who are the initial members of the new group.
4. Select `OK` to add the new user. If you have made an error, the utility prompts you to correct it.
5. The Local Groups table dialog box opens, with a message confirming the successful addition. Select `OK` to return to the SysMan Menu.

To modify an existing group, invoke the SysMan Menu and choose the Manage local groups option as described in Section 7.2.3. The Local Groups table is displayed, listing all the existing local groups. Use the following procedure to modify a group entry:

1. Scroll through the list of groups and choose the group that you want to modify.

2. Choose `Modify...` to open the Manage Local Groups: Modify a Group window.
3. Change the contents of data fields as required. For example, you can scroll through the list of users and add new users to the group.
4. Select `OK` to confirm the changes.

   To modify more than one group, select `Apply` instead of `OK`. All changes are deferred until you select `OK` to exit.
5. The Local Groups window opens, with a message confirming the successful modification. Select `OK` to return to the SysMan Menu.

Online help provides explanations for the fields, and defines valid data.

7.5.2 Using Account Manager (dxaccounts)

Invoke the Account Manager (`dxaccounts`) utility as described in Section 7.2.2. The Account Manager on <host> window opens first. Using the data from the worksheet in Table 7–4, use the procedures in the following sections to add, modify, and delete groups when using `dxaccounts`. The process for administering NIS groups is identical to the process for administering Local Groups, except that you must be authorized to change the NIS databases. You can still use any options, such as Find, that do not change the databases.

If there are many groups on your system, use the Find option described in Section 7.5.2.4 to locate groups that you want to modify or delete.

The Account Manager utility does not administer LDAP groups.

7.5.2.1 Adding Groups

Add a group as follows:

1. Pull down the View menu and choose the Local Groups option.
2. Select `Add` to open the Add/Modify Local UNIX group window.
3. Enter the new group name in the Name field.
4. Choose the next available GID or enter a new GID.
5. Double click on any user name to add that user to the group. This action is optional.
6. Select `OK` to add the group and return to the Account Manager on <host> window. This window is immediately updated with an icon for the new group.

An alternative method of adding a new group is to clone it from an existing group as follows:

1. Select an existing group icon to highlight it.
2. Select `Copy` to copy the group.
3. Select `Paste` to create a new version of the group. The new icon label has the original name, appended with the string `_copy`*n*, where *n* represents the sequential number of the copy. You can make as many copies as required.
4. Double click on the newly copied icon to highlight it and display the Add/Modify Local UNIX group window. `Modify` is selected automatically.
5. Make any required modifications to the group as follows:
   - Enter the new group name
   - Change the GID, or choose the next available GID
   - Add or delete members
6. Select `OK` to add the group and return to the Account Manager on <host> window. This window is updated immediately with an icon for the new group.

7.5.2.2 Modifying Groups

Invoke the `dxaccounts` utility as described in Section 7.2.2. The Account Manager on <host> window opens first. Use the following procedure to modify a group:

1. Double click on the group that you want to modify. The Add/Modify Local UNIX group window opens.
2. Make any required modifications to the group. For example:
   - Rename the group
   - Change the GID
   - Add or delete members
3. Select `OK` to confirm the changes and return to the Account Manager on <host> window. This window is updated immediately with any changes for the group.

7.5.2.3 Deleting Groups

Invoke the `dxaccounts` utility as described in Section 7.2.2. The Account Manager on <host> window opens first. Use the following procedure to delete a group:

1. Select the group that you want to delete.
2. Select `Delete`. The utility prompts you for a confirmation that you want to delete this group.
3. Select `Yes` to confirm the deletion and return to the Account Manager on <host> window. This window is updated immediately, removing the deleted group.

7.5.2.4 Finding Groups

The Account Manager utility (`dxaccounts`) enables you to locate groups and users who are members of groups.

Invoke the `dxaccounts` utility as described in Section 7.2.2. The Account Manager on <host> window opens first. To find a group:

1. Select `Find`.
2. Enter one of the following search strings:

   A group name or name fragment (text string)

   The Find option selects and displays all groups where the group name contains this string. For example, the string `mem` is matched to groups `mem` and `kmem`.

   A GID (integer)

   Any number entered is treated as a string. The Find option selects and displays all groups where the GID contains this string. For example, the string `20` is matched to groups `20` and `220`.

   A user name (text string)

   The Find option selects and displays all groups with users whose user name contains this string. For example, the string `wal` is matched to groups containing users named `wallyB` and `cadwalZ`.

7.6 Administering Windows Domain Accounts and Groups

When the Advanced Server for UNIX (ASU) is running, the account management utilities can be configured to support the creation and administration of Windows domain accounts. For information on installing and configuring ASU, see the ASU *Installation and Administration Guide*.

When ASU is installed, you can use the account management utilities to perform certain operations on associated (synchronized) accounts. These are accounts for the same user that exist both in the Windows domain and the UNIX environment, and are referred to as synchronized accounts in the UNIX utilities. For specific information on Windows 2000, see Section 7.6.2.

To configure a UNIX system to create associated Windows NT domain and UNIX accounts, and to set the default account creation options, you must set the account environment variables using the `usermod` (or `useradd`) command as shown in Example 7–1.

Note

When ASU is installed and configured, the creation of associated Windows NT domain and UNIX accounts is enabled by default. All account management utilities have their PC support features enabled. The value of the `Synchronized UNIX/PC Accts` environment variable is one (`1`), which indicates that the setting is on.

Example 7–1: Changing the Default Environment Variables Using usermod

```
# usermod -D 1

Local                          = 1
Distributed                    = 0
Minimum User ID                = 12
Next User ID                   = 200
Maximum User ID                = 4294967293
Duplicate User ID              = 0
Use Hashed Database            = 0
Max Groups Per User            = 32
Base Home Directory            = /usr/users 2
Administrative Lock            = 1
Primary Group                  = users
Skeleton Directory             = /usr/skel
Shell                          = /bin/sh
Synchronized UNIX/PC Accts     = 0
PC Minimum Password Length     = 0
PC Minimum Password Age        = 0
PC Maximum Password Age        = 42
PC Password Uniqueness         = 0
PC Force Logoff After          = Never

# usermod -D -x pc_synchronize=1 pc_passwd_uniqueness=1 \
pc_max_passwd_age=60 3

# usermod -D
```

Example 7–1: Changing the Default Environment Variables Using usermod (cont.)

```
.
.
.
Synchronized UNIX/PC Accts      = 1
PC Minimum Password Length      = 0
PC Minimum Password Age         = 0   [4]
PC Maximum Password Age         = 60
PC Password Uniqueness          = 1
PC Force Logoff After           = Never
```

[1] This command displays the current default environment variables.

[2] The output from the `usermod` command is a list of default values for the environment variables. When you create an account, these values are assigned to the new account. For example, all new accounts are created in the base home directory of `/usr/users`.

[3] This command specifies new default values for three environment variables that apply only to Windows NT domain accounts.

[4] This (truncated) list shows the new default values for the environment variables, which are as follows:

`pc_synchronize=1`

Creates associated Windows NT domain and UNIX accounts if ASU is running

`pc_passwd_uniqueness=1`

Forces validation of the password for uniqueness

`pc_max_passwd_age=60`

Specifies the maximum number of days that can elapse before a password must be changed by the user

Use the `groupmod -D` command to set the default environment variables for creating new groups. You can specify alternate values for the environment variables when you create a new account, overriding the defaults. See `useradd`(8), `usermod`(8), and `userdel`(8) for more information.

At the command line prompt, you can enter `-h` after each command to open a help screen showing the various command options. In ASU User Manager for Domains, you perform a similar task when you edit the default policy, which establishes similar default environment variables for newly created accounts.

You cannot use ASU account management utilities to perform operations on UNIX accounts only, or use UNIX utilities to perform operations on accounts that exist only in the Windows NT domain. The following sections provide information on how the UNIX and ASU account administration utilities behave when ASU is running and when you are administering synchronized accounts.

7.6.1 Administering Synchronized Accounts

If you have set up ASU and configured the creation of synchronized accounts, certain features in the account administration utilities are enabled automatically. The following sections describe how those features appear in the different account management utilities.

A lock file prevents you from using two different utilities (or two instances of the same utility) at the same time. This scenario easily could arise in large installations with many administrators managing many accounts. This lock file is at `/etc/.AM_is_running`. If the lock file exists, only one process can access the system files that relate to user and group data. If you attempt to invoke a second instance of any UNIX account management utility, an error message informs you that the data files are locked.

When using the ASU utilities to add accounts, ASU detects the presence of the lock file, and is unable to create an associated UNIX account. It only creates a Windows NT domain account. No lock file error message is displayed, and you do not receive a confirmation that the associated account was not created. When using ASU tools, verify the creation of an associated UNIX account by examining the contents of the `/etc/passwd` file.

7.6.1.1 Using SysMan Menu Accounts and Groups Options

The user interfaces for SysMan Menu Accounts utilities do not display any visual differences when ASU is running. If synchronized accounts are enabled, there are no differences in the windows and screens. However the following changes in behavior should be noted:

Add a user	You can choose from several `DOS----` groups when assigning the account holder to a group as part of account creation (the Primary Group option). If the creation of associated Windows NT domain accounts is enabled as described in Example 7–1, the associated account is created automatically and you cannot override its creation.
Delete a user	The associated Windows NT domain account is deleted automatically. You cannot override this

	deletion. If you want to retain the users' Windows NT domain account, do not perform this operation.
Add/Modify a group	Several `DOS----` groups are included in the selection list of groups, showing the default Windows NT domain accounts, such as `lanman` and `lmxadmin`.

See Chapter 1 for information on using the SysMan Menu.

7.6.1.2 Using Account Manager (dxaccounts)

The Account Manager utility (`dxaccounts`) is an X11–compliant GUI and as such can be used only in an X-window user environment such as CDE. The `dxaccounts` main window provides an option to create **PC** (Windows NT domain) accounts. This option is dimmed and unusable unless ASU is running. When ASU is running, the following features are available:

- When creating an account in one user environment, such as the Windows NT domain, you can choose to create a synchronized account in the other user environment, such as the UNIX environment.
- You can choose not to create an associated Windows NT domain account or UNIX account, even if creation is enabled by default as shown in Example 7–1.
- Additional options appear on the View menu, enabling you to open all Windows NT domain accounts and groups. When you choose these options, the PC (Windows NT domain) user and group accounts icons are displayed. You can add, modify, and delete PC accounts and groups as if they were UNIX accounts.
- From the Options menu, you can use the PC Defaults option to set characteristics that are inherited by any newly created account. You use the General Options menu item to set account synchronization and to set characteristics for UNIX accounts.
- When removing accounts with `Delete`, you are prompted to choose the UNIX account, the PC account, or both.
- When using the View menu, Local Groups option, the PC groups (`DOS----`) are visible and you can perform administrative tasks on these groups.
- When using the View menu, PC Groups option, the PC domain groups are visible and you can perform administrative tasks on these groups.

You use the processes described in Section 7.5.2 to perform administrative operations on PC accounts and groups.

The advantage of using `dxaccounts` is that it is a native X11 application and can use the features of the windowing environment, such as dragging

and dropping icons or cutting aand pasting icons, to clone new user accounts and groups from existing entities easily. However, unlike the portable SysMan Menu Account utilities, it runs only in an X-window user environment such as CDE.

The Account Manager utility does not administer LDAP groups.

7.6.1.3 Using Command Line Utilities

The command line utilities for administering user and group accounts are used to configure the default account characteristics also, as shown in Example 7–1. These characteristics are applied to all newly created accounts, and are referred to as the account **policy** in the ASU utilities. Unlike the graphical utilities, when using the commands you can choose to override the default environment variables and specify customized values for new accounts.

When ASU is installed, the following account and group creation options become available for use:

- `useradd`, `usermod` – The following extended options are provided to set the default Windows NT domain account characteristics using the `-D` option. Also shown are the default values:

  `pc_synchronize=` (value: 1, on)

  Use this option to determine whether synchronized accounts are created by default when a new account is created either for the Windows NT domain or on a UNIX system. Synchronized accounts are not created if this value is zero.

  `pc_min_password_age=` (value: 0, off)

  Use this option to specify how many days must elapse before a password can be changed. The user is not allowed to change passwords more frequently than this.

  `pc_max_password_age=` (value: 42 days)

  Use this option to specify how many days can elapse before a password must be changed. The user must change passwords at least this frequently.

  `pc_passwd_uniqueness=` (value: 0, off)

  Use this option to force verification of user-supplied passwords, ensuring that users do not reuse passwords.

`pc_force_logoff=` (value: Never, off)

Use this option to set up temporary accounts where the account holder is logged out automatically after a certain time when the account expires.

You invoke these extended options with the `-D -x` options, as shown in Example 7–1. To override the default characteristic, you specify the extended option with the `-x` flag during an account administration operation, such as account creation:

```
# useradd -x pc_passwd_uniqueness=1 guest9
```

The following command options are not extended options and do not set default account characteristics. These account characteristics can also be created using the ASU utilities. Use these command options when adding a new account:

- `pc_username=`*`name_string`*

  The user account name in the Windows NT domain. This can be identical to, or different from, the user's UNIX account name.

- `pc_unix_username=`*`login_name`*

  The synchronized UNIX account name. If no name is entered, it is the same as the Windows NT domain account name.

- `pc_fullname=`*`text_string`*

  The full name of the user or a description of the account.

- `pc_comment=`*`text_string`*

  A brief description of the account that can be changed only by the administrator.

- `pc_usercomment=`*`text_string`*

  A brief description of the account. This string can be changed by the user.

- `pc_homedir=`*`pathname`*

  The path to the user's home directory, specified as a Windows NT share format.

- `pc_primary_group=`*`group`*

  The primary group (Windows NT domain) to which the user belongs.

- `pc_secondary_groups=`*`group,group....`*

  The secondary Windows NT domains to which the user belongs. This value is specified as a comma-delimited list.

- `pc_logon_workstations=`*`client_name`*

A list of client host systems from which the user can log on. This value is specified as a comma-delimited list. A null value (" ") means that the user can log on from all workstations.

- `pc_logon_script=`*`pathname`*

  The directory where the default logon script is located. (This directory is created during ASU configuration.)

- `pc_account_type=`*`local|global`*

  Specifies whether the account is a local or global account in the Windows NT domain.

- `pc_account_expiration=`*`date_string`*

  Specifies the date on which the account expires and logins are prevented.

- `pc_logon_hours=`*`Dd0000-0000,Dd0000-0000....`*

  Specifies the days of the week and hours of the day during which logins expire and logins are permitted or denied.

- `pc_user_profile_path=`*`pathname`*

  Specifies the pathname to the default user profile directory.

- `pc_disable_account=`*`0|1`*

  Specifies whether the account is locked initially, disabling logins.

- `pc_passwd`*`0|1`*

  A text string used as the initial account password. You must precede this option with the `-x` flag and you are prompted to enter a password, and then confirm the entry. The password is not be echoed to the display.

- `pc_passwd_choose_own=`*`0|1`*

  Controls whether users can set their own passwords.

- `pc_passwd_change_required=`*`0|1`*

  Forces the user to change the password at the initial login.

- `userdel` – The only supported PC (Windows NT domain) option you can use with this command is `Synchronized UNIX/PC Accts`.

  Use this option to delete synchronized accounts, as follows:

  ```
  # userdel -r -x pc_synchronize=1 studentB
  ```

- `groupadd`, `groupmod`

  The following extended options can be used with the `-x` flag to administer groups in Windows NT domains:

  `pc_group_description=`*`string`*

  Specifies a text string that provides a description of the group.

pc_group_members=*user,user....*

Specifies a comma-delimited list of group members.

The advantage of using the command line is that it offers complete control over administrative tasks, enabling you to specify any and all command options and override the default account environment variables.

Commands can be used as part of a shell script to customize and automate account creation. However, the command options can be lengthy, so it is often easier to set up an account using the graphical utilities.

See useradd(8) and groupadd(8), and the related reference pages identified therein.

7.6.1.4 Using the ASU User Manager for Domains

ASU provides its own utility for administering Windows NT domains, domain user accounts, and groups. This application must be installed on and can only be used from a system running Windows NT. It provides the same features as the net command line options.

You can specify default environment variables for all newly created accounts. These environment variables are referred to as account policies in the Windows NT domain. You cannot set the default environment variables for synchronized UNIX accounts when using the User Manager for Domains (usrmgr.exe).

See the ASU *Installation and Administration Guide* and the User Manager for Domains online help for more information.

7.6.1.5 Using ASU net Commands

ASU provides an extensive set of net commands that you enter on the UNIX command line or from a DOS window on a Windows NT server.

For example, the following command displays the help for net user, the command you can use to add, modify, or delete user accounts:

```
# net help user | more

The syntax of this command is:

NET USER [username [password | \*] [options]]
          username [password | \*] /ADD [options]
          username [/DELETE]
.
.
.
# net user josef /add
```

Enter the following command to display a list of `net` command options:

```
# net help view
```

See the *Installation and Administration Guide* for more information on using `net` commands.

7.6.2 Windows 2000 Single Sign-On

If your local computing environment consists of UNIX servers and Windows 2000 client systems, and you have one or more domain controllers in the environment, you can configure the optional Windows 2000 Single Sign-On (SSO) software. The SSO software enables account holders in the Windows 2000 domain to access computing resources on the UNIX server without needing a separate UNIX account.

The SSO software modifies the Windows Active Directory and the associated Windows account management utilities. These modifications enable administrators in the Windows 2000 domain to record UNIX account information in the user's Windows 2000 account records. The UNIX server systems have secure access to the account holder's data and can read the account holder's UNIX login information, such as password or GID.

You can create SSO user groups using the same software and administrative tools.

7.6.2.1 Single Sign-On Installation Requirements

Configuration and use of this feature has the following installation prerequisites:

- You must have root access to the UNIX system and be an administrator of every Windows 2000 domain controller on which the SSO software is to be installed. You must run an installation procedure on the UNIX system and at least one domain controller.
- The UNIX system cannot be running C2 level security. See the *Security Administration* manual for more information on security levels.
- You need the *Associated Products Volume 2* CD-ROM on which you find the SSO software kit. The *Windows 2000 Single Sign-On Installation and Administration Guide* is included in the kit in the `/doc` directory.
- You need the following information:
  - The domain name, such as `sso.w2k.com`.
  - The domain controller host name, such as `w2kserv.sso.w2k.com`.
  - The account name and password of a privileged domain account. This account should belong to the Administrators group and hold administrative privileges, but should not be the main Administrator

account. If no such account exists, create one before starting the installation.

7.6.2.2 Installing the Single Sign-On Software

Install the software as follows:

1. Load the CD-ROM into the reader.
2. Create a mount point and mount the CD-ROM using commands similar to the following:

   ```
   # mkdir /apcd
   # mount -r /dev/disk/cdrom4c /apcd
   ```

3. Locate the installation kits and documentation as follows:

   ```
   # ls /apcd/Windows2000_SSO
   ```

4. Use the `setld` command to install the software subset named `W2KSS0100`. The configuration script, `/usr/sbin/w2ksetup`, runs automatically when the installation is complete. Complete the configuration as described in the *Windows 2000 Single Sign-On Installation and Administration Guide*.

7.6.2.3 UNIX Requirements for Creating Single Sign-On Accounts

The following requirements for UNIX account characteristics apply to SSO accounts:

- You can create SSO user accounts in the Windows 2000 user environment using a modified version of the standard Windows 2000 user management tools only. You cannot create SSO accounts using UNIX tools such as `dxaccounts` or `useradd`.

  You can upgrade existing Windows 2000 accounts to provide account holders with SSO privileges for UNIX resources.

- There are terminology differences between UNIX accounts and Windows 2000 accounts. For example, user account data that describe the characteristics of an account are referred to as properties in Windows 2000 and attributes in the UNIX operating system. In the UNIX environment, this information is called GECOS data. The data is used by certain UNIX commands and utilities to perform account operations or to identify users. See Section 7.3.3 and subsequent sections for a description of UNIX account attributes.

  Prepare the following account data for each user or group. If necessary, use the UNIX account management tools described in this chapter to ensure that the account data is of an appropriate format and is unique for each user:

Username

In Windows 2000, the *Username* is the user logon name. For SSO it must meet two requirements; length and uniqueness. This also applies to group names.

Windows 2000 can support very long user names although in practice most users prefer short adaptations of their name and initials, which are easier to remember and type. The maximum length of the user name is determined by the current restriction to eight characters in the UNIX environment.

The actual name can be as short as the user's initials but must be unique on both systems for every user. If a user with only a UNIX account has the user name `chs`, you cannot assign that name to an SSO account.

Password

Each user requires a password. You determine the length of the password by the current settings on the UNIX system. These settings can vary depending on the security mechanisms in force. See the *Security Administration* manual for more information.

UID and *GID*

Each account requires a unique identification integer called a UID and each group has a GID. See Section 7.3.2 and Section 7.3.4 for a description of these identifiers.

User Comment

This field enables you to enter a text description of the GECOS data for future reference.

Home Directory

In the UNIX environment, the user's home directory is synonymous with a disk share on Windows 2000 system. The home directory is a section of the `/usr` UNIX file system that is reserved for user accounts, typically using the user's account name in the path to the directory. For example, `/usr/staff/songch` or `/usr/users/chs`.

Shell

This is the user's default UNIX command environment that is invoked when the user logs on, such as the Bourne shell (`sh`) or Korn shell (`ksh`). See the `shells`(4) and Section 7.3.1 for more information.

7.6.2.4 Creating Single Sign-On Accounts and Groups

Using the information prepared in Section 7.6.2.3, create SSO accounts as follows:

1. Log in to your administrator's account on the Windows 2000 domain controller.
2. Invoke the Microsoft Management Console (MMC) interface and open the Active Directory Users and Computers Window.
3. Open the `Users` folder and either choose an existing user or open the Action menu and choose the New option then the User option.
4. Three dialog boxes open in succession. You are prompted to enter the following information for each new user account:
   - The user account details, such as name.
   - The initial password for the account and any password characteristics.
   - The UNIX account properties. Use the information identified in Section 7.6.2.3, such as the UID and GID.

To create an SSO group use the same procedure, selecting the New and Group menu options in step 3.

7.6.2.5 Single Sign-On System Files

When you install and configure the software, the following system files are created:

- The `ldapcd` daemon, which is the connection to the registry of account information on the domain server. If the daemon is killed or stopped accidentally, restart it using the following command:

  ```
  # /sbin/init/dldapw2k restart
  ```

- The `/etc/ldapcd.conf` configuration file, which contains settings for the `ldapcd` daemon.
- The `/etc/w2kusers.deny` configuration file, which forces UNIX authentication only for the named users.

See the file headers and the *Windows 2000 Single Sign-On Installation and Administration Guide* for more information on these files.

8

Administering the Print Services

This chapter describes how to gather information for, then set up and administer the print services. You can set up and administer the print services immediately after a new installation or an upgrade to a new version of the operating system, or you can wait until later. The files and utilities that you use to administer the print services are discussed in this chapter.

During initial configuration of your system after a full installation, you see a checklist titled *System Setup*. On this menu is an option for `Printer Configuration`, which runs the `printconfig` graphical user interface. See Section 8.3.2 for more information.

The following topics are discussed in this chapter:

- An overview of the administrative tasks, describing the different configuration methods and the setup utilities that you can use (Section 8.1)
- An overview of the information gathering required for printer configuration (Section 8.2)
- A description of how to use different utilities to set up physically-connected, remote, and networked printers, including descriptions of the `printconfig` and `lprsetup` utilities. (Section 8.3)
- Routine print system maintenance, such as adding and removing printers or controlling print jobs. (Section 8.4)
- Reference information on advanced topics such as the structure of system files, spooling, daemons, error reporting, and print filters (Section 8.5)
- Current restrictions on the use of certain print filters (Section 8.6)
- Information that enables you to test printers and resolve problems (Section 8.7)

8.1 Print Administrative Tasks

There are two categories of print administration tasks: setting up the print system and maintaining the print system:

Perform these tasks to set up the print system initially:

- Connect the printer to the system. This may be a physical connection or access through a network.

- Add information about the printer in the `/etc/printcap` file.
- Create the required device files and spooling directories.
- Start the print daemon, `lpd`.
- Optionally initiate printer accounting.
- Verify printer installation and perform a test printing.

After a printer is set up and running on your system, you need to:

- Perform routine managment tasks, including adding new printers and changing the characteristics of existing printers.
- Administer the print queues and files as needed.
- Control the daily operations and throughput of print jobs.

The tools that you use to perform these operations are described in Section 8.1.6.

8.1.1 Printer Connection Methods

Depending on your local system configuration, you have several methods for installing and connecting printers, for example:

Network	Connection to a shared network printer across a local area network (LAN) or a local area transport (LAT) connection or through TCP/IP.
Direct	Installation of a single physically-connected printer is the simplest installation. There are serial or parallel hardware ports at the rear of the system to which you connect a printer with a cable; see the printer documentation for a description of the hardware installation. Any user on the local system can access the printer.
Remote	Connection to a printer directly connected to another system on the network. The remote option requires that your system can access and use services on the system to which the printer is connected.
PC Network	Connection to Personal Computer (PC) print queues when using the Advanced Server for UNIX (ASU). This application is used to manage mixed environments incorporating PCs and UNIX systems. When ASU is installed, you have additional options to configure PC print queues and share printers between PC clients.

8.1.2 Printer Administration Methods

There are also several printer administration methods, each of which provides certain advantages.

8.1.2.1 Using the Printer Configuration utility (printconfig)

This utility features a graphical user interface, which is recommended if you are a first-time user and for setting up a system quickly. This utility is part of the standard set of system administrative tools. See Chapter 1 for general information on these tools.

When ASU is installed, `printconfig` also enables you to manage PC printers.

The operating system supplies drivers and configuration files a number of third-party printers. The `printconfig` utility automatically displays a list of all the supported printers and enables you to configure them quickly.

Note

There are restrictions on using `printconfig` with older `/etc/printcap` files. See `printcap`(4) for information.

8.1.2.2 Using the lprsetup utility

This command line utility that you run from a terminal provides backwards compatibility with previous releases.

The `lprsetup` utility performs the same tasks as `printconfig`, but does not support PC printers, even if ASU is installed. Invoking the command line option of `printconfig` initiates the `lprsetup` utility.

The `lprsetup` utility does not support the management of PC printers under Advanced Server for UNIX (ASU), although ASU itself offers features for PC queue management.

8.1.2.3 Manually editing system files

Experienced system administrators may want to manage printers by editing the `/etc/printcap` file directly; for example, you may want to clone a particular printer configuration across a number of systems or merge parts of one system's printcap file with another. The information in this chapter is useful for performing such tasks. It is recommended that you use `lprsetup` or the `printconfig` utility to manage individual print queue entries.

Note

The `/var/spool/lpd` file is a special link used in a TruCluster Server environment. It cannot be used as a spool directory. You must take care not to break this link when manually editing a file. See the section on CDSLs in Chapter 6.

8.1.3 Advanced Printing Software

The Advanced Printing Software is an optional subset on the Associated Products CD-ROM. For information about this software, see the *Installation Guide*. When the Advanced Printing Software is installed, you must configure a gateway, as described in Section 8.3.2, to run Advanced Printing Software and the print daemon `lpd` on the same system. You can run Advanced Printing Software with `lpd` disabled, in which case it receives all inbound remote print requests (on socket 515) instead of LPD. However, with this configuration, local `lpd` commands such as `lpr` do not work.

8.1.4 Related Documentation

Additional documentation on using printer configuration tools is available in manuals, reference pages, and online help.

8.1.4.1 Manuals

The following lists references to information on using printer configuration tools in the Tru64 UNIX operating system documentation set.

- The *Network Administration: Connections* manual provides information connections used by networked printers.
- The *Writing Software for the International Market* manual provides information on internationalization support for printers that offer local-language capabilities, such as support for Asian languages.
- The *Common Desktop Environment: User's Guide* and *Common Desktop Environment: Advanced User's and System Administrator's Guide* manuals provide information on setting up printer services in the Common Desktop Environment.

Other Tru64 UNIX documentation references include the *Installation and Administration Guide*, which describes ASU features for managing PC print queues.

The Advanced Printing documentation provides information about using the Advanced Printing Software, which is included as an optional component of the Tru64 UNIX operating system. The *Advanced Printing Software User Guide* describes how to submit jobs to your printer using the command line

interface. It also shows you how to set up your local print environment and monitor jobs you have submitted. This manual is also packaged with the software kit, as is the following companion documentation:

- *Advanced Printing Software System Administration and Operation Guide*
- *Advanced Printing Software Command Reference Guide*
- *Advanced Printing Software Release Notes*
- *Advanced Printing Software Installation Guide*

Also, be sure to see the printer manufacturer's documentation for information on installing the printer and for required software settings such as data communication (baud) rates. This is particularly important if you are attempting to configure a printer that is not in the list of supported devices. The printer documentation provides information that you may need to provide to the configuration utility to use any of the printer's special capabilities, such as tray selection. Usually, you see your printer included in the list of supported devices when you use an installation utility (or if you look in the `/usr/lbin/lprsetup` directory). If your printer is not defined by a file in that directory, and the manufacturer does not provide information for using the printer, use the generic settings provided by the configuration tools. Access to printer capabilities often is restricted when you use a generic configuration.

8.1.4.2 Reference Pages

The reference pages listed here provide further information regarding utilities, files, and daemons.

`lpc`(8)	Controls the operation of the line printer system. For each line printer configured in the `/etc/printcap` file, the `lpc` command may be used for disabling or enabling a printer; disabling or enabling the printer spooling queue; rearranging the order of jobs in a spooling queue or finding the status of printers, their associated spooling queues, and the printer daemons.
`ports`(7)	Contains information about the printer ports that you use to connect printers to a system, and how they map to printer device special file names in `/dev`.
`printconfig`(8), `lprsetup`(8)	Contain information about the configuration tools and their command line options.
`printcap`(4), `lprsetup.dat`(4)	Contain information about the system files in which printer configuration information is located.

`wwpsof`(8), `pcfof`(8)	Contain information about generic print filters. The `wwpsof`(8) reference page describes a generic internationalized print filter for PostScript printers, which enables you to support local-language PostScript printing. The `pcfof`(8) reference page describes a generic print filter for ANSI, PCL (Printer Control Language), and PostScript printers.
`lpd`(8)	Contains information about the print daemon.
`latcp`(8)	Contains information about the local area transport (LAT) control utility. This utility is used for adding services, such as print services, to a host and is only of interest if you are using networked printers.
`lpr`(1), `pr`(1), `lprm`(1), `lpq`(1), and `lpstat`(1)	Describe the commands used to print files. See the *Command and Shell User's Guide* for information on using these commands.
`lptest`(8)	Describes the printer test pattern utility.
`ppdof`(8)	Describes the text to PostScript print filter.
`services`(4)	Describes the format of the `/etc/services` file where you can define the service ports for TCP/IP printing.

8.1.4.3 Online Help

The Printer Configuration application (`printconfig`) features an online help volume that explains its use.

The `lprsetup` utility has command line help.

8.1.5 System Files

The following system files contain printer configuration information. Some files, such as `/var/spool`, are defaults or UNIX conventions. You can use your own preferred file names and locations.

`/etc/printcap`	Contains the data on configured printers.

`/usr/lbin/lprsetup`	The `/usr/lbin/lprsetup` directory contains a series of files (`*.lpd`), each of which contains the configuration data for a supported (known) printer. The name of each file corresponds to the printer name. This information is compiled into the `/etc/lprsetup.dat` file and transferred to the `/etc/printcap` file when a printer is installed and configured for use.
`/etc/lprsetup.dat`	The `/etc/lprsetup.dat` file is a compilation of the files in the `/usr/lbin/lprsetup` directory.
`/usr/spool`	This file is a symbolic link that points to `/var/spool` (see below). This symbolic link is used to satisfy the legacy programs such as `mail` and `uucp`.
`/var/spool`	This directory is the default directory where print jobs are stored temporarily during printing. The spool queue is identified by the `sd` entry for a device in `/etc/printcap`. Keep newly-created spool directories under `/var/spool`; `/var` is defined for variable data.
`/var/spool/lpd`	This file is a symbolic link that points to `/var/spool/cluster/members/{memb}/spool/lpd`. This is a member-specific spool directory on node of a cluster; a standalone system is considered as a single node cluster (member0). The lock file `lpd.lock` is stored in this directory.
`/usr/adm/lpd*err`	The `/usr/adm/lpd*err` files are the error log files for each installed printer. These are only created if error logging is enabled.
`/var/adm`	The `/var/adm` directory contains printer accounting files when accounting is enabled. These files have a file name format of `/var/adm/<`*`printer`*`>acct_sum`, where <*printer*> is the name that you assign to the printer during installation.

`/usr/lbin/lpd`	The `/usr/lbin/lpd` file is the line printer daemon. Configuration files are located in the `/var/spool/*` (or `/usr/spool/*`) directory.
`/dev`	The `/dev` directory contains the local UNIX socket `/dev/printer`. This socket is created by the `lpd` daemon and exists for as long as the parent `lpd` is running.

8.1.6 Related Utilities

The following utilities are also available for use in your printer environment:

`lpc`	The line printer control utility enables you to manage print queues and control access to printers; it also allows you to examine printer description files for potential configuration errors. See `lpc`(8) for information.
`pac`	The printer/plotter accounting utility formats the data from printer accounting log files and displays it or stores it in a text file. See `pac`(8) for more information. See `acct`(8) for information on accounting.
Print Manager GUI	The Print Manager graphical user interface, located under `Desktop_Apps` in the CDE Application Manager, enables you to perform the following tasks. See its online help for information on how to use these graphical interfaces to manage print queues, control access to printers, and customize your view of the printer data. These features are similar to features offered by the `lpc` utility and the `lpq` or `lpstat` commands, which you can run from the command line in a terminal.
Print Screen GUI	The Print Screen graphical user interface, located under `Desktop_Apps` in the CDE Application Manager, enables you to print all or a portion of the display, or save it to a file. See its online help for information.
CDE front panel's printer icon	The printer icon on the CDE front panel allows you to select printers and manage print queues

locally. You also can run `Printer Configuration` (`printconfig`) from CDE Application Manager – Configuration, in addition to invoking it from SysMan Menu or the SysMan Station. The latter user environments use graphical tools remotely or from a different workstation such as a PC or another UNIX system. See the CDE documentation for information on setting environment variables such as `LPDEST` to assign system default printers in CDE environments.

8.2 Gathering Information

Before adding a printer, you need to gather the information about the printer that you enter when using the `lprsetup` or `printconfig` utilities. The information required depends on whether the printer is remote, a direct connection, or a network connection using LAT or TCP/IP:

If your system is part of a network, contact your local network administrator or see the *Network Administration: Connections* and *Network Administration: Services* manuals for information required when adding or accessing a network printer.

8.2.1 Network and Direct Printer Connections

The following list identifies the information you need when installing a direct physical connection, or a network connection using LAT or TCP/IP.

Available Printer Types (supported printers)

Determine the printer type from `lprsetup.dat`(4) or by viewing the files in the `/usr/lbin/lprsetup` directory if you have added third party printers. Usually, the name embossed on the printer is similar, such as LN03, however the printer type for the DECLaser 5100 is `ln09`. The `printconfig` utility displays a list of supported devices, including PC printer options if ASU is installed.

Printer Aliases (alternative names)

You can assign one or more aliases for the printer. An alias is a name that you can use with printer commands. For example, if your local system is named `alfie2`, you can assign that as an alias and use that name when printing files as follows:

```
# lpr -Palfie2 prt_accounting.txt
```

Connection type

This depends on how the printer is connected to your system. It can be:

TCP — A networked print server device.

Direct — Connected to a port at the rear of the system box.

LAT — Connected as a Local Area Transport port or service. (See the *Network Administration: Connections* manual for more information.)

Device Pathname

This depends on the Connection type:

TCP — For the TCP/IP connection type, you need the TCP address, in one of two formats:

`@node/port`

The printer host (or node) name followed by either the port number such as 9100 for an LN32, or the service name defined for this port in the `/etc/services` file. If no service name is defined, you must use the port number. For example `@alfie.nic.ccc.com/ln32port` maps to the entry `ln32port 9100/tcp` in `/etc/services`. The entry `@alfie.nic.ccc.com/9100` directly specifies a port number, and no service entry in `/etc/services` is required.

`@tcp_address/port`

The TCP/IP address in `nnn.nnn.nnn.nnn` format, followed by the port number (such as 9100 for an LN32) or the service name defined in `/etc/services`. For example `@123.321.123.321/9100`.

Direct — For the Direct connection type, this entry specifies the device file name in the `/dev` directory. For example, if you connected the printer cable to the 9–pin socket labeled 1 or COMM1, the corresponding device special file is `/dev/tty00`.

LAT — For the LAT connection type, `printconfig` supplies a default `/dev/lat` port or service name, LAT server

node name, or LAT port name. (See the *Network Administration: Connections* manual for more information.)

Advanced Options

Most of these options are not required to complete a basic installation and you can accept the defaults. However, the owner's manual for your printer may state requirements for certain settings, such as the communications baud rate. Communications rates can depend on features such as the length and type of printer cable that you chose for the installation.

The advanced options are set as symbols in the `/etc/printcap` file and `printcap`(4) contains a definitive list of supported symbols and values. When using `printconfig`, the online help provides a description of the symbols.

This screen scrolls down to list all available options and also contains default entries, which you can override if required.

The most commonly used options are:

Accounting file name	Choose the default, or enter your preference, if you want to use printer accounting to track print consumables.
stty baud rate (hard-wired ports only)	If your printer specifies communication rate requirements you can enter it here. You can sometimes increase the default rate to improve printer throughput.
Restrictions on use	Set the restrictions here if you want to control the volume and quantity of print jobs.
Default page layouts	If you want default values for certain page layout characteristics, set the characteristics here. Supported layouts may depend on printer restrictions and capabilities. See the owner's manual for the printer.
Destination directories and files	Specify the locations here if you want certain directory and file locations for print spooling or for error output.

8.2.2 Remote Printers

The following is a list of the information you need when installing a remote printer:

Printer aliases (alternative names)	You can assign one or more aliases for the printer. An alias is a name that you can use with printer commands. For example, if the remote system is named `alfabet`, you can assign that name as an alias and use that name when printing files as follows: `# lpr -Palfabet prt_accounting.txt`
Remote system name	The host name for the remote system, such as `alfabet.ccc.nic.com`
Remote printer name	The name of the printer on remote system, such as `lp0`, or a valid alias.
Advanced options	These options are not required for a basic installation. Remote printing allows you to configure a few advanced options (or deconfigured, such as error logging). See Section 8.2.1 for more information. Most advanced options are not passed on to the remote system.

The `Generic_Remote_LPD` is the printer type that needs to be selected for remote printing setup.

8.3 Configuring Printers

The following sections describe the information you need in order to use the `printconfig` utility to connect a printer to your computer; there is also a section that describes how to use the `lprsetup` utility to configure a printer.

Before proceeding, verify that the printer is connected to your system physically, accessible on the network (for remote printing), and functions as described in the owner's manual. A good strategy for avoiding installation problems is to accept the default data presented by the configuration utility. After you have the printer working, you can read about the advanced options and use the same utility to tune your configuration, as required.

You must have the Printer Support Environment subset installed. To see if you have this subset installed, enter:

```
# setld -i | grep OSFPRINT
```

If the OSFPRINT subset is installed, the following information is displayed:

```
OSFPRINT540     installed       Local Printer Support (Printing Environment)
```

If the OSFPRINT subset is not installed, see the *Installation Guide* for information on adding this, or any, subset with the setld utility.

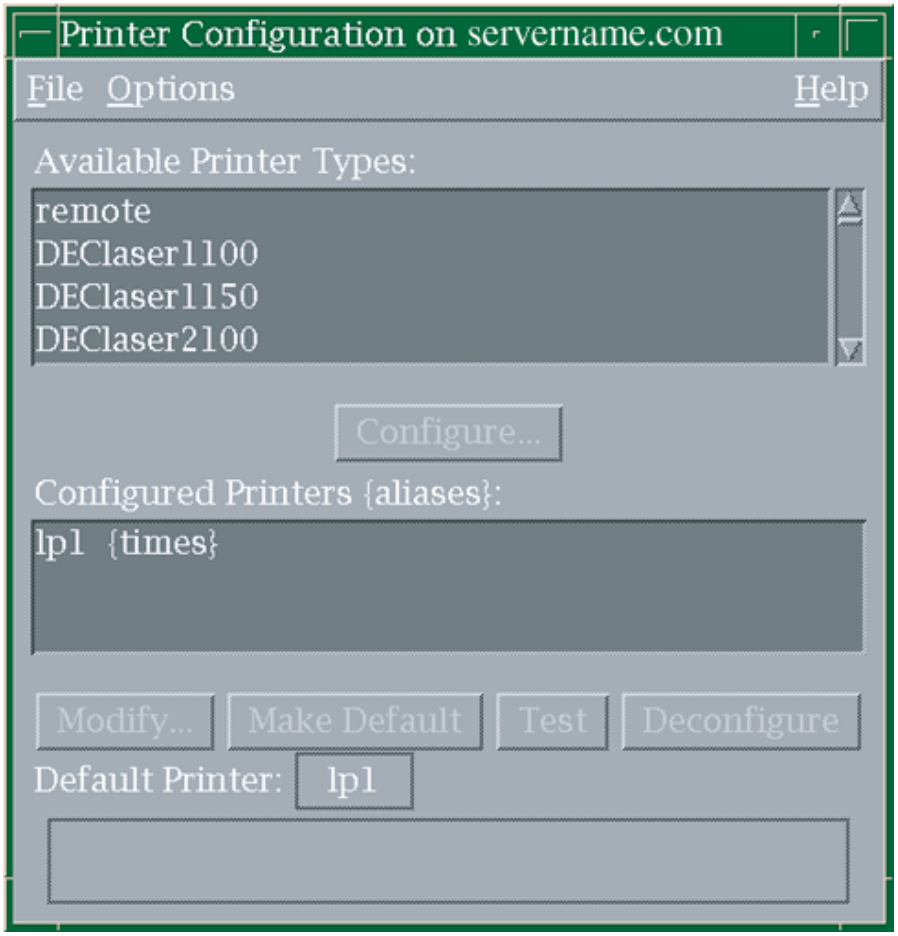
The following lists the printer configurations discussed in this section:

- Using printconfig to Configure TCP/IP Printing (Section 8.3.1)
- Installing a Physically-Connected Printer with printconfig (Section 8.3.2)
- Setting Up Remote Printers with printconfig (Section 8.3.3)
- Configuring PC Print Queues with printconfig (Section 8.3.4)
- Using lprsetup to Install a Printer (Section 8.3.5)

You can invoke printconfig from CDE, or from the command line; see printconfig(8) for more information.The Printer Configuration main window, as illustrated in Figure 1–1, appears; the first display is the main window titled Printer Configuration on *host name*.

Figure 8–1 shows the Printer Configuration (printconfig) main window.

Figure 8–1: Printconfig Main Window



8.3.1 Using printconfig to Configure TCP/IP Printing

TCP/IP printing allows you to submit print jobs to a printer that is directly connected to the network. To use this feature, your printer must contain a network interface card or be connected to a print server or terminal server, and you must register it with a TCP/IP node name and node address.

With TCP/IP printing, the local host manages print jobs in the same way as it does for a directly connected printer. The only difference is that with TCP/IP printing, the local print daemon (`lpd`) communicates with the remote printer over TCP/IP (similar to LAT printing). Each printer listens for connection requests on a socket number that is specified in the network interface hardware or that is user-defined through the printer console.

Although multiple hosts can talk to a single printer connected to the network in this way, the hosts are handled on a first-come, first-served basis. Therefore, TCP/IP printing is not the same as remote printing, in which the remote host or printer manages a print queue on the remote site and listens for network connections on socket 515 (as specified in the entry for `printer` in `/etc/services`).

8.3.1.1 Using printconfig for TCP/IP Printer Configuration

TCP/IP printing allows you to submit print jobs to a printer that is directly connected to the network as a host device. See Section 8.3.1 for information on TCP/IP printing. Gather the information as described in Section 8.3.

Invoke `printconfig` as described previously.

In this window you use the data you have gathered to select the printer type, for example:

- Select `Compaq LN32`
- Choose the Configure option

Because the printer is treated as a physically—connected printer, the next window is Printer Config: Local Printer Settings. The next available printer name is displayed, such as `lp4`. Complete the fields as follows:

- Printer Aliases, such as the name of the local host or perhaps something to help users to identify the printer type and physical location, such as `LN32_office23`.

  You also can specify this value as `@`*`nodename`*`/`*`servicename`*, where *`servicename`* is defined in `/etc/services` and associated with the printer's TCP/IP port.
- The Connection type, which is TCP.
- Finally, you need to specify the Device pathname as a port number or service, which in this case is the network address of the printer, such as `@alfie.nic.ccc.com/9100` or `@123.321.123.321/9100`.
- Choose Commit to write the options to the `/etc/printcap` file.

For a basic TCP/IP printer configuration, that is all you need to do. If you decide to use advanced options, such as setting print job limits, choose the Advanced option to display the "Printer Config: Local Printer Settings: Advanced" window.

After committing the configuration, you are returned to the Configuration on *host name* window, and the printer now appears in the list of configured printers. Use the Test option to print a test page to the printer. If you do not get any output, review the data carefully and refer to the troubleshooting section.

It may be necessary to configure the printer for its TCP/IP address. This may involve making entries on a hardware console panel or using `telnet` or a web browser to communicate with the printer. See your printer documentation for more information.

The remaining option on the window Configuration on *host name* is Make Default, which enables you to choose any configured printer as the default printer for this system. Printing a job without specifying the print queue causes the job to be printed on this printer.

Select the required printer and choose Make Default. The current default printer is displayed in the field labeled Default Printer.

8.3.1.2 Additional Manual Steps Required for Setting Up TCP/IP

Use the following steps, which describe how to set up TCP/IP printing on a local host, in addition to the information in the previous section.

1. Set up the printer. Assign a TCP/IP address and node name to each printer with a network card. Also, determine the TCP/IP socket number on which the printer listens for connection requests. You can either specify a name that is defined in the `/etc/services` file, or directly specify the port number assigned to the printer. If you want to create a service name, you need the socket number in Step 2b when you edit the `/etc/services` file.

   Table 8–1 lists the socket numbers for five Compaq printers, one Lexmark printer, and one Hewlett-Packard printer.

   Table 8–1: TCP/IP Socket Numbers

Printer	Socket Number
DEClaser 3500 (LN14)	10001
DEClaser 5100 (LN09)	10001
HP Printers	9100
Digital_LN17	2501
Lexmark Printers	9100
Compaq LN16	9100
Compaq LN32	9100

   To obtain the socket number for other printers, see your printer documentation under the network card. Some printers may allow you to specify this number yourself.

2. Configure the local host. This step describes the utilities that you need to run and the files that you need to modify on the local host in order

to configure TCP/IP printing. You must have superuser privileges to perform the following tasks:

a. Assign the following values to the `ct` and `lp` variables:

```
ct=tcp
lp=@nodename/servicename
```

Replace *nodename* with the name of the printer's node as registered for use on your network. Replace *servicename* with either the name that you choose to enter in the `/etc/services` database in the next step or the port number (for example, `lp=myHPLaserjet/9100`) . If you want to modify an existing `/etc/printcap` printer entry to use TCP/IP printing, edit the `/etc/printcap` file and modify the values for the `ct` and `lp` variables.

You also can remove the values for the `xs`, `xc`, `fs`, and `fc` control variables. These variables establish settings that are relevant to the serial port driver but are ignored by the network socket driver.

b. Configure the services database. You must register a service name and `tcp` port number (socket number) in the `/etc/services` database file. Enter the socket number that you determined when you configured the printer in step 1 and associate it with a service name of your choice. For example, to configure the services database for a DEClaser 3500, you add the following line to the `/etc/services` file:

```
declaser3500    10001/tcp
```

The user-defined string `declaser3500` represents the service; it is the same string that you entered as the *servicename* in the `/etc/printcap` file in step 2a. Save the changes to the `/etc/services` file.

Do not modify the `/etc/services` file if you provided the port number in step 2a.

c. Configure the remote hosts database. The *nodename* value that you specified as part of the `lp` variable value in the `/etc/printcap` file must be known by your local host's network management services; therefore, you must enter the *nodename* and its network address in the `/etc/hosts` database file. If you are running a BIND server for remote host names, you do not necessarily need to add the printer's node name to the `/etc/hosts` file, though if there is ever a problem with the BIND server, an entry in `/etc/hosts` is a useful fallback.

After configuration, TCP/IP printing is used like local and remote printing. From the command line, execute the `lpr` command specifying

the node name of the printer, command options, and file names. You also can view the printer status and submit print jobs with the CDE print utilities.

8.3.2 Installing a Directly Connected Printer with printconfig

This section describes how to install a directly connected printer using the `printconfig` utility. The example given is a DEClaser 5100 printer installed using the graphical user interface. It assumes that you have made all the physical connections and gathered the required information. You also can use `printconfig` to modify a printer configuration or remove a printer. These other tasks are described in Section 8.4.

Note

Do not use `printconfig` if you are modifying an `/etc/printcap` file on a system running Version 3.2 or older. There are incompatibilities in older `/etc/printcap` files that may cause `printconfig` to corrupt the file. Use `lprsetup` instead.

A typical installation of a printer (after the hardware is installed) takes about ten to fifteen minutes, including time required to gather the data.

The recommended action is to accept the default values for an initial printer installation. Then you can use `printconfig` to modify the configuration later if required.

Invoke `printconfig` as described previously.

You must have superuser privileges to run the `printconfig` utility. Depending on the type of printer you are adding and the information you provide, the utility may do the following:

- Create, or edit the existing `/etc/printcap` file
- Create a spooling directory
- Create an error log file
- Create an accounting file
- Create the device special files
- Prompt you to modify previously selected symbols

When you run the `printconfig` utility, the first display is the main window titled `Printer Configuration` on *host name*. In this window you select the printer type using the data already gathered as described in Section 8.2:

- Select `Digital_DEClaser_5100`
- Choose the Configure option

Because this is not a remote printer, the next window is `Printer Config: Local Printer Settings`. The next available Printer name is displayed (`lp0` if this is the first printer that you are configuring on this system). In this window you enter:

- The printer alias names, such as the name of the local host or perhaps something to help users to identify the printer type, such as `local_DL5100`.
- The connection type. Because you have connected the cable to a local serial port such as `COMM1`, you must choose the Direct option.
- Finally, you need to specify the Device pathname, which is the special device file that maps to the serial port. The device file `/dev/tty00` maps to the `COMM1` port.
- Choose Commit to write the options to the `/etc/printcap` file.

For a basic printer configuration, that is all you need to do. If you decide to use advanced options, such as setting print job limits, choose the Advanced option to display the "Printer Config: Local Printer Settings: Advanced" window. See Section 8.5.

After committing the configuration, you are returned to the `Printer Configuration on` *host name* window, and the printer now appears in the list of configured printers. Use the Test option to print a test page to the printer. If you do not get any output, review the data carefully and see Section 8.7.

The remaining option on the `Printer Configuration on` *host name* window is Make Default, which enables you to choose any configured printer as the default printer for this system. Printing a job without specifying the print queue causes the job to be printed on this printer.

Select the required printer and choose Make Default. The current default printer is displayed in the field labeled Default Printer.

Your printer is now ready for use. Test the printer's capabilities with appropriate files, such as PostScript or color graphics files. The printer utilities described in Section 8.1.6 can verify printer and queue status.

8.3.3 Setting Up Remote Printers with printconfig

A remote printer refers to a printer that is directly connected to a remote host or is otherwise treated as local by the remote host. You can connect remote printers directly to the network if their network cards emulate the remote `lpd` protocol, so they appear as remote hosts with a printer attached.

You configure your local print queue so that print jobs are sent to the remote host over the network. These jobs are then printed on the remote host. If

you are setting up a remote printer from a remote system, list the local system (the client) in the `hosts.lpd` file or `hosts.equiv` file of the remote system (the host).

See Section 8.2 for information on the data that you need to gather before performing this task, then invoke `printconfig` as described previously (the remote queue is the same queue as the example created in that section). The following illustrates how to use `printconfig` to create some remote print configurations:

1. Select `remote`.
2. Choose the Configure option.

Note

Because this is a remote printer, the next window is `Printer Config: Remote Printer Settings`. The next available printer name is displayed (`lp0` if this is the first printer that you are configuring on this system).

3. Enter the printer alias names, such as the name of the remote host and printer type.
4. Enter the remote system name.
5. Enter the remote printer name. For example in Section 8.3.2, a printer `lp0` was added to the system.
6. Choose Commit to write the options to the `/etc/printcap` file.

For a basic printer configuration, that is all you need to do. Choose the Advanced option to display the "Printer Config: Local Printer Settings: Advanced" window if you decide to use advanced options, such as setting print job limits. Because the printer is remotely configured, you can specify only a small number of advanced options that affect local processing, such as the local error log file and spooling directory.

After committing the configuration, you are returned to the `Printer Configuration on` *host name* window, and the printer now appears in the list of configured printers. Use the `Test` option to print a test page to the printer. If you do not get any output, review the data carefully and see Section 8.7.

8.3.4 Configuring PC Print Queues with printconfig

If the Advanced Server for UNIX (ASU) is installed and running, you can configure client PC printer queues. ASU also offers features for configuring and managing print queues. See *Installation and Administration Guide*

for more information on using ASU. You must have at least one printer configured in your /etc/printcap file before you can create a printer share queue for PC clients.

Invoke printconfig as described previously. When you invoke printconfig under ASU, an available printer type is the Advanced_Server_Shared_Printer_Queue. Choose this option and the next window displayed is titled "Printer Config: Advanced Server Shared Print Queue Setting". There are only three options on this window:

Advanced Server shared print queue name

Enter a queue name such as psq1

Printer devices

Enter a comma-separated list of device names such as: lp0,lp2

Comment

Enter any comments or notes on queue use and restrictions.

Select OK to create the queue and return to the printconfig main window. The new queue is displayed.

To test the status of the queue, use the following ASU command and examine the output for the queue name as shown in the example:

```
# net share
.
.
.
Share name  Resource             Remark
----------  ------------------   ------------------------
psq1        lp0, lp2             Spooled
```

You may need to perform other ASU tasks to make the queue available to PC systems. See the ASU documentation for more information.

8.3.5 Using lprsetup to Install a Printer

Use the lprsetup utility to install a printer locally (directly connected to your computer) using the lprsetup utility. You also can use lprsetup to modify a printer's configuration or remove a printer. These other tasks are described in Section 8.4.

The recommended action is to accept the default values for an initial printer installation.

The printer described in the following example is an LN03R.

You can run `lprsetup` by entering `/usr/sbin/lprsetup` at the command prompt in a terminal window. You must have superuser privileges to run `lprsetup`. Depending on the type of printer you are adding and the information you provide, you can use `lprsetup` to:

- Create, or edit the existing `/etc/printcap` file
- Create a spooling directory
- Create an error log file
- Create an accounting file
- Create the device special files
- Prompt you to modify previously selected symbols

When you run the `lprsetup` script, the first display is the main menu:

```
# /usr/sbin/lprsetup
Tru64 UNIX Printer Setup Program

Command < add modify delete exit view quit help >:
```

Table 8–2 lists the `lprsetup` command options.

Table 8–2: lprsetup Options

Command	Description
`add`	Adds a printer
`modify`	Modifies an existing printer's characteristics
`delete`	Removes an existing printer from your configuration
`exit`	Exits from the `lprsetup` program
`view`	Displays the current `/etc/printcap` file entry for the printer you are configuring
`quit`	Exits from the `lprsetup` program
`help`	Displays online help about the `lprsetup` program

You can abbreviate any command option with its initial letter.

You can enter information at each prompt or press Return to select the default information provided. (In most instances, you can accept the defaults.) You also can enter a question mark (?) to get a description of the information you specify at the prompt.

Note

Some of the symbols displayed by `lprsetup` are not supported by the operating system. See `printcap`(4) for information on the supported symbols.

The following example shows how to use the `lprsetup` command to set up an LN03R printer on the local system. Some tables are truncated to shorten the example:

```
# /usr/sbin/lprsetup
Tru64 UNIX Printer Setup Program
Command  < add modify delete exit view quit help >: add
Adding printer entry, type '?' for help.
Enter printer name to add [lp1] :  Enter

Printer Types:

  1. Compaq Advanced Server ClientPS
  2. Compaq Advanced Server ClientText
  3. Compaq LN16
  4. Compaq LN32
  5. Digital Colormate PS
  6. Digital DEClaser 1100
  7. Digital DEClaser 1150
  8. Digital DEClaser 2100
  9. Digital DEClaser 2150
 10. Digital DEClaser 2200
 11. Digital DEClaser 2250
 12. Digital DEClaser 2300
 13. Digital DEClaser 2400
 14. Digital DEClaser 3200
 15. Digital DEClaser 3250
 16. Digital DEClaser 3500
 17. Digital DEClaser 5100
 18. Digital LA100
 19. Digital LA120
 20. Digital LA210
 21. Digital LA280
 22. Digital LA30N

Press 'ENTER' to continue scrolling, type '(q)uit' to end scrolling:  Enter
 23. Digital LA30N A4
(and so on until)
 59. Digital LN03
 60. Digital LN03R
 61. Digital LN03S
 62. Digital LN03S-JA
 63. Digital LN15
 64. Digital LN15 A4
 65. Digital LN17
 66. Digital LN17 A4

Press 'ENTER' to continue scrolling, type '(q)uit' to end scrolling: q

Help Types:

  ?         - General help
  printer?  - Specific printer type information

Enter index number, help type, '(q)uit', or 'ENTER' [Generic Unknown type]: 60
```

```
You chose printer type 'Digital LN03R'.
Is that correct? [y]:y

Enter printer synonym: marks
Enter printer synonym:  Enter

Set device pathname 'lp' [] ? /dev/tty01

Do you want to capture print job accounting data ([y]|n)? y

Set accounting file 'af' [/usr/adm/lp1acct]? Enter

Set spooler directory 'sd' [/usr/spool/lpd1] ?  Enter

Set printer error log file 'lf' [/usr/adm/lp1err] ?  Enter

Set printer connection type 'ct' [dev] ?  Enter

Set printer baud rate 'br' [9600] ?  Enter
```

After you respond to each of the prompts, `lprsetup` prompts you to determine if you want to change any of the values assigned to the various symbols in your `/etc/printcap` file or if you want to specify any additional symbols. For example, you can set a specific page length or width. If you want to make any changes or add information, enter the appropriate symbol name. See `printcap`(4) for more information.

```
Enter the name of the printcap symbol you wish to modify.  Other
valid entries are:
        'q'     to quit (no more changes)
        'p'     to print the symbols you have specified so far
        'l'     to list all the possible symbols and defaults
The names of the printcap symbols are:

 af  br  cf  ct  df  dn  du  fc  ff  fo  fs  gf  if  lf  lo  lp
 mc  mj  mx  nc  nf  of  on  pl  pp  ps  pw  px  py  rf  rm  rp
 rs  rw  sb  sc  sd  sf  sh  st  tf  tr  vf  xc  xf  xn  xs  ya
 yd  yj  yp  ys  yt

Enter symbol name: q

        Printer #1
        ----------
Symbol  type  value
------  ----  -----
  af    STR    /usr/adm/lp1acct
  br    INT    9600
  ct    STR    dev
  fc    INT    0177777
  fs    INT    03
  if    STR    /usr/lbin/ln03rof
  lf    STR    /usr/spool/lp1err
  lp    STR    /dev/tty01
  mx    INT    0
  of    STR    /usr/lbin/ln03rof
  pl    INT    66
  pw    INT    80
  rw    BOOL   on
  sd    STR    /usr/spool/lpd1
  xc    INT    0177777
```

```
  xf    STR    /usr/lbin/xf
  xs    INT    044000

Are these the final values for printer 0 ? [y] y
```

Next, the `lprsetup` script prompts you to add comments to the `/etc/printcap` file. Enter `n` at the prompt if you do not want to add comments. Enter `y` at the prompt if you want to add comments. At the number sign (#) prompt, enter your comment. Press Return or Enter at the number sign (#) prompt to exit. The comments are inserted directly above the `printcap` entry in the `/etc/printcap` file.

```
Adding comments to printcap file for new printer, type '?' for help.
Do you want to add comments to the printcap file [n] ? : y
Enter comments below - Press ENTER on empty line to exit
# Use this printer for drafts only
#  Enter

Set up activity is complete for this printer.
Verify that the printer works properly by using
the lpr(1) command to send files to the printer.


Command  < add modify delete exit view quit help >: exit
```

See `lprsetup`(8) for more information.

8.3.6 Advanced Printing Software Print Symbols

When setting up Advanced Printing Software, you should select the `Generic_Remote_LPD` printer type and set the following print symbols:

`rm` — Specify `@dpa` to indicate that jobs submitted to this printer are directed to the Advanced Printing Software inbound gateway. The inbound gateway submits the job to an Advanced Printing Software spooler.

`rp` — Specify the name of the Advanced Printing Software logical printer.

See the *Advanced Printing System Administration and Operation Guide* for more information.

8.4 Routine Print System Maintenance

The first part of this chapter showed you how to set up the first printer on a system. The following sections describe the routine administrative tasks for a print system that is set up. You can use the `printconfig` and the X11–compliant (CDE) or command line tools to perform these tasks. The tasks described in the following sections are:

- Adding additional new printers to the system (Section 8.4.1)
- Modifying characteristics of existing printers (Section 8.4.2)
- Removing printers from the system (Section 8.4.3)
- Controlling printer operations by using the CDE tools or the `lpc` command (Section 8.4.4)
- Enabling printer accounting (Section 8.4.5)

If you manually remove printers from the `/etc/printcap` file, you also have to manually remove spooling, accounting, and error directories and files.

8.4.1 Adding Printers

After you set up one printer, you can add other local, remote and networked printers at any time. Gather the information about each printer as described in Section 8.3.

You can add printers by running `printconfig`, or you can add printers manually by performing the following steps:

1. Create a printer spooling directory, if it does not already exist. See Section 8.5.2.6.2.
2. Modify the `/etc/printcap` file and edit it to include a description of the printer using the configuration data from the file in `/usr/lbin/lprsetup` that corresponds to the printer. See Section 8.5.1.
3. Create an accounting file and a log file and enable printer accounting. See Section 8.4.5. Set the protection and ownership of this file appropriately.

Ensure that the `/etc/inittab` file does not invoke the `getty` process on serial lines that have printers attached. If you use `printconfig`, this is done for you.

8.4.2 Modifying Printer Configuration

To modify a printer's configuration, run `printconfig` and choose the configured printer. Then choose Modify to display the Settings window.

If you change the name of the spooling directory, the accounting file, or the error log file, `printconfig` prompts you to verify that the information is correct before it deletes the original information.

To modify a printer's configuration manually, edit the `/etc/printcap` file and modify the printer entry. See Section 8.5.1 and `printcap`(4) for information about the `/etc/printcap` file symbols.

8.4.3 Removing Printers

To remove a printer, run the `printconfig` utility and choose the printer that you want to delete, then choose Deconfigure. Then you are prompted for confirmation that you want to delete the error log file and the accounting file. Several printers can share an accounting file. If you have such shared files, do not delete them.

The `lprsetup` command line utility does not delete the comments when you remove a printer if you included comments for the printer in the first line of its `/etc/printcap` file entry. You can edit the `/etc/printcap` file and delete the comments.

To remove a printer manually, edit the `/etc/printcap` file and delete the entry that relates to the printer. Manually delete the accounting and log file and the spooling directory if no longer required.

You also can use `lpc` and the CDE print management tools to temporarily control access to printers and queues. See Section 8.4.4.

8.4.4 Controlling Local Print Jobs and Queues

To manage the flow of print jobs and the contents of local print queues, you can use the `lpc` command line utility.

If CDE is your local user environment, you also can manage print jobs using the Print Manager located in the CDE Application Manager – Desktop_Apps folder. See the online help for information on how to use these graphical user interfaces and to the *Common Desktop Environment: User's Guide* and *Common Desktop Environment: Advanced User's and System Administrator's Guide*.

You can use the `lpc` command to:

- Enable and disable printers and spooling queues
- Change the order of queued jobs
- Display the status of the printer, queue, and daemon

You must have superuser privileges to perform some `lpc` commands, for example, the `disable` command.

Note

You can use the `lpc` command only to manage print queues that are local to your system. Although a remote printer has both a local queue and a remote queue, the `lpc` command manages only the local queue.

There are several command arguments that you can specify with the lpc command. You also can use the lpc command interactively. If you enter the lpc command without any command arguments, the lpc> prompt is displayed. After that, you can enter command arguments.

The lpc command has the following syntax:

/usr/sbin/lpc [*argument*] [all | *printer*...]

Some of the command arguments allow you to specify all to indicate all the printers or to specify one or more *printer* variables to indicate a specific printer.

Table 8–3 lists the command arguments of the lpc command.

Table 8–3: lpc Command Arguments

lpc Argument	Description
help [*argument*]	Prints a one-line description of the specified lpc command argument. If an *argument* variable is not specified, the list of arguments is displayed.
? [*argument*]	Same as the help argument.
abort	Terminates an active lpd daemon and then disables printing. This prevents the lpr or lp command from starting a new lpd daemon.
check	Examines the printer description file and other components of the printing environment for potential configuration errors for each named printer
clean	Removes any temporary files, data files, and control files (for example, files that do not form a complete printer job) from the specified print spooling directory.
disable	Turns off the specified print spooling queue. This prevents the lpr or lp command from entering new jobs in the queue.
down *message...*	Turns off the specified print queue, disables printing, and enters the specified message in the printer status file. You do not need to quote the message because remaining arguments are treated the same as echo. You can use the down argument to take down a printer and inform users. If a printer is down, the lpq command indicates that the printer is down.
enable	Enables spooling for the specified printers. This enables the lpr or the lp command to enter print jobs in the spooling queue.
exit	Exits from lpc.
quit	Exits from lpc.

Table 8–3: lpc Command Arguments (cont.)

lpc Argument	Description
`restart`	Attempts to start a new `lpd` daemon for the specified printer. This argument is useful if some abnormal condition causes the daemon to terminate unexpectedly and leave jobs in the queue. If this occurs, the `lpq` command indicates that no daemon is present. If a daemon is hung, you must kill the process, then restart the daemon by using the `restart` argument.
`start`	Enables printing and starts a spooling daemon for the specified printer.
`status` [*printer*]	Displays the status of the specified printer daemon and queue. The `status` argument shows if the queue is enabled, if printing is enabled, the number of entries in the queue, and the status of the printer's `lpd` daemon. If a printer name is not supplied, information about all printer daemons and queues is displayed.
`stop`	Stops a spooling daemon after the current job is complete and disables printing.
`topq` *printer*	Puts print jobs in the queue in the specified order. You can specify the print jobs by also specifying a *request_ID* variable or a *username* variable.
`up`	Enables all printing and starts a new printer daemon. Cancels the `down` argument.

The following example shows that the `lpd` daemon is active on the printer named `tester` and there is one entry in the queue:

```
# /usr/sbin/lpc
lpc> status tester
tester:
        printer is on device '/dev/tty02' speed 9600
        queuing is enabled
        printing is enabled
        1 entry in spool area
lpc>
```

See `lpc`(8) for more information.

8.4.5 Enabling Printer Accounting

Printer accounting allows you to charge users for printing services and to determine the amount of printer usage.

Note

Accounting is provided only for unformated text files; it is not provided for preformatted files such as PCL and PostScript.

There are two types of printer accounting:

printer user accounting	Printer user accounting provides information about printer use according to the system and user name that issues the print request.
printer summary accounting	Printer summary accounting provides information about the amount of media (number of printed pages or number of feet of roll paper or film) the printer produces. Specify the `pac` command with the `-s` option to produce printer summary accounting information.

The printer accounting files default to the `/var/adm` directory. Adding a printer with the `lprsetup` command creates the accounting file you specify. The `/usr/adm/lpd/lpacct` file is the default accounting file.

The `printconfig` utility provides default accounting files in the Advanced Options. If you do not require accounting, you can remove these entries during printer configuration or at any later date using the Modify option. If you add a printer manually, you must create the accounting file.

Note

User `adm` owns the `/var/adm/lpd` directory and it belongs to the `adm` group. User `adm` also owns printer accounting files, which have a protection mode of 644 and belong to the `system` group.

In the printer's `/etc/printcap` entry, the `af` parameter specifies the name of the accounting file. The accounting process uses this file to record the number of pages printed by each user on each printer. The name of the accounting file is unique for each printer on your system. Use the `pac` utility to display information in the printer accounting files. User `daemon` owns the accounting file, and it is a member of the `daemon` group. The correct file ownership is set automatically if you use the `printconfig` utility to specify the printer accounting file. The `af` parameter is not applicable for remote printer entries.

Accounting is accomplished through programs called print filters. The `printconfig` utility suggests a default print filter. The `if`print filter symbols is needed for accounting. For example:

```
if=/usr/lbin/ln03rof
```

If you want to use separate accounting files for each printer on your system, ensure that the file names are unique. An unlimited number of printers can share an accounting file but you cannot specify an accounting file for remote printers. The print daemon owns the accounting files. If you specify an accounting file, intermediate directories are created automatically as needed.

8.5 Reference Information

The following sections discuss the information you need to configure a printer, on the line printer daemon (`lpd`), and on the print system files. The system files are created automatically if you use `printconfig` as described in Section 8.3.2.

You also can create and modify the files manually. If you do so, you also must manually change the `/etc/printcap` file so the changes can take effect.

8.5.1 The /etc/printcap File

The `lpd` daemon uses information in the `/etc/printcap` database file to print requests. Each entry in the `/etc/printcap` file describes a printer. You specify printer characteristics using two-letter abbreviations called print symbols. The print symbols are described in this section and in `printcap`(4). The `lprsetup` utility modifies the `/etc/printcap` file.

The default printer is named `lp` and it can be used as an alias for any type of printer, local or remote. The names `lp0`, `lp1`, and so on are default printer names that `printconfig` provides. You can use or ignore them as you want.

The following examples show `/etc/printcap` entries for a TCP/IP connected printer, an LPD remote printer, a serial printer, and a parallel printer.

TCP/IP connected printer example

```
lp0|myprinter|hp8000:\
        :ct=tcp:\
        :if=/usr/lbin/ppdof +OPageSize=Letter +Chplj8000.rpd:\
        :lf=/usr/adm/lp0err:\
        :lp=@myprinter.com/9100:\
        :mx#0:\
        :of=/usr/lbin/ppdof +OPageSize=Letter +Chplj8000.rpd:\
        :pl#66:\
        :pw#0:\
        :rw:\
        :sd=/usr/spool/lpd0:
        :xf=/usr/lbin/xf:
```

LPD Remote printer example

```
lp0|remote:\
        :lf=/usr/adm/lp0err:\
        :lp=:\
        :rm=system:\
        :rp=queue:\
        :sd=/usr/spool/lpd0:
```

Note

Most of the printcap options for this configuration are disabled

Serial port example

```
lp0|myprinter|la75:\
        :af=/usr/adm/lp0acct:\
        :br#9600:\
        :ct=dev:\
        :fc#0177777:\
        :fs#03:\
        :if=/usr/lbin/la75of:\
        :lf=/usr/adm/lp0err:\
        :lp=/dev/tty00:\
        :mx#0:\
        :of=/usr/lbin/la75of:\
        :pl#66:\
        :pw#80:\
        :rw:\
        :sd=/usr/spool/lpd0:\
        :xc#0177777:\
        :xf=/usr/lbin/xf:\
        :xs#044000:
```

Parallel port example

```
lp0|myprinter|la75:\
        :af=/usr/adm/lp0acct:\
        :ct=dev:\
        :if=/usr/lbin/la75of:\
        :lf=/usr/adm/lp0err:\
        :lp=/dev/lp0:\
        :mx#0:\
        :of=/usr/lbin/la75of:\
        :pl#66:\
        :pw#80:\
        :sd=/usr/spool/lpd0:\
        :sh:\
        :xf=/usr/lbin/xf:
```

The following example shows an /etc/printcap entry for both a local printer and a remote printer. The callouts describe the symbol entries:

```
#
#
lp|lp0|0|dotmatrix|mary:\
      :af=/usr/adm/printer/lp.acct:\
      :br#9600:\
      :ct=dev:\
      :fc#0177777:\
      :fs#023:\
      :if=/usr/lbin/la75of:\
      :lf=/usr/adm/lperr:\    1
      :lp=/dev/tty01:\
      :mx#0:\
      :of=/usr/lbin/la75of:\
      :pl#66:\
      :pw#80:\
      :sh:\     2
      :sd=/usr/spool/lpd:\
      :xc#0177777:\     3
      :xf=/usr/lbin/xf:\
      :xs#044000:\
#
sqirrl|3r3|ln03r3|postscript3|In office 2T20:\
      :lp=:rm=uptown:rp=lp:sd=/var/spool/printer/ln03r3:mx#0:\ 4
#
```

1 Specifies a symbol with alphabetic characters.

2 Specifies a symbol that represents a Boolean expression.

3 Specifies a symbol with a numeric value.

4 Specifies an entry for a remote printer. The lp, rm, rp, and sd symbols are required for remote printers for which you are a client.

The first line of a printer entry contains the fields that specify the printer primary reference name and printer name synonyms. This first line and these fields are required for every printer, both local and remote.

The printer reference name is the name that you subsequently use in order to specify printing to this printer. You can give each printer as many alternative reference names as you want, separating each field on the first line by using a vertical bar (|). The first line must end with a colon (:).

Note

A local printer entry in the /etc/printcap file should have the default printer reference name lp0 so that print jobs can have

a destination when printer reference names are not specified in print commands.

The remaining lines of each printer entry contain the descriptive symbols and values that define the printer's configuration. Symbols are two-character mnemonics and you can specify them by using an equal sign (=) and alphabetic characters or with a number sign (#) and a numeric value. Some symbol names have Boolean equivalents, which do not use parameters. You can specify the symbols on one line or on individual lines, but you must separate them with colons (:).

To make the `/etc/printcap` file easy to read, you can place a colon (:) at the beginning of a line and a backslash (/\) at the end of a line to separate the symbols.

See `printcap`(4), which lists the `printcap` symbol names, the type of values they accept, default values, and descriptions of the symbols.

8.5.2 Data in /etc/printcap

The following sections describe the information typically required for a printer entry in the `/etc/printcap` file.

8.5.2.1 Printer Name

The printer name is the name by which you want to identify the printer through the `lpr` command. For example:

```
# lpr -Pprintername
```

The `lprsetup` utility uses an internal numbering scheme from 0 to 99. The next available number is the default name. You can choose the default by pressing the Enter key or by entering any other alphanumeric name that is appropriate. The `lprsetup` utility always assigns at least two printer synonyms. The default number *N* is one synonym. The string `lp` plus the default number (`lp`*N*) is the other system-assigned synonym. If the default number is 1, the two assigned names are `1` and `lp1`. You can queue jobs to this printer using either of the following commands:

```
# lpr -P1
# lpr -Plp1
```

You also can assign your own synonyms and use them to direct jobs to printers.

The first printer that `lprsetup` sets up has the names `0` and `lp0`. The name `lp` is reserved for the default printer (that is, the printer used when none is specified in an `lpr` command).

If you have only one printer or are entering the first of many printer names, the first name has a printer number of 0. This is recognized as your system's default printer if it has an additional name of `lp`.

Ask your network administrator for the names of the remote printers on the network.

8.5.2.2 Printer Type

The printer type corresponds to the product name of the printer, such as the LN03 laser printer. If you are using the `lprsetup` utility, printers are listed by type; only supported types are listed. These printers have some default values already included in the setup utility.

The supported printer types are defined in the files in the `/usr/lbin/lprsetup` directory. See `lprsetup.dat`(4) for information on generic printer types and for information on the printer types used in conjunction with HP's Advanced Server.

You can set up unlisted printers by selecting the `Generic_Unknown` printer type and responding to the prompts, using values similar to those for supported printers.

When specifying the printer type, you must use full command names and printer names. The default printer type is `Generic_Unknown`.

To install third-party printers, see the documentation that came with the printer.

8.5.2.3 Printer Synonyms

The printer synonym is an alternate name for the printer. Some examples include `draft`, `letter`, and `LA-75 Companion Printer`. You can enter as many alternate names for a printer as you like, but the total length of the line containing all the names must be less than 80 characters. When entering printer synonyms that can consist of many names, the entry process is terminated when you either enter a blank line or enter a line containing only white space.

In command line mode, after entering a synonym, you are prompted again. If you do not want to enter any more synonyms, press Enter to continue.

Each synonym (including the printer number) identifies the printer to the print system. For example, if you chose the synonym `draft` for a printer, the following command prints files on this printer:

```
$ lpr -Pdraft files
```

The special printer synonym `lp` specifies the default printer.

8.5.2.4 Device Special File

The device special file provides access to the port on the computer to which the printer is connected. The device special file is used if the printer is connected directly to a local serial or parallel port. In this case, you must equate a printer device logical name to the printer's device special file name by using the `lp` symbol in the `/etc/printcap` file. For example:

```
lp=/dev/lp0
```

The installation procedure creates some device special files for the hardware that is connected to your computer. Usually, the device special files for parallel printers are named `/dev/lp`*n* (for example: `lp0`, `lp1`, `lp2`), and the device special files for serial line printers are named `/dev/tty`*nn* (for example: `tty00`, `tty01`, `tty02`). The *n* and *nn* variables specify the number of the printer. On most systems, the device names map to default physical ports (connectors).

Table 8–4 shows the mapping of device names to default physical ports.

Table 8–4: Communication Ports and Printer Device Special Files

Device Special File	Communication Type	Connector Label
`/dev/lp0`	parallel	printer, or lp
`/dev/tty00`	serial	COMM1 or 1
`/dev/tty01`	serial	COMM2 or 2

If only one 9–pin serial connector is provided on a system, it is not always labeled as such. Some systems also use graphical icons instead of labels. See the owner's manual for the system for more information.

Note

If the port is used for logins, the `lprsetup` script turns off the terminal line established by the `getty` process.

For TCP/IP printers, specify the `lp` symbol with an asterisk character (@) followed by the *`printer hostname`* and either a port or service name as shown here:

```
lp=@printer hostname/servicename
lp=@printer hostname/portnumber
```

The *`printer hostname`* is the network name or address that specifies the TCP/IP address of the printer's network interface. A *`portnumber`* is an integer TCP port number that your printer uses for raw socket printing. A *`servicename`* is a name you define in the `/etc/services` file. The service

definition includes the service name, the protocol “tcp”, and the port number that your printer uses for raw socket printing. To illustrate this, consider this entry in the /etc/services file:

```
print_port_9100       9100/tcp       # printer port 9100
```

Also consider this entry for the /etc/printcap file:

```
lp1|1|myprinter:\
          :lp=@printer123.sf.ourcomp.com/print_port_9100:\
          :lf=/usr/adm/lp1err:\
          :if=/usr/lbin/ppdof +OPageSize=Letter +Chplj9000.rpd:\
          :mx#0:\
          :of=/usr/lbin/ppdof +OPageSize=Letter +Chplj9000.rpd:\
          :pl#66:\
          :pw#0:\
          :rw:\
          :sd=/usr/spool/lpd1:\
          :xf=/usr/lbin/xf:
```

8.5.2.5 Connection Type

The ct parameter specifies the type of connection to the printer. You can connect a printer directly to your computer from a port or terminal line. You can access networked printers that are connected to a LAT (Local Area Transport) terminal server or to a remote host. If you are using lprsetup, the choices for the connection type are:

dev	for local devices
lat	for LAT devices
tcp	for TCP/IP devices

8.5.2.6 Spooling Directories

In /etc/printcap, the sd parameter specifies the spooling directory where files are queued before they are printed. Each spooling directory is unique. All printcap file entries must specify a spooling directory, both local and remote. When the spooling directory is created with printconfig, intermediate directories are created as necessary.

Each printer must have its own spooling directory located under the /usr/spool directory. The spooling directory acts as a printer’s spooling queue; it contains the files that are queued for printing on that printer. A printer spooling directory should have the same name as the printer reference name and is located on the system attached to the printer. The

printer reference name is the name that you specify to print on a particular printer.

If you are using `lprsetup`, the utility supplies the default value `/usr/spool/lpd`*n*. The *n* variable specifies the printer number. For example, the default name of the spooling directory for a second line printer is `/usr/spool/lpd2`.

Each printer entry in the `/etc/printcap` file must specify a spooling directory even if the printer is connected to another system or is on another network. You specify a spooling directory with the `sd` symbol. For example:

```
sd=/usr/spool/local_printer1
sd=/usr/spool/network_printer1
sd=/usr/spool/remote_printer1
```

Note

Do not put subdirectories under `/usr/spool/lpd` or `/var/spool/lpd`. Do not rename or delete this pathname because it is a CDSL used by the parent lpd process and is unique for each node in a cluster. See the section on CDSLs in Chapter 6

8.5.2.6.1 Spooling Directory Files

A spooling directory contains a `status` file and a `lock` file that are created by the `lpd` daemon when a file is queued for printing. The `/usr/spool/lpd/lpd.lock` file contains the process identifier of the parent `lpd` process that listens for print jobs request on the local `/dev/printer` socket and the network socket 515. The processes that actually print the jobs are child daemons forked by the parent. Their process identifiers are stored in the `lock` file in the spool directory, such as `/usr/spool/local_printer1/lock`.

The `lock` file is used to regulate the creation of a child process, or `lpd` daemon, for job processing in a single queue, so that only one queue daemon runs at a time. The `lock` file contains the process identification number of the daemon that is currently running and the control file name of the job currently being processed. The `status` file contains a line that describes the current printer status. This line is displayed if a user inquires about printer status. If a printer whose status is queried is not active, the status message written to standard output is `no entries`. Two additional temporary files may appear in the spooling directory:

`.no_daemon` — This file is created when the queue has entries and no daemon is running and no files in the spooling

	directory are removed or have changed in the past ten seconds.
`.seq`	This file, created by the `/usr/bin/lpr` command, generates job numbers in sequence starting from zero.

When the `lpd` daemon is activated as a result of a print request, it looks in the printer spooling directory for a `lock` file. If a `lock` file is not found, the `lpd` daemon creates one and writes the identification number and the control file name on two successive lines in the file. The `lpd` daemon then scans the printer spooling directory for control files whose names begin with `cf`. Control files specify the names of print files that users have submitted and contain printing instructions for these files. Each line in a control file begins with a key character that indicates what to do with the remainder of the line. The key characters and their meanings are described in detail in `lpd`(8).

Note

Job control files (they begin with `cf`) are first created as temporary files, which begin with `tf`. If the job creation process terminates abnormally, temporary control files may be left behind. You should delete these files because they interfere with future job creation when the job number in the `.seq` file equals any value used in the temporary control file name.

Data files, whose names usually begin with `df`, also are located in the spooling directory. Data files contain the job to be printed. The job data may or may not be modified by a print filter, which is specified in `/etc/printcap` by an`if` or by `of` if `if` is omitted.

After a file is printed, the `lpd` daemon removes the control and data files from the printer spooling queue, updates the status file, and sets up the next file in the spooling queue for printing.

For example, if users submit jobs to a printer named `milhaus`, the following command lists the files that are stored in the spooling directory:

```
# ls –l /usr/spool/local_printer1
-rw-rw----   1    root      75  Jan 17  09:57  cfA0220mothra
-rw-rw----   1    root      96  Jan 17  10:03  cfA143harald
-rw-rw----   1    root  199719  Jan 17  09:57  dfA0220mothra
-rw-rw----   1    root    9489  Jan 17  10:03  dfA143harald
-rw-r--r--   1    root      20  Jan 17  10:06  lock
-rw-rw-rw-   1  daemon     113  Jan 17  10:00  status
```

8.5.2.6.2 Creating a Spooling Directory

If you want to manually add a printer, use the `mkdir` command to create the spooling directories for each printer. Set the spooling directory permission mode to 775. Set the directory's group and ownership mode to the name `daemon`. For example:

```
# cd /var/spool
# mkdir ./printers
# mkdir ./printers/lp1
# cd printers
# chmod 775 lp1
# chgrp daemon lp1
# chown daemon lp1
# ls -l lp1
drwxr-xr-x  2  daemon  daemon  24  Jan 12  1994  lp1
```

8.5.2.7 Baud Rate

The baud rate applies to directly connected serial line printers only.

The baud rate is the maximum rate at which data can travel between the data source and the printer (for example, 4800 or 9600). The default baud rate for your printer should appear in the printer documentation. If you reset this baud rate yourself during the installation of the printer hardware, the rate that you set on the printer must match the rate that you enter in the `/etc/printcap` file.

8.5.3 Line Printer Daemon

Printers are controlled by the line printer daemon, `lpd`, which is located in the `/usr/lbin` directory. Printing cannot take place unless the `lpd` daemon is running. The `lpd` daemon has many functions:

- Handles printer spooling conjunction with the `lpr` and `lprm` commands. Spooling is the mechanism by which a file is placed in a queue until the printer can print the file.
- Scans the `/etc/printcap` file to determine printer characteristics.
- Uses specific print filters for print requests. Print filters translate an input format into a printer-specific output format.
- After a system reboot, prints any files that were not printed when the system stopped operating.

When you use the `lpr` command, it copies files into the spooling directory and activates the `lpd` daemon. Requests are printed in the order in which they enter the queue. A copy of the file remains in the queue until the printer is ready to print it; then the `lpd` daemon removes the file from the spooling queue after it sends the job to the printer.

After you install and boot your system, the lpd daemon usually is started by the /sbin/init.d/lpd utility. You can start and stop the lpd daemon with the following commands:

/sbin/init.d/lpd [-l] **start**

/sbin/init.d/lpd stop

The -l option causes the lpd daemon to log valid requests from the network. This option is useful for debugging.

To test whether the line printer daemon is running, enter:

```
# ps agx | grep /usr/lbin/lpd | grep -v grep
```

8.5.4 Error Logging

Errors logged by the lpd daemon are logged to /var/adm/syslog.dated/<*date*>/lpr.log (or /var/adm/syslog.dated/current/lpr.log, which is a symbolic link to the most recent log file). The directory <*date*> is named for the date and time that the logs were saved. A typical log file entry is as follows:

```
Apr 15 16:36:28 cymro lpd[1144]: ERROR -- lpr: cannot open printer description file
Apr 15 16:36:28 cymro lpd[1144]: ERROR -- exiting ...
Apr 15 16:36:46 cymro lpd[1130]: ERROR -- lpq: cannot open printer description file
Apr 15 16:36:46 cymro lpd[1130]: ERROR -- exiting ...
#
```

Monitor log files regularly for errors and deleted (or archive) the logs to prevent them from filling up the available disk space. You can establish a regular clean up procedure using the cron utility. You also can control the severity level of the messages that are posted to lpr.log by specifying only the required priorities in the file /etc/syslog.conf. See the syslogd(8) for information.

The lf parameter specifies the log file where most print filter errors are reported. The default log file, if one is not specified, is /dev/console. If you have more than one printer on your system, give each log file a unique name. When the error log file is created using printconfig, intermediate directories in the pathname are created as necessary.

The lpd daemon logs most of its printer errors to the error log file rather than the error file specified by the lf parameter. Therefore, specifying an error log file is optional. If you used lprsetup to install the printer, the utility provides the default value /usr/adm/lpd/lperr. If you do not specify an error log file, errors are logged to /dev/console.

The error log file is specified with the lf symbol in the /etc/printcap file. For example:

```
lf=/var/adm/lpd/lpderr
```

Error log files usually are located in the /var/adm directory. Local printers can share an error log file, but you should specify the file in each /etc/printcap file printer entry.

8.5.5 Print Filters and Filter Directories

The filters for the lpd daemon translate the data that you want to print into the format appropriate for your printer. In many cases the filter is specified to match a specific printer. For example, use the ln03rof filter to print files with the LN03R printer.

When both input and output filters are specified, input filters process job data being sent to the printer and output filters process banner page data generated by the lpd. You can specify many filters as either input or output filters, and each operates differently depending on whether they are called as input or output filters. These filters are specified in the /etc/printcap file as follows:

```
if=/usr/lbin/ln03rof
of=/usr/lbin/ln03rof
```

Input filters are also responsible for printer accounting, which enables you to keep a record of printer usage for text jobs (not for PostScript printing). See Chapter 10 for information on configuring printer accounting.

If only an input filter is specified, it filters the job data and the banner page data, and performs accounting.

If only an output filter is specified, it filters the job data in addition to the banner page data, but no accounting is performed.

These actions can be summarized by the following table.

Filter	Job Data	Banner Page Data	Accounting
Both input and output filters			
Input filter	x		x
Output filter		x	
Input filter only	x	x	x
Output filter only	x	x	

See lpd(8) for more information.

See lprsetup.dat(4) for information on generic print filters and print filters used with HP's Advanced Server; these print filters are located in the /usr/lbin/lprsetup directory. For printers not listed in

`lprsetup.dat`(4) or defined in that directory, see the printer documentation or contact the printer manufacturer for filter information.

Note

Manufacturers of many third party printers that provide Tru64 UNIX support supply filters that are installed by their own installation software. It is recommended that third party printer software, and the queues that they create, be modified by the printer manufacturer only.

8.5.6 Flag Bits

Flag bits specify characteristics about data transmission from the host to the printer and, if possible, from the printer to the host on a serial line only (LAT and RS232). Data that is passed from the printer to the host may include stop and start status information, which tells the host that the printer input buffer can accept input or that it is about to overflow.

Delays are specific times used to slow the transmission of the next group of characters to the input buffer. Delays give the printer mechanism time to perform operations such as a carriage return, newline, tab, and form feed.

Flag bits are cleared with the `fc` symbol and set with the `fs` symbol. All printers do not use all the flag bits, but you must either set the bits or clear them. See the manual for the printer for specific information about flag bits.

The flag bits are specified as octal numbers in a 16-bit word. Octal values are preceded with the number zero (0). To clear all the bits, specify the value 0177777 with the `fc` symbol. To set all the bits, specify the value 0177777 with the `fs` symbol. Clear all bits by using `fc#0177777`) before you call the `fs` symbol. To set or clear any groups of bits, specify the octal sum of the combined bits for the number of flag bits.

The following is an example of flag bit specifications:

```
fc#0177777
fs#0141
```

In this example, `fc#0177777` clears all bits and the `fs` symbol is set to `0141` specifying the `OPOST`, `ONLRET`, and `OFILL` flag bits.

Table 8–5 lists each flag bit name, its octal value, and its description.

Table 8–5: Flag Bits

Flag	Octal Value	Description
OPOST	0000001	Enable output processing
ONLCR	0000002	Map NL to CR-NL
OLCUC	0000004	Map lower case to upper case
OCRNL	0000010	Map CR to NL
ONOCR	0000020	No CR output at column 0
ONLRET	0000040	NL performs CR function
OFILL	0000100	Use fill characters for delay
OFDEL	0000200	Fill is DEL, else NUL
NLDLY	0001400	Newline delay
NL0	0000000	
NL1	0000400	
NL2	0001000	
NL3	0001400	
TABDLY	0006000	Horizontal tab delay
TAB0	0000000	
TAB1	0002000	
TAB2	0004000	
TAB4	0006000	
CRDLY	0030000	Carriage Return delay
CR0	0000000	
CR1	0010000	
CR2	0020000	
CR3	0030000	
FFDLY	0040000	Form feed delay
FF0	0000000	
FF1	0040000	
BSDLY	0100000	Backspace delay
BS0	0000000	
BS1	0100000	
OXTABS	1000000	Expand tabs to spaces

See `tty`(7) for more information.

8.5.7 Mode Bits

Mode bits specify details about the capability of a particular terminal and usually do not affect printer operation. Mode bits are cleared with the `xc` symbol and set with the `xs` symbol. Some printers use all the mode bits, so you either must set them or clear them. The mode bits are specified as octal numbers in a 16-bit word format. You should clear all bits by specifying `xc#0177777` before you specify the `xs` symbol.

See `tty`(7) for more information.

The following is an example of mode bits specifications:

```
xc#0177777
xs#044000
```

As shown in the previous example, `xc#0177777` clears all bits and the `xs` symbol is set to `044000` specifying the `ECHO` and `ECHOCTL` mode bits.

Table 8–6 lists a description of each mode bit.

Table 8–6: Mode Bits

Mode	Octal Value	Description
ECHOKE	0000001	Echoes KILL by erasing the line
ECHOE	0000002	Visually erase characters
ECHOK	0000004	Echoes NL after KILL
ECHO	0000010	Enable echoing
ECHONL	0000020	Echoes NL even if ECHO is off
ECHOPRT	0000040	Echo erased chars between and /
ECHOCTL	0000100	Echo control characters as ^(char)
ISIG	0000200	Enable special chars INTR, QUIT and SUSP
ICANON	0000400	Enable canonical input
ALTWERASE	0001000	Use alternate word erase algorithm
IEXTEN	0002000	Enable FLUSHO and LNEXT
XCASE	0040000	Canonical upper/lower presentation

8.5.8 Remote Printer Characteristics

For users to access a printer from a remote system, both the local system and the remote system require printer information in their `/etc/printcap` files. On the local system attached to the printer, security is controlled by the entries in `/etc/hosts.lpd` or `/etc/hosts.equiv`.

Optionally you can specify the rs symbol, which specifies a Boolean value that takes only a true (yes) or false (no) value, along with the other printer configuration symbols. If you define the value as true, remote users must have an account on the local system that is attached to the printer. If you define the value as false, remote users can access the local printer if the local printer is listed in the /etc/hosts file. See Section 8.5.1 for an example of an /etc/printcap file.

On the remote system, you must specify the rm, rp, lp, and sd symbols.

The rm symbol specifies the name of the system attached to the printer. For example:

```
rm=deccom
```

The rp symbol specifies the printer spool name on the remote system. For example:

```
rp=ln03lab
```

For remote printers, specify the lp symbol without a value:

```
lp=
```

The sd symbol specifies the spooling directory. For example:

```
sd=/usr/spool/lpd
```

8.6 Print Filters

Tru64 UNIX provides print filters for generic and for local-language use.

8.6.1 The pcfof Print Filter

The pcfof print filter is designed to accommodate many different printers through the use of a printer control file (PCF). PCF files contain printer control strings to set up and select printer-specific features such as paper tray selection, duplexing, and printing orientation. The filter is designed to work with text, ANSI, PCL, and auto-sensing multilanguage PostScript printers, but does not work with PostScript-only printers.

PCF files are text files. You can use any text editor to modify an existing file to customize printing behavior or create a new file for an unsupported printer. PCF files provided in Tru64 UNIX are replaced during an installation update, so you should take care that you preserve any customizations in backups. Using a file name prefix for new or modified PCF files prevents potential file name conflicts. For example, copy file names before customizing as follows:

```
# cp ln32.pcf my_ln32.pcf
```

The print filter is located in /usr/lbin and the PCF printer specific files are in /usr/lbin/pcf. The file template.pcf provides documentation on the PCF file format.

8.6.2 The wwpsof Print Filter

The wwpsof filter uses settings in a printer customization file (PCF) to find the font glyphs for local language characters and then embeds the font data in the PostScript file. The filter uses PostScript outline fonts, if installed on the local system, or bitmap fonts, which the filter obtains through a font server. This feature supports print jobs that contain multinational character sets and you do not need to send such jobs to special printers. See wwpsof(8) for a list of options. See the *Writing Software for the International Market* and i18n_printing(5) for more information on local-language printer support.

8.6.3 Known Restrictions of Filter Use

The following are current restrictions on the use of print filters:

- TCP/IP printing works when printing within a local subnet; however, printing in complex networks across one or more routers may cause reliability problems. You may need to configure network cards in the printer in order to identify the router. See the printer documentation for information.
- Printing non-PostScript files with some PostScript and non-PostScript filters may yield unexpected results.

  Table 8–7 lists the filters which may cause problems.

Table 8–7: Non-PostScript and PostScript Filters

Filter Name	Filter Type
lpf	Non-PostScript
la75of	Non-PostScript
la324of	Non-PostScript
lqf	Non-PostScript
hplaserof	PostScript

To provide expected behavior with older printers, these non-PostScript filters maintain a dependence on the serial port driver to automatically supply carriage returns after line feeds when you specify the (octal) 020 bit to the fs control variable in the /etc/printcap file.

Because this control bit is not interpreted by the network socket driver, the formatting behavior supplied by the serial port driver is

absent. Therefore, non-PostScript files that are not preformatted for the printer may not print as in serial-port-connected configurations. In particular, this may affect ASCII text files that do not contain embedded carriage-returns.

- Most printers using the `lpf`, `la75of`, `la324of`, and `lqf` non-PostScript filters do not provide network interface card support. However, users who use serial-and-parallel-port to network-port converters may still have problems. (An example is the HP RapidPrint network interface box, which allows printers to behave like TCP/IP printers.)
- The `hplaser4psof` PostScript filter works for PostScript files and for preformatted non-Postscript files (like PCL files), but it may produce unexpected results for files that are not preformatted (such as ASCII text without embedded carriage-returns).
- Some filters designed to work with character-set printing (such as ASCII) may not work for TCP/IP printing.

8.7 Testing and Troubleshooting Printers

A checklist for diagnosing printer problems is provided in this section. Most printer errors are logged in the `/var/adm/syslog.dated/current/lpr.log` file while some are logged in the `/usr/adm/lperr` file.

The `printconfig` window Printer Configuration on `host` contains an option to send test output to the printer immediately after configuration is complete. If the output is not printed, follow the troubleshooting steps described in this section.

You also can test a printer by using the `lpr` command to print a few pages of text. You should test any special printer features that you intend to use regularly on this printer, for example, PostScript or double-sided print. See `lpr`(1) for more information.

The `lptest` command writes a ripple test pattern to the standard output, or you can direct the output to a printer. A pattern that contains all 96 printable ASCII characters in each column is printed using 96 lines. In the pattern, each printed character is displaced rightward one character column on each successive line. This test is also useful for ascertaining the number of lines per page and the default page parameters. You can use the ripple test pattern to test printers, terminals, and drive terminal ports during debugging.

The `lptest` command has the following syntax:

/usr/sbin/lptest [*length* [*count*]]

Use the `lptest` command if you need quick output of random data. For example:

```
# /usr/sbin/lptest |lpr -P3r44
```

See lptest(8) for more information.

If a problem occurs on an existing printer or when adding a printer to a system, diagnose the problem as follows:

- See the error log files specified in Section 8.5.4.
- Examine the physical connections and if possible, swap the cable.
- Verify all part numbers to ensure that cables and connectors are appropriate and suitable for the configuration. Cable length can affect the available baud rate or communications method.
- Most printers have internal test and print test options. Use these test options to verify the hardware.
- Review the printer configuration, ensuring that the data entered is appropriate for the device. If the entries appear correct, try a generic or pass through filter to print a simple ASCII text file. Review the manufacturer's documentation to check the settings. Ensure that the correct settings are recorded in the /etc/printcap file. See Section 8.5.1.
- Ensure that the printer daemon is present by using the following command:

  ```
  # ps agx | grep /usr/lbin/lpd
  ```

  Sometimes, the parent lpd process becomes hung, or a child process does. If the daemon is not running, use the kill -9 command on each process, or using the stop and start commands described in Section 8.5.3.

  Using the -l option with /usr/lbin/lpd causes the daemon to log requests from the network. This flag is useful for debugging problems with remote printers. See lpd(8) for information.
- Examine the queue status and printer status using the CDE graphical tools or the lpc command line utility to ensure that printer and queue are enabled and available. If queues are stalled, try resetting the queues (see Section 8.4.4). If print jobs are being created and queued, try configuring a different local or remote printer.
- Ensure that the appropriate spooling or device files exist and that ownership and access are correct (see Section 8.5.2.6.1).
- Use the lpc check *printername* command to examine for configuration problems in the /etc/printcap file and elsewhere.

For networked and remote printers, you also have to ensure that the systems are properly connected and authorized to transfer print jobs. See the *Network Administration: Connections* manual for information on network troubleshooting.

9

Administering the Archiving Services

One of the more common tasks of a system administrator is helping users recover lost or corrupted files. To perform that task effectively, you must set up procedures for backing up files at frequent and regular intervals. This chapter describes how you use resident commands and utilities to back up (archive) and restore files and directories.

Design and implement a disaster recovery plan that describes how you intend to restore your entire operating system and user files to normal operations in the event of a catastrophic failure. This chapter does not describe the disaster recovery process, because it is often very specific to site operations and business requirements. However, backup operations are an important component of such a plan.

The following topics are included in this chapter:

- An overview of the steps and options involved in creating a backup (Section 9.1)
- A discussion on the main tasks involved in creating a backup (Section 9.2)
- Information on how to set up a backup schedule (Section 9.3)
- A discussion on the methods of creating a backup (Section 9.4)
- Information that enables you to prepare for a backup, such as references to other documents that you may need to read, system files created, related utilities, and prerequisite tasks (Section 9.5)
- A discussion on the use of the `dump` command to perform a backup (Section 9.6)
- A discussion on the use of the `restore` command to recover data from a backup (Section 9.7)
- A description of the commands that enable you to archive individual files and directories, rather than complete file systems (Section 9.8)
- A discussion on the use of `dxarchiver`, a graphical user interface for archiving files and directories (Section 9.9)
- Information on creating a bootable tape, which is a bootable backup of the root file system and key system files that may be useful for disaster recovery (Section 9.10)

9.1 Understanding Backup Tasks

This chapter describes basic backup operations for a system using the UFS file system. You also may need to use other backup and restore utilities if any of the following conditions apply to your local system:

- If you are using the Advanced File System (AdvFS) file system exclusively, or if you are using AdvFS domains on some of the disks attached to your system, see the *AdvFS Administration* manual. Using the AdvFS file system provides you with more backup features, such as the ability to clone domains. One of the disadvantages of the UFS file system is that you must prevent access to a UFS file system during a backup. If a user accesses a file while a backup is in process, the backup may not record changes in the file. To ensure a completely accurate back up of a UFS file system, you may need to take a disk off line or shut the system down to single user mode. If you are unable to schedule system shut downs, consider using the AdvFS file system.
- If you are using the Logical Storage Manager (LSM), see the *Logical Storage Manager* manual. Using features of LSM such as mirroring volumes, you may be able to overcome some of the backup limitations of UFS. For example, you can take an instant, accurate snapshot of a UFS file system by mirroring the file system on a different disk. Then you can break the mirror at any time to create an archive, with only a brief pause in system operations. Using LSM requires spare disk capacity and may be unsuitable for small systems with few disks.
- If you want to back up and restore a root volume to a different system, consider using configuration cloning. This feature is described in the *Installation Guide — Advanced Topics*. Configuration cloning enables you to recreate a customized operating system on another processor in the event of a disaster, or to recreate an environment on one or more systems.
- This chapter describes only those backup and archiving utilities that are provided in the base operating system when installed.

  The *Associated Products* CD-ROM may include additional backup applications (which may require additional licenses). See the *Installation Guide* for information. See the documentation that comes with your backup application for information on using third-party products.

The main tasks comprising backup and restore operations are:

- Creating your data recovery and disaster recovery plans
- Backing up data, which consists of the following:
  - Choosing a backup schedule

  - Creating small archives by using the `pax`, `tar`, and `cpio` commands or the associated graphical user interface, `dxarchiver`
  - Performing a full UFS backup using the `dump` utility
  - Performing an incremental backup
  - Performing a remote backup
  - Using backup tools
- Restoring data, specifically:
  - Restoring files from small archives
  - Restoring a file system from a dump
  - Restoring a dumped file system on a new partition
  - Restoring files
  - Restoring files interactively
  - Performing remote restorations
  - Restoring standalone systems from bootable tape

9.2 Backing Up Data and System Files

For basic backup, you can use the `dump` and `restore` commands. See `dump`(8) for full details of all command options that are supported. The operating system also provides graphical and command line tools for archiving and for creating a bootable tape of the standalone system (SAS).

Prevention of data loss is an important part of any backup and recovery strategy. There are many tools for system monitoring that you can configure to help prevent situations that may result in data loss. For example, some systems support environmental monitoring, and there are tools to test and exercise peripherals. There are also the event and error logging systems that you can configure to monitor the system for priority events such as a backup failure. See Chapter 13 for information on using the Event Manager to set up the event reporting strategy for your system and site. You can use the Event Manager to report on the success of your backups, ensuring that you do not miss a scheduled backup event.

It is important that all the files on your system, user files and system files, are protected from loss. Back up your entire system, including the system software. Many system files are static; that is, after you install them they remain unchanged. Therefore, you do not need to back up system files as frequently as data files. Incremental backups are also possible, and you may consider implementing them if your data changes significantly in a short period.

Each file system backup is a single process. To ease the backup process, organize your file systems so that dynamic files are on file systems that you back up regularly and static (system or program) files are on file systems that you back up occasionally. You may find that you have dynamic files on file systems that you back up occasionally. If this happens and you need to back them up regularly, just prior to performing a backup, copy the frequently changing files to systems that you back up regularly. This allows you to back up those files without backing up an entire file system. You can write shell scripts to automate these tasks and use the `cron` command to automate the schedule. See `cron`(8) for more information.

9.3 Choosing a Backup Schedule

To decide how often to back up each file system, consider the balance between the potential loss of user time and data and the time it takes you to perform backups. Ask yourself the question, "How much information can I afford to lose?" The answer to this question helps you determine your minimum backup interval. On most systems the backup interval is daily, but you can choose any other interval.

It is not necessary to back up all the files in a file system at each backup. Back up only those files that changed since the previous backup; this is called an incremental backup. Using the `dump` and `restore` commands, you can perform up to nine levels of incremental backups. For example, while a level 0 dump backs up an entire file system, a level 1 dump backs up only those files changed since the last level 0 dump, and a level 7 dump backs up only those files changed since the last lower level dump.

To integrate incremental backups into your file backup schedule, you need to balance the time and tape space required for backup against the amount of time it could take you to restore the system in the event of a system failure. For example, you could schedule backup levels following the 10-day sequence:

[0 1 2 3 4 5 6 7 8 9]

On the first day you save an entire file system (level 0). On the second day you save changes since the first backup and so on until the eleventh day when you restart the sequence. This makes the amount of time spent and data saved on each backup relatively small each day except the first; however, if a system failure on the tenth day requires that you restore the entire system, you must restore all ten tapes.

Most systems follow some variant of the common Tower of Hanoi backup schedule. Once monthly you make a level 0 dump to tape of all the file systems that you backup regularly. Once weekly, you make a level 1 dump to start a daily sequence of:

[...3 2 5 4 7 6 9 8 9 9 ...]

If you do backups only once a day on the weekdays, you end up with a monthly backup schedule as follows:

[0 1 3 2 5 4 1 3 2 5 4 ...]

This schedule, although slightly complex, requires that you restore at most four tapes at any point in the month if a system failure corrupts files. Of course, doing a level 0 dump daily requires that you restore at most one tape at any point, but requires a large amount of time and tape storage for each backup. On most days in the Tower of Hanoi schedule, you require very little time and tape storage space for the backup.

9.4 Backup Methods

Depending on your needs and your local system configuration, there are several options for backing up data, as follows:

- You can run the following command line interfaces from a terminal:
  - `dump`, `rdump`, `restore`, and `rrestore`
  - `tar`, `pax`, and `cpio`

    Use these to create quick file archives or to create scripts that you run with the `cron` scheduler.
- The bootable tape feature, `bttape`, is a SysMan Menu application that you can invoke from the command line, the SysMan Menu, or from CDE. Depending on how you invoke it, it either runs the command line interface or a graphical user interface that is appropriate to your windowing environment. (See Chapter 1 for more information.) The commands are `btcreate` and `btextract`.

  Use the bootable tape feature to create a bootable tape for recovery and to back up critical system data and customized system files. This feature also enables you to use any terminal and a number of windowing environments, and is therefore recommended for remote operations.
- From the CDE folder Application Manager – System Admin, open the Storage Management folder and click on the Bootable Tape icon. This action invokes the graphical user interface to the `bttape` utility.
- From CDE, open the Application Manager pop-up menu from the front panel and open the Desktop_Tools folder to use the following utilities:
  - Archive – For quick archiving of files and folders, such as when archiving projects or user accounts. The related interfaces, Archive List Contents and Archive Unpack, enable you to manage these archives. These are simple graphical interfaces with minimal options.

– From the CDE Application Manager – System Admin folder, open the DailyAdmin folder to use the Archiver utility. The Archiver is a graphical user interface to the command line tools that enables you to select archive type and options such as compression. This interface allows you to drag and drop entire file systems or directories (folders) into the backup set.

Some tools provide you with additional options when you run them as superuser (root).

9.5 Preparing to Perform a Backup

The following sections contain information that you may need to prepare for a backup. Also included is a list of utilities that can assist you in preparing for a backup, and a list of prerequisite tasks.

Chapter 6 contains information on the UFS file system. The *Hardware Management* manual contains information on using disk and tape devices and on determining which disk and tape devices you want to back up. Also, see the information about the `cron` command in Chapter 3 for information on scheduling regular backups. The following sections contain other information that you may need to perform a backup:

9.5.1 Related Documentation

Additional documentation on using the backup utilities is available in manuals, reference pages, and online help.

9.5.1.1 Manuals

These manuals also provide useful information for archival services.

- The *AdvFS Administration* and *Logical Storage Manager* manuals contain information on the AdvFS file system and LSM storage management features.
- The owner's manual for any peripherals used (such as tape drives) contain important information. These documents provide you with information on storage volume, media type, compression densities, and general operating instructions for a device.

9.5.1.2 Reference Pages

Each utility has its own reference page that describes how to invoke the utility and the available options for that utility.

The following reference pages provide information on the basic utilities for dumping file systems to tape and restoring them back to disk:

- `dump`(8), `rdump`(8)

- vdump(8)
- restore(8), rrestore(8)

These reference pages provide information on the basic utilities for creating and manipulating archive files:

- tar(1)
- pax(1)
- cpio(1)

The following reference pages provide information on the bootable tape interfaces:

- btcreate(8)
- btextract(8)
- bttape(8)

The following reference pages provide information on creating cron entries for backup scripts that execute at specific dates and times.

- cron(8)
- crontab(1)

The mcutil(1) reference page describes the media changer manipulation utility.

9.5.1.3 Online Help

Both the Archiver and Bootable Tape graphical user interfaces provide online help that describes your options and defines what data you can enter into the data fields in each window.

9.5.2 System Files

Apart from the file system that you specify and the archive files created, the following files are used or created when you create backups:

- The dump and restore commands create or use the following files:

/etc/dumpdates	Contains a list of file systems that were backed up, the date that each file system was backed up, and the backup level
/tmp/rstdir*	Lists directories stored on the default tape
/tmp/rstmode*	Records the owner, permission mode, and timestamps for stored directories

`./restoresymtab` Holds information required during incremental restore or `rrestore` operations

- The bootable tape feature creates or uses the following files:

  `/var/adm/btcreate.log`

  Provides a log of the `btcreate` process

  `/usr/lib/sabt/sbin/custom_install.sh`

  Specifies which files are added to the miniroot

  `/usr/lib/sabt/etc/addlist`

  A data file that specifies which files and directories are added to the miniroot file system that is created on the bootable tape

  `/usr/lib/sabt/etc/fslist`

  A data file that specifies which file systems are backed up

  `/usr/run/bttape.pid`

  A lock file that prevents multiple instances of the `btcreate` utility

9.5.3 Related Utilities

The following utilities are useful when performing backups:

SysMan Station	The SysMan Station provides a graphical view of the storage devices available on the system. Use this interface to help you identify disk and tape devices and find their device names.
CDE Disk Usage Manager	The CDE Application Manager — Desktop_Tools folder provides a Disk Usage tool that runs the `du` command and returns statistics on disk usage. Use the Folder Size option to examine the size (in blocks) of any directory, such as `/usr/users`. Command line utilities `du` and `df` provide the same data.
CDE Application Manager	DailyAdmin folder provides the System Information interfaces, a graphical view of system resources such as file space usage. You can set this monitor to flash a visual warning when your preset file space limits are exceeded. You can also use the SysMan Station to monitor file systems as described in Chapter 1.

Event Manager	Provides a way of monitoring file system limits and alerts you of problems or can automatically start backups and cleanup of file systems.
`dsfmgr` and `hwmgr`	The command line interfaces dsfmgr and hwmgr enable you to query the system for information about devices, such as device names and disk partition size.

You can get information from the Disk Configuration GUI, which you can invoke from CDE Application Manager - Configuration folder, or from the SysMan Menu. This interface provides size information in megabytes, bytes, and blocks. (The disklabel command provides a command line disk configuration interface).

9.5.4 Prerequisite Tasks

The following prerequisite tasks apply to all the backup methods:

- Become familiar with using the interfaces and the sources of information about the commands. Ensure that this information is available to you when the system is down. Often, you must perform recovery operations in single-user mode, and reference pages may not be available.
- Ensure that all the required products or utilities are installed and configured (if necessary). The simplest way to do this is to see the reference page for information on invoking the tool, and run a test by invoking command line interfaces with null input, or by starting up the graphical user interfaces.
- Verify that the tape hardware is installed and configured. If you are unsure, you can use the `/usr/field/tapex` tape exerciser and see the hardware documentation for other test features. See also the hardware information tools listed in Section 9.5.3.
- Examine the size of the directories that you want to back up. For example, you can use the following commands:

```
# df /usr
Filesystem          512-blocks     Used   Available Capacity  Mounted on
/devices/disk/dsk0g     1498886   688192     660804    52%    /usr

# du -s -x /usr/users
1835      /usr/users
```

  You can use the graphical or command line tools listed in Section 9.5.3.
- Obtain sufficient quantities of the correct media, ensuring that there is enough storage volume for the files that you intend to back up. This also applies if archiving to disk or any other writeable media, such as WORM drives or magneto-optical diskette drives.

- Identify the files or directories that you intend to work with, and choose appropriate names for the archives. You may need some temporary scratch disk space if assembling different directories into a single volume before archiving (although you can do this directly to the archive from the command line or by adding directories to existing archives). See the documentation for the backup utility that you choose to use. Some tools provide default file names and locations. For example, the bootable tape interface prompts you for the following file names. (You can accept the default or provide another file name):

  `/usr/lib/sabt/etc/fslist`

  A data file that specifies which files and directories are added (appended) to the miniroot

  `/usr/lib/sabt/etc/addlist`

  A data file that specifies which file systems are backed up

  The Archiver requires the following:

  – One or more source files or directories. In CDE, directories are identified as folders, and you can drag and drop them into the Archiver window from File View windows instead of entering long pathnames such as `/usr/lib/sabt/sbin`.

  – A destination file, such as `/usr/backups` for a tar file on disk, or the device name for a tape device, such as `/dev/tape/tape0_d0`. (You do not need to supply an extension or suffix for the archive file name. The utilities listed in Section 9.5.3 can assist you in finding the required device information, particularly if more than one tape drive is attached to a system.)

  – The archive name (for example, `/usr/archives/user-files_990802.Z` or `/dev/tape/tape0_d0` for a tape archive) if you are restoring (that is, unpacking) an archive.

- The device name for the device or devices that you want to access, and any associated device special file. For example, the following are valid device names and device special files:

Device name	Device Special File	Description
`dsk0a`	`/dev/disk/dsk0a`	Partition a of disk number 0
`disk1b`	`/dev/rdisk/dsk1b`	Partition b of raw disk 1

Device name	Device Special File	Description
tape0c	/dev/tape/tape0c	Default density rewind tape (with compression)
tape0_d0	/dev/ntape/tape0_d0	Nonrewind tape device 0. The _d0 suffix specifies the density

Device names are located in the /dev directory under the /disk, /rdisk, /tape, or /ntape subdirectories. Also, you can use the graphical or command line tools listed in Section 9.5.3 to locate devices and match them with their device names.

Note

Tape devices often support different densities and compression options that enable you to put more information into a single archive. See tz(7) for information on tape density options, and how you select them by specifying different device names.

- Full backups may require that you shut down the system. You can back up the system while in either multiuser mode or single-user mode. However, backups performed on file systems actively being modified may corrupt the backup data. The dump command operates by verifying the inodes of the files you want to back up. The inodes contain data such as table entries and other statistics. If you use the dump command to back up files in a file system, an inode is attached to each file. If the system or user activity changes a file after the inode data is recorded, but before the file is backed up, it may corrupt the backup.

  To shut down the system, unmount a file system, and verify the integrity of a file system:

  1. Shut down the system by using the SysMan Menu General Tasks option, or with the /usr/sbin/shutdown command. For example, to shut down the system in 5 minutes and give users periodic warning messages, enter:

     ```
     # /usr/sbin/shutdown +5 'System going down to perform backups'
     ```

     See Chapter 2 for more information on shutting down the system.

  2. Use the umount command with the -a option to unmount the file systems that you want to back up:

     ```
     # /sbin/umount -a
     ```

     The root file system remains mounted.

  3. Use the fsck command to ensure the integrity of the file system.

For example, use the following command to verify a file system on the c partition (the whole disk):

```
# /sbin/fsck -o /dev/disk/dsk0c
```

9.6 Using the dump Command

The dump command copies all designated file systems or individual files and directories changed after a specified date to a file, pipe, magnetic tape, disk, or diskette. See the *AdvFS Administration* manual for information on copying AdvFS file systems. You must have superuser privileges to use the dump command.

Note

To produce valid backups on a file system, you must back up a file system while it is inactive. It is recommended that you unmount the file system and examine it for consistency. As an added precaution, put the system into single-user mode before starting your backup operations. This is not true for AdvFS.

9.6.1 Performing a Full Backup

Set up a schedule for performing a full backup of each file system on your entire system, including all the system software. A conservative schedule for full system backups is to do one with each normal level 0 dump (using Tower of Hanoi, once a month), but you can set any schedule you like within the reliability of your storage media, which is about two years for magnetic tapes. To back up your file system, use the dump command. See dump(8) for a description of the command options that you use to specify the characteristics of the tape device, such as block size, tape storage density, and tape length. Specify the file system with a full pathname when you use the dump command. The dump command can back up only a single file system at a time, but there may be several dump processes simultaneously writing files to different tape devices.

The following list describes the most commonly used options to the dump command:

-integer — Specifies the dump level as an integer (0-9). A dump level of 0 causes a full dump of the specified file system. All other dump levels cause an incremental backup. That is, only files that have changed since the last dump of a lower dump level are backed up. The /etc/dumpdates file contains a record of when the dump command was used on each file system

	at each dump level. The `-u` option to the `dump` command updates the `dumpdates` file.
`-f` *dump_file*	Writes the dump to the device specified by *dump_file* instead of to the default device, `/dev/tape/tape0_d0`. If you specify the *dump_file* as a dash (-), the `dump` command writes to the standard output.
`-u`	Updates the `/etc/dumpdates` file with the time of the dump and the dump level for the file system in the backup. You use this file during incremental dumps (by using the dump level option) to determine which files have changed since a particular dump level. You can edit the `/etc/dumpdates` file to change any record or fields, if necessary. See `dump`(8), which describes the format of this file.

To back up your entire file system to the default backup device, use the `dump` command for each file system on your machine. The `dump –0u` command option causes a level 0 dump and updates the `/etc/dumpdates` file with the time and date of the backup for each file system. This creates an initial point on which to base all future incremental backups until the next full or level 0 dump. Each file system must be backed up individually.

For example, if you want to perform a level 0 dump of the root, `/usr`, and `/projects` file system partitions, follow these steps:

1. To back up the root file system:
   a. Load a tape into your tape drive.
   b. Enter:

      # **dump -0u /**

   c. Remove the tape from your tape drive after completing the backup.
2. To back up the `/usr` file system:
   a. Load a new tape into your tape drive.
   b. Enter:

      # **dump -0u /usr**

   c. Remove the tape from your tape drive after completing the backup.

3. To back up the /projects file system:

   a. Load a new tape into your tape drive.

   b. Enter:

      # **dump -0u /projects**

   c. Remove the tape from your tape drive after completing the backup.

You can either back up each file system on an individual tape, or you can back up multiple file systems on one tape by specifying the no-rewind device, /dev/ntape/tape0_d0, as the output device. The following examples show the root, /usr, and /projects file systems being backed up on one tape:

```
# dump -0uf /dev/ntape/tape0_d0 /
# dump -0uf /dev/ntape/tape0_d0 /usr
# dump -0uf /dev/ntape/tape0_d0 /projects
```

This example may require additional media management to cross-reference dump files with tapes, especially when a single dump file spans media. Exercise care when labeling this type of backup media.

9.6.2 Performing an Incremental Backup

Set up a routine as part of your backup schedule to make it easier to remember which backup to do each day. Include a mechanism for logging your backups and their dump level and for listing the tapes on which they are made. Because of the chance of system corruption, do not keep this information on the local computer system.

After you establish a system for making incremental backups, the procedure is simple. Assume you use the following backup schedule to do a daily backup of /usr:

```
0 1 9 9 9 1 9 9 9 9 …
```

On Monday, perform a level 0 dump:

```
# dump -0u /usr
```

On Tuesday, perform a level 1 dump:

```
# dump -1u /usr
```

The level 1 dump backs up all the files that changed since Monday. On Wednesday through Friday, perform a level 9 dump (which always backs up all the files that have changed since Tuesday's level 1 dump):

```
# dump -9u /usr
```

To perform the same level 9 dump to the tape device named /dev/tape/tape1_d0 instead of the default tape device, use the -f option as shown in the following example:

```
# dump -9uf /dev/tape/tape1_d0 /usr
```

The argument to the -f option specifies a tape device local to the system from which you are performing the dumps.

9.6.3 Performing a Remote Backup

Some machines in a networked system environment may lack a local tape drive that you can use for making backup tapes. You can use the rdump command to make backups on a remotely located tape device. The rdump command is identical to the dump command except that it requires the -f option to specify the machine name and an attached backup device. See dump(8) for a description of the options to the rdump command.

The rdump command updates the /etc/dumpdates file on the local machine in the same way as does the dump command. The rdump command starts a remote server, /usr/sbin/rmt, on the remote machine to access the storage medium. This server process is transparent. See rmt(8) for more information.

To back up the /projects file system from bhost1 onto a tape drive on bhost2 with the attached backup device /dev/rmt0h, enter the following command from bhost1. The name of bhost1 must be defined in the /.rhosts file of bhost2 to allow access.

```
# rdump -0uf bhost2:/dev/tape/tape0_d0 /projects
```

9.6.4 Using Backup Scripts

You can automate the backup process by using shell scripts. The cron daemon can execute these shell scripts late in the evening when there is less chance of the dump commands making errors from a changing system.

Backup shell scripts often perform the following tasks:

- Determine the dump level
- Warn the system of the dump
- Make a listing of tape contents
- Notify the operator upon completion

Some time during the day, load a tape into the tape drive. At the specified time, the cron daemon runs the backup shell scripts. After the shell procedures are finished, remove the backup tape and archive it.

Backup shell scripts are best used when the dump is small enough to fit on a single tape. You must specify the no-rewind device and the -N option to the dump command to inhibit the tape from going off line automatically when each dump is completed. After the dump command reaches the end of the tape, it takes the tape off line and you must replace the tape.

9.7 Restoring Data

Occasionally, you need to retrieve files from your backup tapes, and possibly need to restore entire file systems at some time. If you have set up a good backup procedure, then restoring files or full file systems is a simple task.

If a serious problem occurs, you may have to restore your entire system. Before restoring, determine what caused the problem with the system.

After determining the cause of the problem, reinstall your system from the initial boot tapes. The installation instructions that came with your system explain this procedure.

After your system is up and running, restore the system to the state it was in just prior to the system crash. If you are using AdvFS, use the `vrestore` command. See the *AdvFS Administration* manual for information on restoring the AdvFS file system. If you used the `vdump` command to back up a UFS file system, you can use the `vrestore` command to recover it. However, if you used the `dump` command you must use the `restore` command to recover files. Because the `dump` command saves only a single file system at a time, you must execute the `restore` command for each file system you want to restore. See `restore`(8) for information on the command syntax.

9.7.1 Restoring a File System

A general procedure for restoring a file system, such as after a disk failure or other loss of data, is described here. To restore individual files, go to Section 9.7.2.

To restore a file system, create a new file system and restore the files from the dump files by using the following commands:

`newfs`	Creates a new UFS file system. See `newfs`(8) for more information.
`mount`	Mounts the file system, making it available for general use. See `mount`(8) for more information.
`cd`	Changes your current working directory. See `cd`(1) for more information.
`restore`	Restores archived files from a backup media to a disk. See `restore`(8) for more information.

See the *AdvFS Administration* manual for information on restoring an AdvFS file system.

If the disk does not have a label, write the label by using the `disklabel` command before you create the new file system. See `disklabel`(8) for more information.

Writing a label with customized partition table settings may affect the entire disk. Use the following command to write the default disk partition table:

```
# /sbin/disklabel -rw dsk1
```

Invoke the editing option of the `disklabel` command to use the customized partition table settings. See Chapter 6 for more information. You can use the Disk Configuration interface. See `diskconfig`(8) for more information.

The following example shows the commands you use to restore a file system called `/usr/projects` from a tape:

```
# disklabel -rw dsk1
# newfs /dev/rdisk/dsk1c
# mount /dev/rdisk/dsk1c /usr/projects
# cd /usr/projects
# restore -Yrf /dev/tape/tape0_d0
```

9.7.2 Restoring Files Manually

If users lose data files, they ask their system administrator to restore those files. Users also may ask you to restore an earlier version of a file. Whatever the reason for a file restoration, you must determine which tape contains the correct version of the file. Inquire when the file was lost and when it was last modified, you can use your backup log to determine which tape contains the most recent version of the wanted file.

Use the `-t` option with the `restore` command to determine whether a file is on the selected tape. The `-t` option creates a list of files and directories on the tape. For example, to list the contents of the `working` subdirectory of the `/usr` file system on a particular backup tape, load the tape and enter:

```
# restore -t ./working
```

To create a list of the entire contents of a backup tape, load the backup tape and enter:

```
# restore -t
```

Make a listing of each backup tape after you create it. This verifies a successful backup and gives you a place to look up what files are on the tape.

After determining the location of the file, create a new directory for the file. If you restore the file into an existing directory and the file already exists, the restored file overwrites the existing file.

For example, to restore the `working/old.file` file from a `/usr` file system backup tape into your current directory, load the backup tape and enter:

```
# restore -x ./working/old.file
```

To restore the entire contents of the working subdirectory from the same tape, enter:

```
# restore -x ./working
```

If your dump media contains multiple dump images, you need to know the sequence of the dump images in order to restore a file from one of the images. To examine the contents of the first dump image on the media, load the tape and enter:

```
# restore -ts 1
```

The `-s` option followed by the number `1` specifies the first dump image.

For example, to restore the `working/old.file` file from a `/usr` file system, which is the third dump image on the backup tape into your current directory, load the backup tape and enter:

```
# restore -xs 3 ./working/old.file
```

9.7.3 Restoring Files Interactively

To ease the task of restoring multiple files, use the `-i` option to the `restore` command. This option starts an interactive `restore` session. The interactive mode has commands similar to shell commands.

To begin an interactive `restore` session, enter:

```
# restore -i
```

The system responds with the following prompt:

```
restore >
```

The following command line options are available in the interactive restore mode:

`ls` [*directory*]	Lists files in the current or specified directory. Directory entries end with a slash (`/`). Entries that are marked for reading begin with an asterisk (`*`).
`cd` [*directory*]	Changes the current directory to the directory specified by the *directory* argument.
`pwd`	Lists the pathname of the current directory.
`add` [*files*]	Adds the files in the current directory or the files specified by the *files* argument to the list of files recovered from the tape. Files are marked with an asterisk (`*`) if they are identified as “to be read” by

	the `add` command. You see this asterisk when you use the `ls` command to list files.
`delete` [*`files`*]	Deletes all the files in the current directory or the files specified by the *`files`* argument from the list of files recovered from the tape.
`extract`	Restores from the tape the files that are marked "to be read" into the current working directory. The `extract` command prompts you for the logical volume that you want to mount (usually 1), and whether the access modes of the dot (`.`) current directory are affected; answer `yes` when you are restoring the entire `root` directory.
`setmodes`	Sets owner, access modes, and file creation times for all directories added to the files-to-read list; no files are recovered from the tape. Use this command to clean up files after a `restore` command is prematurely aborted.
`verbose`	Toggles verbose mode. In verbose mode, each file name is printed to the standard output. By default, verbose mode is set to off. This is the same as the `-v` command line option to the `restore` command.
`help`	Lists a summary of the interactive commands.
`?`	Lists a summary of the interactive commands.
`what`	Lists the tape header information.
`quit`	Quits the interactive restore session.
`xit`	Exits from the interactive restore session. The `xit` command is the same as the `quit` command.

To interactively restore the `./working/file1` and `./working/file2` files from a backup tape, load the tape and enter:

```
# restore -i
```

After you switch to interactive mode, follow these steps to add the files to the list of files that you want to extract:

1. Change to the `working` directory:

   ```
   restore > cd working
   ```

2. Enter the file name at the prompt:

   ```
   restore > add file1
   ```

3. Enter the name of the second file as follows:

   ```
   restore > add file2
   ```

4. Extract the files as follows:

   ```
   restore > extract
   ```

5. You are prompted for the logical volume you want to mount; usually you respond to this prompt with `1` as shown in the following example:

   ```
   You have not read any tapes yet.
   Unless you know which volume your file(s) are on you can start
   with the last volume and work towards the first.

   Specify next volume #: 1
   ```

   You are then asked whether the extract affects the access modes of the dot (.) current directory. For this example, reply with `n`.

   ```
   set owner/mode for '.'? [yn] n
   ```

6. Quit the interactive session after the files are extracted:

   ```
   restore > quit
   ```

The `file1` and `file2` files are now in the current directory.

You can automate this procedure in a command file that is read by the `-F` option to the `restore` command. For example, the following command file, named `restore_file`, performs the restore operation shown in the previous example:

```
cd working
add file1
add file2
extract
1
n
quit
```

To read and execute this shell script, enter the following command:

```
# restore -iF restore_file
```

The result of the procedure in this script is identical to that of the previous interactive restore session.

9.7.4 Restoring Files Remotely

You use the `rrestore` command to restore files to local directories from a remote tape device. The `rrestore` command requires the `-f` option to specify the machine name and its backup device. See `rmt`(8) for more information and Section 9.7 for a description of the options to the `rrestore` command.

You must specify the name of the remote system where the backup device is attached, and the name of the backup device on that remote system in the format *system:device*.

To restore the `./working/file1` file onto the local directory on system1 from a backup tape mounted on system2 where the backup device `/dev/rmt0h` is attached, enter the following command from system1. The name system1 must be in the `/.rhosts` file of system2 to allow access from system1 to system2.

```
# rrestore -xf system2:/dev/tape/tape0_d0 ./working/file1
```

The `rrestore` command starts a remote server, `/usr/sbin/rmt`, on the remote system to access the storage medium.

9.7.5 Restoring or Duplicating a System (Root) Disk

In previous versions of the operating system, device names were assigned based on the physical location of the drive, according to the SCSI bus target. In Version 5.0 and higher, device names are assigned logically and stored in a database. They have no relationship to the bus address of the device. The device database must be recovered and possibly updated to successfully restore the root file system or if you want to move the root disk to a disk with larger capacity. Also, you may need to install devices (such as a tape device) to the device database when you restore the device from tape backup media.

After you reboot the system during the restoration, you may see the following message:

```
Unable to save existing hardware configuration.
New configuration will be used
```

This message indicates that the device database is not recoverable and you must restore it.

The following procedure is a generic method for recovering or duplicating (cloning) a root disk. It covers the following possible scenarios:

- The disk and the root partition are not damaged but you want to replace it with a different disk, possibly a different model with larger capacity.
- The disk drive on which the root partition is located is damaged and you must:

  - Install a new disk drive, possibly of a different type and capacity
  - Choose an alternate disk drive that is installed and available for use
- The root (/), and possibly the /usr or /var file systems are corrupt, but the disk drive on which they are located is fully functional.

Note

This procedure does not specifically address recovery methods from network backups and it does not address recovery of an AdvFS file system. See *AdvFS Administration* for more information.

Depending on your knowledge of your system, you may not need to read all the following sections:

- How you prepare for a recovery in Section 9.7.5.1
- The requirements for recovery in Section 9.7.5.2
- The recovery (or duplication) procedure in Section 9.7.5.3
- Alternate procedures in Section 9.7.5.3

9.7.5.1 Preparing for Recovery or Duplication

Depending on how your system is set up, and your level of system knowledge, you may need the following:

- A replacement root disk drive. This procedure assumes that if the original root disk is unusable, you have either installed a new replacement disk, or you have decided to use an alternate disk that is already installed in your system. Install the drive as described in the owner's manual for the drive. The operating system automatically detects the drive.

  There are steps in the procedure that assist you in identifying the new or alternate drive.

- Firmware update.

  Verify that your system's firmware is current. You can obtain information and download kits from the *Firmware Updates* web page at the following URL:

  `http://ftp.digital.com/pub/DEC/Alpha/firmware`

  (You can also go to the web site at **http://www.hp.com** and select the Support option to search for information.)

- Information about console commands.

You use Alpha System Reference Manual (SRM) console commands at the system console prompt (>>>) to perform some tasks. These commands are documented in the owner's manual for your AlphaServer system.

If you cannot find the printed document, look for a printable file on a CD-ROM supplied with the system. If the CD-ROM is unavailable, you can find the documentation at the *Alpha Systems Technology* web page at the following URL:

http://www.compaq.com/alphaserver/technology/index.html

- The procedure instructions are typical for newer processors. If your system is older, you must see the owner's manual and *Installation Guide* for your version of the operating system to obtain the actual commands and procedures.

The status of your system must be as described in Table 9–1:

Table 9–1: Recovery Preparation

Requirement	Description
A Full and Recent Backup	You need a full backup of all operating system file sets that are on the root volume. This may include root (/), /usr, and /var.
System Configuration	This procedure applies to all configurations where there is a single disk drive used for the root partition, which may contain the /usr and /var file systems. You need a functional disk drive to contain the restored root volume. This disk must have a minimum storage space as defined in the operating system limits for the restored release. The restore device (typically a tape drive) must be local and not a remote backup device.
Logical Storage Manager	If you are using the Logical Storage manager (LSM), see the *Logical Storage Manager* manual for information on recovering the root volume.
User Interface	This procedure requires a console login.
Affect on System Availability	Except on clustered systems, loss of the root disk invariably involves one or more shutdowns and reboots of the system. This procedure is intended to help you restore full operation as quickly as possible. The time required for duplicating or recovering a disk depends on the disk size.
Privileges	You must be a root user with physical access to the system's storage array and backup devices

9.7.5.2 Determining the Restoration Requirements

You may need the following resources to complete the restoration of your root disk. If you are very familiar with your system's configuration, or if you have a recovery plan which records all the information you need to perform a recovery, you do not need to read this section. You may need the following items:

- Distribution Media for the Operating System

  You use the installation shell to restore the root disk. The installation shell is a compact version of the operating system from which you can execute commands, such as `mount`. The shell is packaged with the operating system software as part of the distribution kit.

  Your local site may provide a Remote Installation Service (RIS) server from which you can boot your system across the network. If RIS services are available in place of CD-ROM media, follow your site-specific procedures and see the *Installation Guide*.

- CD-ROM Drive Name or Network Device Name

  To restore the root volume, you boot your system from a CD-ROM drive or a network device. You may need to see the owner's manual for your system to find the correct commands. Typically, you determine the CD-ROM device name at the console prompt as follows:

  ```
  >>> show device | grep -E 'RR|CD'
  dka400.4.0.5.0  DKA400  RRD47   1206
  ```

  You typically determine the network device name as follows:

  ```
  >>> show config | more
  ```

  After you enter the preceding command, the complete system configuration is displayed one page at a time. Scroll down to the section headed `Slot Option` and locate the network device. Network devices are typically named `ew*` or `ei*`, where `*` is a letter. For example:

  ```
  11   DE500-BA Network Con  ewa0.0.0.0.11.0   08-00-99-1Z-67-BB
  ```

  For information on booting your system from a RIS server, see the *Installation Guide — Advanced Topics* manual. For a full discussion of the System Reference Manual (SRM) console device naming conventions, see the owner's information for your system.

- Boot Device Name

  Determine the default boot device name as follows:

  ```
  >>> show bootdef_dev
  bootdef_dev            dka0.0.0.5.0
  ```

  In this example the default boot device is `dka0`.

Note

If you are using Fibre Channel, the name of the boot device is as you defined it during configuration of the storage devices.

If the current root device is usable and you are restoring to the same device, you use the device name later in the restore procedure. If you intend to install a new disk or use an alternate, you must specify the name of the disk. You determine the alternate by translating its *b/t/l* into the boot device name during the restore procedure.

- Backup Media and the Restore Device

  Depending on what file systems were on your original root disk, you may need full and current backup tapes for the root (`/`), `/usr` and `/var` file systems.

  In cases where you are duplicating (cloning) a disk, such as to increase the disk space available by using a disk with larger capacity, you can opt to back up directly from the source disk to the target disk.

- Disk Label for the Target Disk

  If the original drive is usable, you can choose to restore the root file system to the same drive. If the drive is damaged, you must select an alternate drive or install a new drive. The alternate or new drive must have enough storage capacity to hold the restored file systems and it must be partitioned to hold the restored file systems.

  If the original root drive held custom partitions, restore the custom disk label or at least plan to select partitions that can adequately contain the restored file systems (and possibly allow for future expansion of those file systems). Depending on what data is stored on the original disk, you may need to plan for the following partitions:

  – The 256MB `a` partition to hold the root (`/`) file system.

  – If the `/usr` file system is on the root drive and you also need to restore it, you need a partition at least large enough to contain the restored file system. (Consider expansion requirements, if appropriate.)

  – If the `/var` file system is on the root drive and you also need to restore it, you need a partition at least large enough to contain the restored file system unless it is included in `/usr` on the original root drive.

  – If primary or tertiary swap partitions were on the original root drive, you must recreate these partitions on the replacement drive.

  You can restore other file systems as required, or restore them to different devices and then remount them by updating the `/etc/fstab` file after the restoration is complete.

See `disklabel`(8) for more information on creating a disk label with custom partitions.

9.7.5.3 Applying the Procedure

Some steps in the procedure are dependent on your system's original configuration. Ignore these steps if they do not apply to your configuration. The optional steps are marked [Configuration Dependent].

In the procedure, you always proceed to the next step unless redirected.

1. Boot the system from the operating system distribution media by using one of the following methods:

   - Insert the distribution CD-ROM that contains the operating system version that you want to restore and boot the operating system specifying the CD-ROM reader device name that you determined previously. For example:

     ```
     >>> boot dka400
     ```

   - Boot from your local RIS server. For example:

     ```
     >>> boot ewa0
     ```

2. [Configuration Dependent] If you are already using the character-cell installation procedure, go to step 3, otherwise complete the following task.

   If your system has a graphics console, the installation defaults to graphical mode. Wait until the installation procedure displays a dialog box titled Installation Welcome.

   Pull down the File menu and select `Exit` to invoke character-cell mode.

3. Verify the status of the backup device and the target disk (the restored root disk) by using the following command:

   ```
   # hwmgr view devices
   ```

   The `hwmgr` command displays a list of all devices currently recognized by the system as shown in the following example:

   ```
   HWID: Device Name         Mfg      Model            Location
    ------------------------------------------------------------------------
       4: /dev/kevm
      28: /dev/disk/floppy0c          3.5in floppy      fdi0-unit-0
      31: /dev/disk/dsk0c    DEC      RZ26L     (C) DEC  bus-0-targ-0-lun-0
      32: /dev/disk/dsk1c    DEC      RZ26      (C) DEC  bus-0-targ-1-lun-0
      33: /dev/disk/dsk2c    COMPAQ   HB00931B93         bus-0-targ-3-lun-0
      34: /dev/disk/cdrom0c  DEC      RRD45     (C) DEC  bus-0-targ-4-lun-0
      35: /dev/disk/dsk3c    COMPAQ   HB00931B93         bus-0-targ-5-lun-0
      37:                    DEC      TLZ06     (C)DEC   bus-0-targ-6-lun-0
   ```

   Locate and write down the following data:

   - The device name for the target disk.

This is important if you installed a new replacement disk. The device name is the entry under the `Device Name` column, such as `/dev/disk/dsk2`. Ignore the partition suffix (`c`).

If there is no entry for a newly installed target disk, you cannot proceed. You must shut down the system, verify the disk's physical installation, and restart the recovery procedure.

- The device name for the backup device.

  The device name appears in the `Device Name` column. For example, if you are restoring the root disk from the default tape device `TLZ06` the `Device Name` column must contain a device special file name such as `/dev/ntape/tape0`.

  There may be no device special file name in the `Device Name` column for the backup device, as shown in the preceding example. In this case, go to step 4 to install the tape device.

4. Install the backup device by using the following command:

   ```
   # dn_setup -install_tape
   ```

   To verify the installation and determine the device name (such as `tape0_d0`), repeat the `hwmgr` command in step 3.

5. [Configuration Dependent] If the original file system format is unknown, you can now ascertain it and verify that you have a readable backup tape as follows:

   a. Load the backup (dump) media into the device.

   b. Invoke the interactive mode of the `restore` command, specifying the backup device name that you determined in step 4. For example:

      ```
      # restore -i -f /dev/ntape/tape0_d0
      ```

   c. If the backup is good, a prompt for interactive restoration is displayed. Enter the `what` command to display the header and record the information.

6. Create and apply a disk label by using the following information:

   a. The partition plan that you created during recovery planning, including any swap space requirements. (See Section 9.7.5.2.)

   b. The new root device name determined in step 3.

   Specify the `a` partition and label the drive as a bootable device. For example:

   ```
   # disklabel -wr /dev/disk/dsk2a
   ```

7. Create your UFS target file systems as follows:

You must create file systems on the new root drive for each file system that you need to restore. For example, to create the new root and `/usr` file systems on partitions `a` and `g`, use commands similar to the following:

```
# newfs /dev/disk/dsk2a
# newfs /dev/disk/dsk2g
```

8. Mount the replacement disk on the temporary mount point `/mnt` according to the type of file system. For example:

   ```
   # mount /dev/disk/dsk2a /mnt
   ```

9. Use the `vrestore` or `restore` command to restore files. For example:

   ```
   # cd /mnt
   # vrestore -x device
   ```

10. Shut down and halt the system by using the following command:

    ```
    # shutdown -h now
    ```

11. Boot the system to single-user mode, specifying the restored root drive as the boot device. For example:

    ```
    >>> boot dka2 -flags s
    ```

    If you are using an alternate drive, or if you installed a new drive, you may need to translate the system device name to the appropriate boot device name. In step 3, you used the `hwmgr` command to determine the device database entry for the new device. For example:

    ```
    33: /dev/disk/dsk2c COMPAQ HB00931B93 bus-0-targ-3-lun-0
    ```

    Use the following command to display the devices:

    ```
    >>> show device
    ```

    Map the value of the *b/t/l* (in this case 0.3.0) to the alternate or new device and identify its boot device name, such as `dka300`.

12. If the boot is successful, run the following script to update the device database:

    ```
    # /sbin/mountroot
    ```

    While the `dsfmgr` command attempts to update the device database, some error or warning messages may be displayed. You can ignore the messages.

13. [Configuration Dependent] If you installed a new drive for root, or you specified an alternate device, you need to rename devices. Using the device name information that you determined in step 3, rename the devices as follows:

- If you remove the old root disk and replace it with a new device, use the `dsfmgr` command with the `-m` option to move the device names. For example:

  ```
  # dsfmgr -m dsk20 0
  ```

- If the old root disk is still connected to the system, use the `dsfmgr` command with the `-e` option to exchange the device names. For example:

  ```
  # dsfmgr -e dsk20 0
  ```

14. Using the interactive mode of the `vrestore` command, load the backup media into the restore device and restore the device directories. This step ensures that all appropriate devices, including any custom device drivers are recreated:

    - Delete the existing directories as follows:

      ```
      # rm -rf /cluster/members/member0/dev*
      ```

    - If you used the `dump` command, restore the directories as follows:

      ```
      #  restore -i -f /dev/ntape/tape0_d0
      restore >  add /cluster/members/member0/dev
      restore >  add /cluster/members/member0/devices
      restore >  extract
      ```

15. Use the `dsfmgr` command to verify the device databases and device special file names as follows:

    ```
    # dsfmgr -v
    ```

16. This is the end of the procedure. Assuming that you have restored all the required file systems, including `/usr` and (if necessary) `/var`, you can now shut down the system, redefine the boot device, and reboot the system to multiuser mode as follows:

    ```
    # shutdown -h now
    >>> set bootdef_dev dka300
    >>> boot
    ```

You can verify success by examining the boot process for any error messages relating to devices. If you determine that the procedure is not successful, your only option is to reinstall the operating system from the distribution media and recreate your customized environment.

9.7.5.4 Using Alternative root Disk Duplication Methods

Other methods of restoring or duplicating the root disk depend on whether or not you configured your system in anticipation of such problems. The following methods are available:

- Methods of duplicating the root disk, such as mirroring, which are available when LSM and AdvFS are in use. See the *AdvFS Administration* and *Logical Storage Manager* manuals for more information.

  See the *Logical Storage Manager* manual for information on setting up a recoverable root volume.

- Bootable tape provides a method of restoring customized systems that is faster than a conventional backup tape. This feature is available only on certain configurations. See Chapter 9 for more information.

- Backing up your customized system files can assist recovery if you have to reinstall the operating system. If you used the update installation feature (`updateinstall`) for the previous installation, you can create an archive of customized system files. See the *Installation Guide* for more information.

9.7.6 Restoring the /usr and /var File System

You may need to restore the root file system as described in Section 9.7.5 before you can restore the `/usr` file system. If the `/var` directory is on a file system other than `/usr`, repeat the steps in this section for restoring the `/var` file system.

The procedure in this section requires that you have access to the most recent dump files of your `/usr` file systems. The following steps show how you restore from a level 0 dump of files, by using the text-based (or character cell) interface to perform the task:

1. Label the disk if required by using the `disklabel` command as follows:

   ```
   # disklabel -rw /dev/disk/dsk0
   ```

   Note

   The options used with the `disklabel` command in this procedure write a default disk partition table to the disk. If the disk you are restoring has a customized partition table, invoke the editing option of the `disklabel` command or restore the partition table from your protofile label. See Chapter 6 and `disklabel`(8) for more information.

2. Create a new file system by using the `newfs` command. For example:

   ```
   # newfs /dev/rdisk/dsk1c
   ```

3. If necessary, create mount points by using the `mkdir` command. For example:

   ```
   # mkdir /usr
   ```

4. Mount the file system by using the mount command. For example, to mount the file system created in the previous step, enter:

   ```
   # mount /dev/disk/dsk1c /usr
   ```

5. Restore the file system:

   - If you are restoring dump files from a local file system, change to the restore directory, insert the medium containing the dump file, and enter the `restore` command. For a tape, you may enter the following commands:

     ```
     # cd /mnt
     # restore -Yrf /dev/tape/tape0_d0
     ```

   - If you are restoring dump files from a remote system, change to the restore directory and use the `rsh` command. You may need to specify the following command options:

remote_hostname	The host name of the remote system that contains the dump file
dumpfile	The full pathname of the dump file on the remote system.
blocksize	The block size is required to read data from a tape.

     Read the dump file using the same block size that you specified when you wrote the tape. The default dump record size is 10 KB.

     For example, to restore a dump file on a TLZ06 from the remote system `remotesystem` that you wrote by using the default block size, enter the following:

     ```
     # cd /mnt
     # rsh remotesystem  "dd if=/dev/tape/tape0_d0 bs=10k" \
     | restore -Yrf -
     ```

9.8 Using the Command Line Utilities: tar, pax, and cpio

The `tar`, `pax`, and `cpio` command line utilities provide a method of quickly creating an archive from the command line or for writing scripts to back up

files. The disadvantage is that you may have to type long command strings, and backing up or restoring large volumes of files and directories is not easy when using these interfaces. You may use these utilities to make a small archive of files for distribution to other users, such as a program, its sources, and associated documentation.

The following examples demonstrate how you can create or restore typical archive files by using the `tar`, `pax`, and `cpio` command line utilities.

Using tar to Create an Archive

The `tar` command saves and restores multiple files on a single device such as a disk or tape.

To use the `tar` to create an archive on tape drive `/dev/tape/tape12_d0`, enter a command such as the following

```
# tar cvfb /dev/tape12 -e ./.netscape -C /usr/glenn
```

The resulting archive contains all files and directories in the `/usr/glenn` directory, except for file `./.netscape`. See `tar`(1) for more information.

Using pax to Create an Archive

The `pax` command extracts, writes, and lists members of archive files. It also copies files and directory hierarchies.

To use the `pax` command to create an archive of the current directory to device `/dev/tape/tape0_d0`, enter:

```
# pax -w -f /dev/tape/tape0
```

The following command reads the archive `a.pax`, extracting all files rooted in the `/usr` directory, relative to the current directory:

```
# pax -r -s ',^//*usr//*,,' -f a.pax
```

See `pax`(1) for more information.

Using cpio to Create an Archive

The `cpio` command copies files between archive storage and the file system. It is used to save and restore data from traditional format `cpio` archives.

To use the `cpio` command to create an archive to tape device `/dev/tape/tape12_d0`, enter:

```
# cpio -ov < file-list -O/dev/tape12_d0
```

See `cpio`(1) for more information.

9.9 Using dxarchiver

The Archiver, `dxarchiver`, is a graphical user interface (GUI) for the command line utilities described in Section 9.8. Use this interface to:

- Copy and store multiple files to a single, named archive file or output device such as a tape or diskette
- Uncompress incoming archive files and compress newly created files
- Retrieve stored files from an archive file or device such as a tape or diskette

Because the Archiver GUI is a CDE application, you can drag and drop files and directories (folders) to assemble an archive set, without having to type long commands.

It is assumed that you gathered the information Section 9.5.4, and you have loaded or unloaded a tape or other media into the target device as described in the owner's manual. To create an archive, proceed as follows:

1. Invoke the `/usr/bin/X11/dxarchiver` GUI from a terminal command line, or open the CDE Application Group: System_Admin. Then open System Admin Subgroup: Daily Admin and click on the Archiver icon.
2. Select the Archive Type: `tar`, `cpio`, or `pax`. Not all command line options are available from the graphical user interface.
3. Select any Archive Options. You can append only to an existing archive, and you cannot further compress an existing archive that was compressed on creation. Specify either an absolute or a relative pathname as the method of storing the directories. (An absolute pathname is the full path, beginning at the root directory such as `/usr/users`. A relative pathname begins at the current directory, for example dot (`.`) or `users/chan`.)

   During future recovery of these files, you can write them to a temporary location only if you specified a relative path during the original archiving process. Otherwise, files are restored to their original locations. (Potentially overwriting the existing version of the file unless you rename it.)
4. Specify the source, the files, and directories to archive. You can type pathnames or you can open a File Manager view and drag files and directories (CDE folders) to the Source Container box within the Archiver window. If you type pathnames, select `OK` to add them to the container.
5. After all required files are specified, choose the Archive... option and the Archiver: Archive window is displayed.
6. Enter a destination path, such as:

- `/dev/tape/tape0_d0` for the default tape device.
- `/usr/backup/myback_991803` for a disk archive. You do not need to enter a file name extension; the Archiver adds an identifier such as `.Z`.

After you choose OK, the destination is displayed under the Destination Container box.

7. Press Create Archive. A window titled Archiver working is displayed, flashing a green button to indicate that the archive is being written. The files being archived are displayed in the Destination Container.
8. You have the option of printing a copy of the files list to keep as a record with the tape after the archive is complete.
9. Choose Cancel to return to the Archiver main window. You can enter the name of the archive file and use the Show Contents... option to verify that the archive was written correctly. The tape or archive file is read and the contents displayed in the Show Contents Window.

To extract an archive, you must specify a destination on a target device such as a disk. If you are not recovering a damaged file system on a complete disk partition, you may consider using a temporary location rather than overwriting existing directories. Then you can restore individual files and directories as needed. Also you can restore selected files from the archive as follows:

1. Invoke the `/usr/bin/X11/dxarchiver` GUI from a terminal command line, or open the CDE Application Group: System_Admin. Then open System Admin Subgroup: Daily Admin and click on the Archiver icon.
2. Choose Show Contents… to select individual files and directories. The tape or archive file is read and the contents displayed in the Archiver Show Contents window. Select individual files or directories as follows:
   - In the Archiver Show Contents window, select a file or directory to highlight it.
   - Move to another file or directory, hold down the Ctrl key and click to select it.
   - After you select all the files that you want, choose OK in the Archiver Show Contents window. The files are displayed in the Source Container box in the Archiver main window. You can use the Edit menu to make additional changes to selections. For example, highlight an entry in the source container and choose Edit: Clear Selected Source to delete it.
3. Choose the Extract... option to display the Archiver Extract window.
4. Enter a destination directory. This directory can be the same as the archive, assuming that files can be overwritten. Alternatively, give the

path to a temporary location. This path must be to an existing directory, or you must open a terminal and create it with the `mkdir` command. Alternatively, create a folder by using the New Folder option in CDE File Manager. The destination is displayed under the Destination Container box.

5. Choose Extract Contents to begin the extraction. A window titled Archiver Working is displayed, flashing a green button to indicate that the archive is being extracted. The files being recovered are displayed in the Destination Container.
6. After the archive is complete, you can optionally print a copy of the files list, to keep as a record.
7. Choose Cancel to return to the Archiver main window. Before exiting, use the File Manager or a terminal window to ensure that the files were recovered as expected and that the file contents are not corrupted.

Note

This step is recommended before you proceed to remove any archives from tape or other media.

You can now remove the tape or other media as described in the owner's manual for the device, and store the media in a safe location (or in accordance with your site backup policy and procedures).

9.10 Creating a Standalone System Kernel on Tape

You can create a bootable standalone system (SAS) kernel on tape. The SAS kernel has a built-in memory file system (mfs), which contains the minimum commands, files, and directories needed to restore the system image. This is referred to as the miniroot file system. You can add required file systems to the tape for data or programs that you may need on the recovered system.

To create the SAS kernel, you must use the SysMan Menu Create a Bootable Tape option or the `btcreate` command line utility. After you create the kernel, you can restore the customized image by using the `btextract` utility. You can invoke only a single instance of the `btcreate` utility. The `/usr/run/bttape.pid` locking file prevents multiple instances of the utility.

The following sections provide an overview of the `bttape` interfaces, SysMan Menu task, and the `btcreate` and `btextract` command line utilities.

9.10.1 Tape Device Requirements

If you use QIC tape drives to create bootable tapes, you must use only high-density tapes of 320 or more megabytes. The QIC-24, QIC-120, and QIC-150 format tapes of fixed-512 blocks do not work. Tapes with a variable block size, such as the QIC-320 and QIC-525, work with bootable tape. Using an improperly configured QIC tape drive to create a bootable tape results in an I/O error, a write error, or permission denied error. Therefore, you must take one of the following actions:

- Configure the drive at installation time
- Rebuild the kernel if the drive was attached to the system after the installation

A QIC tape created with the `btcreate` command may fail with the following error message when booted:

```
failed to send Read to mka... Be sure that the tape is
 write protected before booting.
```

If you are creating a bootable tape with a file system that extends to multiple tapes, the `/sbin/dump` command displays a message indicating that you must change the tape. If you do not change the tape promptly, warning messages repeat periodically until you change the tape.

The behavior of the open call to a tape device has changed. No longer can you use write mode to open a write-protected tape. An attempt to open the tape fails, returning the following message:

```
EACCES (permission denied)
```

If an application is written so that it attempts to open the tape device with `O_RDWR` permission when the intention is to read the tape only, the open attempt fails. Change any applications or scripts to open the device with `O_RDONLY` permission. Use the following command to obtain the previous behavior of the open call for applications that cannot be changed:

```
# sysconfig -r cam_tape open_behaviour=0
```

9.10.2 Using the btcreate Utility

To build a bootable SAS kernel on UFS or AdvFS file systems only, you must use the `btcreate` utility. The following sections provide an overview of the information you must have to create the SAS kernel on tape.

The `btcreate` command provides both a noninteractive and interactive user interface. Both require that you have superuser (root) privileges.

9.10.2.1 Gathering Information

To prepare to use the `btcreate` command, you must have the following information:

- Name of the kernel configuration file in the `/usr/sys/conf` directory. The default is the same as the system (HOST) name in capital letters.
- Name of the disk partition (for example, `dsk2e`) where the miniroot file system is to reside. Minimum size needed on the disk is 38000 blocks (512 bytes per block). This disk partition must not be in a mounted state when `btcreate` is executed.
- Name of the tape device, for example `/dev/tape/tape0_d0`, where the SAS kernel and file systems are to reside.
- Device name, mount point, and type of each file system (UFS or AdvFS) that you want to back up to the tape device. The following examples show valid UFS and AdvFS entries:

  UFS:

  ```
  /dev/dsk1a  /     ufs
  /dev/dsk1g  /usr  ufs
  /dev/vol/rootdg/rootvol  /ufs
  ```

  AdvFS:

  ```
  root_domain#root  /     advfs
  usr_domain#usr    /usr  advfs
  ```

  Note

  Do not select swap partitions for file system backups.

  For UFS file systems that are stored on LSM volumes, the `vdump` and `vrestore` utilities are used during bootable tape creation.
- An *`addlist_file`* file, which lists the files or directories you want to include on the miniroot file system.
- An *`fslist_file`* file, which specifies the file systems to back up.
- A `/usr/lib/sabt/sbin/custom_install.sh` script, if you want to customize the restored system image. Write the script using the Bourne shell language (`sh1`) because it is the only shell provided on the miniroot file system. The `btcreate` command copies the `custom_install.sh` file onto tape and places it in the `sbin` directory on the miniroot file system. The `btextract` command invokes the `custom_install.sh` script before exiting.

The following additional features may be useful in planning your bootable tape layout:

- Use the `-d` option to specify the location where `btcreate` creates its temporary files. If you do not specify a location, 156000 blocks (512 bytes per block) of disk space in the `/usr` file system is required.
- You can label disks using your own custom `disklabel` script, which must meet the following requirements:
  - It must be located in the`/usr/lib/sabt/etc` directory.
  - It must be named `custom_disklabel_file`.

  If a custom `disklabel` script is not present, the `btextract` command labels the disks in the usual manner. See `disklabel`(8) for more information.

9.10.2.2 Creating the SAS Kernel

To create the SAS kernel, the `btcreate` command copies the `/usr/sys/conf/`*`YOUR_SYSTEM_NAME`* configuration file to `/usr/sys/conf/`*`YOUR_SYSTEM_NAME.BOOTABLE`* and modifies it as follows:

```
config    vmunix   root  on md
pseudo-device      memd  38000
```

These modifications configure a memory file system of 38000 blocks. The memory file system and the disk partition where the miniroot file system reside are equivalent in size.

After modifying the configuration file, the `btcreate` command executes the `doconfig` command and moves the bootable kernel to the `/usr/sys/bin` directory. For information on the command syntax and options, see `btcreate`(8).

9.10.3 Using the btextract Utility

The `btextract` command is a shell script that restores file systems from tapes that contain a SAS kernel created by using the `btcreate` utility. You can perform a default restoration or an advanced restoration of the system.

During a default restoration, you can choose to duplicate the customized system on more than one system of the same hardware platform type. You cannot specify which disk partitions to use for the restore operation. Instead, the `btextract` command restores file systems using the disk partition information gathered during the `btcreate` session and all existing information is overwritten. If you are performing an advanced restoration, you can optionally specify which disk partition to use. However you can

duplicate the customized system only on a system of the same hardware platform type.

To use the btextract command, place the system in a halt state, initialize the system, then boot from the tape as follows:

```
>>> init
>>> show dev
>>> boot -fl "nc" MKA500
```

In this example, the show dev command provides the device name under BOOTDEV and MKA500 is the BOOTDEV.

After the initial boot completes, the shell invokes the btextract utility. If you created a /usr/lib/sabt/sbin/custom_install.sh script during the btcreate session, the btextract command invokes the custom_install.sh script before exiting. You can create a custom_prerestore answer file to automate the recovery procedure. See btcreate(8) for more information.

After the btextract command completes its task, you must shut down and reboot the system from the restored disk as follows:

```
# shutdown -h now
>>> boot DKA100
```

In this example, DKA100 is the BOOTDEV.

See btextract(8) for more information and examples.

9.10.4 Using the SysMan Menu boot_tape Option

The following steps describe the basic process for creating a bootable tape. It assumes that you have gathered the necessary device data as described in Section 9.10.2.1, and the tape device is ready to save.

1. Invoke the Create a Bootable Tape task from the SysMan Menu, or enter the following command at the prompt:

   ```
   # sysman boot_tape
   ```

2. A window titled Bootable Tape Creation on *hostname* is displayed. Complete the fields or choose options as follows:

   - In the Kernel Name field, the default kernel name for the host is displayed. This is usually the same as the local host name. However, you can enter any name for the saved kernel.
   - The Miniroot File System field provides the following options:
     – The option to create the miniroot as a memory file system (mfs) or a Disk Partition. Select the option of your choice.

– The option to specify a disk partition name such as dsk0b with Specify Disk Partition/mfs..., which opens a dialog box in which you enter the disk partition name.

- The Tape Device field contains the name of the default tape device, usually tape0_d1. This is the name of the device on which the SAS kernel is saved, but you can specify any other supported device.
- The Customizing the Miniroot File System field displays the default file location for the addlist file. This is a data file that contains a list of additional files that you want to include, such as commands or utilities. You cannot exceed 360 KB of data in the mfs. This list is stored in the /usr/lib/sabt/etc/addlist file by default but you can choose to create your own location.

  To create a new append file, or modify an existing append file:

  a. Select Create/Modify Miniroot Append File to open the Create/Modify window.

  b. Select Add to open the Add/Modify window. Specify the location of the file that you want on the local host. For example, to add the kill command, enter /sbin/kill. Then specify the location on the miniroot file system where the file is located, such as /sbin. Select OK to return to the Create/Modify window.

  c. The Contents of file: box contains a list of the files to be appended. Select OK to return to the Bootable Tape Creation main window.

- The Selecting File Systems option enables you to back up file systems, such as /usr or an AdvFS domain such as root_domain#root. The list of files to be backed up is stored in /usr/lib/sabt/etc/fslist, but you can specify any name that you want. Add file systems as follows:

  a. Select Create/Modify File Systems Backup File... to open the Create/Modify window.

  b. Select Add to open the Add/Modify window. Specify the disk partition mounted on the local host, such as /dev/disk/dsk0g, then specify the mount point, such as /usr. Select OK to return to the Create/Modify window.

  c. The file systems to be backed up are listed in the Contents of file: box. Select OK to return to the Bootable Tape Creation main window.

3. After completing the required fields, you are ready to create the tape. Select OK in the Bootable Tape Creation main window to proceed. A message window opens to indicate that the task has started. The

creation of the tape can take twenty or more minutes, depending on the speed of the devices used.

If the task cannot be completed, a message is displayed informing you that the error log is located in `/var/adm/btcreate.log`.

4. After the tape is written successfully, a message is displayed confirming the success and the location of the log file, `/var/adm/btcreate.log`.

   Print `btextract`(8) and store it with the tape for future reference.

5. Use the instructions in Section 9.10.3 and `btextract`(8) to restore the bootable SAS kernel. Consider running a test recovery to ensure that any future recovery works as intended.

10

Administering the System Accounting Services

This chapter describes how to set up and use the system accounting services and includes the following information:

- An overview of system accounting, what can be recorded, the scripts and commands used, and the system files and logs (Section 10.1)
- How to set up system accounting (Section 10.2)
- How to start or stop accounting (Section 10.3)
- A description of how connections to the system are recorded, the log files and explains the associated commands (Section 10.4)
- A discussion of how user jobs running on the system are recorded, the log files, and explains the associated commands (Section 10.5)
- The use of disk storage is recorded and the associated commands for retrieving data (Section 10.6)
- How the use of system administration services are recorded and the associated commands for retrieving data (Section 10.7)
- How the use of printer services are recorded and the associated commands for retrieving data (Section 10.8)
- The reporting features available for all accounting operations and explains the associated commands (Section 10.9)

10.1 Accounting Overview

The accounting services are shell scripts and commands you use to manipulate an accounting database to obtain a diagnostic history of system resource use and user activity and to create report files.

You can set up accounting so that information is collected automatically on a periodic basis. Also, you can invoke accounting shell scripts and commands manually to obtain accounting information when you need it.

Using the accounting services, you can obtain accounting information for the following:

- Amount of connect time

- Amount of CPU time
- Number of processes spawned
- Number of connect sessions
- Amount of memory usage
- Number of I/O operations and number of characters transferred
- Disk space usage (in blocks)
- Amount of modem usage and telephone connect time
- Printer usage, including the number of printing operations and amount of printed matter, according to user name or printer name

If accounting is enabled, the kernel and other system processes write records to the accounting database files, which are the source of all the accounting information.

The accounting database files are located in the `/var/adm` directory and include the following files:

File	Description
`wtmp`	The login/logout history file
`utmp`	The active connect session file
`pacct`	The active process accounting file
`dtmp`	The disk usage file

The accounting scripts and commands access the records in the accounting database files and reformat them so that you can use the records for purposes such as archiving, diagnostic analysis, or resource billing.

The various accounting shell scripts and commands also can do the following:

- Format the database file records
- Create new source files from the database file records
- Display the database file records
- Merge data from several files into a single formatted file
- Summarize data in files that you can use to create reports

You can redirect or pipe script and command output to files or to other scripts and commands.

System accounting allows you to distinguish between prime time and nonprime time. The system is used most during prime time and least during nonprime time. System use during nonprime time can be assessed at a lower rate than system use during prime time. You specify the period of nonprime

time in the /usr/sbin/acct/holidays database file. Usually, if enabled, automatic accounting is performed during nonprime time.

The accounting period begins when the /var/adm/pacct file is created by the startup shell script when accounting is turned on or by the runacct script, which is usually run every day.

In command output, the order of date and time information is site dependent. You can change the order of date and time specifications by setting the NLTIME environment variable.

10.1.1 Accounting Shell Scripts and Commands

There are 14 accounting shell scripts and 20 accounting commands. The shell scripts often call the accounting commands or other shell scripts. The accounting commands and shell scripts create and write records to the accounting database files. Table 10–1 describes the accounting commands and shell scripts.

Table 10–1: Accounting Commands and Shell Scripts

Name	Type	Description
ac	Command	Displays connect session records.
acctcms	Command	Formats the binary command usage summary files.
acctcom	Command	Displays process accounting record summaries from the default pacct database file or a specified file.
acctcon1	Command	Summarizes the records in the wtmp file in ASCII format.
acctcon2	Command	Summarizes the contents of the files formatted by the acctcon1 command.
acctdisk	Command	Performs comprehensive disk usage accounting.
acctdusg	Command	Performs disk block usage accounting.
acctmerg	Command	Merges accounting record files.
accton	Command	Turns on process accounting.
acctprc1	Command	Displays records of acct type structure by user identification number and login name.
acctprc2	Command	Displays records of acct type structure by user identification number and full name.
acctwtmp	Command	Writes records to the /var/adm/wtmp file.
chargefee	Script	Writes a charge-fee record to the /fee database file.

Table 10–1: Accounting Commands and Shell Scripts (cont.)

Name	Type	Description
ckpacct	Script	Verifies the size of the /var/adm/acct/pacct active binary process accounting file to ensure that it is not too large.
diskusg	Command	Performs disk accounting according to user identification number.
dodisk	Script	Writes daily disk usage accounting records to the /var/adm/nite/dacct disk usage accounting database file.
fwtmp	Command	Displays the /var/adm/wtmp binary file records in ASCII format, allowing you to fix errors.
last	Command	Displays login information.
lastcomm	Command	Displays information about commands that were executed.
lastlogin	Script	Writes the date of the last login for all users to the /var/adm/acct/sum/loginlog file.
monacct	Script	Creates monthly summary accounting report files.
nulladm	Script	Creates files that are owned by the adm user and group and that have 664 permission.
pac	Command	Displays printer accounting records.
prctmp	Script	Displays the /var/adm/acct/nite/ctmp connect session record file.
prdaily	Script	Collects and displays daily accounting records from various files.
printpw	Command	Displays the contents of the /etc/passwd file.
prtacct	Script	Formats in ASCII and displays a tacct daily accounting file.
remove	Script	Removes any /var/adm/acct/sum/wtmp*, /var/adm/acct/sum/acct/pacct*, and /var/adm/acct/nite/lock* files.
runacct	Script	Invokes the daily accounting processes. This command periodically calls various accounting commands and shell scripts to write information to various accounting files.
sa	Command	Displays a summary of accounting records.
shutacct	Script	Turns off accounting.
startup	Script	Enables accounting processes.

Table 10–1: Accounting Commands and Shell Scripts (cont.)

Name	Type	Description
`turnacct`	Script	Controls the creation of process accounting files.
`wtmpfix`	Command	Corrects date and time stamp inconsistencies in the `/var/adm/wtmp` file.

10.1.2 Accounting Files

Many binary and ASCII files are created and maintained by the kernel or by the accounting commands and shell scripts.

You should ensure that the accounting files, particularly those in binary format, do not become too large. Some extraneous files are produced by the accounting commands and shell scripts, but in general these files are temporary and exist only while the process is running. Under some circumstances (if a process terminates prematurely, for example), one or more temporary files can appear in one of the `/var/adm` subdirectories. You should examine these subdirectories periodically and remove the unnecessary files.

Accounting files can become corrupted or lost. The files that are used to produce daily or monthly reports, such as the `/var/adm/wtmp` and `/var/adm/acct/sum/tacct` accounting database files, must have complete integrity. If these files are corrupted or lost, you can recover them from backups. In addition, you can use the `fwtmp` or the `wtmpfix` command to correct the `/var/adm/wtmp` file. See Section 10.4.2 and Section 10.4.1 for more information. You can use the `acctmerg` command to fix errors in the `/var/adm/acct/sum/tacct` file. See Section 10.9.2 for more information.

The `/var/adm/acct/nite` directory contains files that are reused daily by the `runacct` script. Some of these files have binary counterparts in the `/var/adm/acct/sum` directory, which contains the cumulative summary files that are updated by the `runacct` shell script and used by the `monacct` shell script to produce monthly reports.

Table 10–2 lists the database files in the `/var/adm` directory.

Table 10–2: Database Files in the /var/adm Directory

Name	Type	Description
`dtmp`	ASCII	Contains temporary output produced by the `dodisk` shell script.
`fee`	ASCII	Contains output from the `chargefee` shell script.

Table 10–2: Database Files in the /var/adm Directory (cont.)

Name	Type	Description
pacct	Binary	Specifies the active process accounting database file. If a process is called by a user, another process, or a script file, process information is written to this file.
pacct*n*	Binary	Specifies the alternate pacct file created by the turnacct switch command. The pacct database file becomes large quickly if a system has many users. A single pacct file is limited to 500 1024-block disk spaces. The size of these files is monitored by the runacct shell script. Each time a new pacct*n* file is created, the value *n* is incremented by one.
qacct	Binary	Contains queuing (printer) system accounting records. This file is used by the runacct shell script.
savacct	Binary	Specifies the file used by the sa command to store system process accounting summary records.
Spacct*n.mmdd*	Binary	Specifies the pacct*n* files produced by the runacct shell script for the month and day specified by *mm* and *dd*, respectively.
usracct	Binary	Specifies the file used by the sa command to store user process accounting summary records.
utmp	Binary	Specifies the active connect session accounting database file, which is written to if a user calls a process that produces a connect session.
wtmp	Binary	Specifies the cumulative login/logout accounting database file. If a user logs in to the system, connect time and user information is written to this file.

Table 10–3 lists the database files in the /var/adm/acct/nite directory.

Table 10–3: Daily Files in the /var/adm/acct/nite Directory

Name	Type	Description
active	ASCII	Specifies the daily runacct shell script progress file. When the runacct shell script executes, information about its progress is written to this file. This file also contains error and warning messages.
active*mmdd*	ASCII	Specifies the daily runacct shell script error file for the month and day specified by *mm* and *dd*, respectively. This file is similar to the active file.

Table 10–3: Daily Files in the /var/adm/acct/nite Directory (cont.)

Name	Type	Description
`cklock`	ASCII	Specifies the file the `ckpacct` shell script uses to ensure that more than one `runacct` shell script is not called during any 24-hour period. This file is removed each day if the `runacct` shell script has completed.
`cms`	ASCII	Specifies the active total daily command summary file. This file is the ASCII version of the `/var/adm/acct/sum/cms` file. This file is created by the `acctcms` command, which is called by the `runacct` shell script to rewrite the `/var/adm/acct/sum/cms` file records. The `monacct` shell script initializes this file.
`ctacct.`*`mmdd`*	Binary	Specifies the connect accounting records in `tacct.h` format that are obtained from the connect session accounting records for the month and day specified by *mm* and *dd*, respectively. This file is temporary and is deleted after the `daytacct` file records are written for each accounting period.
`ctmp`	ASCII	Specifies the temporary login/logout record file. This file contains the output of the `acctcon1` accounting command, which is called by the `runacct` shell script to rewrite the `wtmp` file records.
`daycms`	ASCII	Specifies the daily command summary file. This file is the ASCII version of the `/var/adm/acct/sum/daycms` binary file. The `runacct` shell script calls the `prdaily` shell script, which invokes the `acctcms` command to create the file.
`daytacct`	Binary	Contains the total accounting records in `tacct.h` format for the previous day.
`dacct`	Binary	Contains the weekly total disk usage accounting records when the `acctdisk` command is called by the `dodisk` shell script.
`lastdate`	ASCII	Specifies the last day that the `runacct` shell script was executed.
`lineuse`	ASCII	Contains terminal (tty) line connect times. This file provides line use statistics for each terminal line used during the previous accounting period.

Table 10–3: Daily Files in the /var/adm/acct/nite Directory (cont.)

Name	Type	Description
`lock`	ASCII	Specifies the file used to ensure that the `cron` daemon does not call the `runacct` shell script more than once during any 24-hour period. This file is removed each day when the `runacct` shell script has completed.
`log`	ASCII	Contains diagnostic output that is produced when the `runacct` script invokes the `acctcon1` command.
`owtmp`	Binary	Specifies the daily `wtmp` file after a correction by the `wtmpfix` command.
`ptacct`*n.mmdd*	Binary	Specifies the additional daily `pacct`*n* files for the month and day specified by *mm* and *dd*, respectively. These files are created if the daily `pacct` process accounting file requires more than 500 disk blocks.
`reboots`	ASCII	Contains a list of system reboots during the previous accounting period.
`statefile`	Binary	Specifies the final `runacct` shell script execution state.
`wtmp.`*mmdd*	Binary	Specifies the fixed daily login/logout accounting database file for the month and day specified by *mm* and *dd*, respectively. Connect session records of users who logged in to the system during the previous day are written to this file.
`wtmperror`	ASCII	Contains any error messages produced when a `wtmp` file is fixed during the execution of the `wtmpfix` command.
`wtmperror`*mmdd*	ASCII	Contains any error messages produced when the `runacct` shell script detects an error during execution of the `wtmpfix` command for the month and day specified by *mm* and *dd*, respectively.

Table 10–4 lists the database files in the `/var/adm/acct/sum` directory.

Table 10–4: Summary Files in the /var/adm/acct/sum Directory

Name	Type	Description
`cms`	Binary	Specifies the active total command summary file. When the `runacct` shell script is executed, records are written to this file to obtain the total command summary file.
`cmsprev`	Binary	Specifies the previous day's `/var/adm/acct/sum/cms` file.
`daycms`	Binary	Specifies the previous day's command summary file. When the `runacct` shell script is executed, monthly command summary records for the previous day are written to this file.
`loginlog`	ASCII	Contains a list of the last monthly login date for each user name.
`rprt`*mmdd*	ASCII	Specifies the daily accounting report for the month and day specified by *mm* and *dd*, respectively.
`tacct`	Binary	Specifies the cumulative total accounting file. This file is the total daily accounting file for system use. It is updated on a daily basis by the `runacct` shell script.
`tacct`*mmdd*	Binary	Specifies the total accounting file for the month and day specified by *mm* and *dd*, respectively.
`tacctprev`	Binary	Specifies the previous day's `tacct` file. This file is the `tacct` binary file for the previous accounting period.

Table 10–5 lists the database files in the `/var/adm/acct/fiscal` directory.

Table 10–5: Monthly Files in the /var/adm/acct/fiscal Directory

Name	Type	Description
`cms`*mm*	Binary	Specifies the active command summary file for the month specified by *mm*.
`fiscrpt`*mm*	ASCII	Specifies the accounting report for the month specified by *mm*.
`tacct`*mm*	Binary	Specifies the cumulative total accounting file. This file is the total accounting file for system use. It is updated on a monthly basis by the `monacct` shell script.

10.2 Setting Up Accounting

You must install the System Accounting Utilities subset to use accounting. Use the following command to see if this subset is installed:

```
# setld -i | grep count
OSFACCTxxx   installed  System Accounting Utilities \
(System Administration)
```

If the subset is not installed, use the `setld` command to install it from the distribution media or from a RIS server. When the subset is installed, you can proceed to enable the required accounting services.

In a system environment where many users compete for system resources, UNIX system accounting allows you to track system use. You must decide the quantity and type of information that you want to track. You also must decide if you want to enable automatic accounting. To enable automatic accounting, you specify accounting commands and shell scripts in the files in the `/usr/spool/cron/crontabs` directory.

To obtain accounting information for all the machines in a network, you should set up accounting on a single machine. Use the following procedure to enable system accounting. The following sections describe these steps in detail.

1. Enabling accounting in the `/etc/rc.config` file. (Section 10.2.1)
2. Verifying the `qacct`, `pacct` and `fee` files. (Section 10.2.2)
3. Editing the `/usr/sbin/acct/holidays` file to specify prime time, nonprime time, and holidays. (Section 10.2.3)
4. Modifying the files in the `/usr/spool/cron/crontabs` directory to invoke accounting shell scripts and commands to enable automatic accounting. (Section 10.2.4)

Resource accounting is discussed separately from printer accounting because the print driver software uses different servers, daemons, and routines. Setting up printer accounting is described in Chapter 8.

10.2.1 Enabling Accounting in the rc.config File

To enable accounting, you must add the following line to the `/etc/rc.config` file:

```
ACCOUNTING="YES"
```

You can use the `rcmgr` command to set the variable, as follows:

```
# rcmgr  set  ACCOUNTING  YES
```

You can start accounting without rebooting your system by using the `startup` command. See Section 10.3 for more information.

10.2.2 Verifying the qacct, pacct, and fee Files

The qacct queuing accounting file and the pacct process accounting database file must exist on the system for accounting to function. These files are preinstalled as blank files with path names that are context-dependent symbolic links (CDSLs). When you use the ls -l directory display command, the links resolve to the following paths:

```
/usr/var/cluster/members/member0/adm/acct/fee
/usr/var/cluster/members/member0/adm/acct/pacct
/usr/var/cluster/members/member0/adm/acct/qacct
```

If the original files do not exist (or have been destroyed accidentally) you must recreate them as CDSLs. See Chapter 6, cdslinvchk(8), and mkcdsl(8) for information on recreating CDSLs. An alternative action is to reinstall the accounting software subsets, after first saving any existing accounting data and configuration files that you want to keep.

The files must be owned by the adm user and group and have permissions of 644. Use the chown and chgrp commands to reset these values as needed.

10.2.3 Editing the holidays File

The /usr/sbin/acct/holidays file uses 24-hour time to specify prime time and nonprime time. The file also specifies holidays, which are included in nonprime time. Only the days Monday through Friday are included in prime time. You can assess system use during nonprime time at a lower rate than during prime time. If you enable automatic accounting, you should specify that the commands be executed during nonprime time.

If the /usr/sbin/acct/holidays file does not exist, you must create it. If the file exists, you must edit it to reflect your accounting needs.

You can set the NHOLIDAYS environment variable to specify the maximum number of holidays that you can include in the holidays file.

10.2.4 Modifying the crontab Files

To enable automatic accounting, you must use the crontab command to modify the files in the /usr/spool/cron/crontabs directory. The files in the /usr/spool/cron/crontabs directory contain commands that the cron daemon runs at specified times under a specific authority. For example, the commands in the /usr/spool/cron/crontabs/root file are run under root authority, and the commands in the /usr/spool/cron/crontabs/adm file are run under adm authority.

You can include the following commands and shell scripts in the /usr/spool/cron/crontabs/adm file:

`ckpacct`	This shell script examines the size of the `pacct` process accounting database file and ensures that it does not become too large.
`runacct`	This shell script includes other accounting shell scripts and commands and creates daily and monthly accounting files. You can modify the `runacct` shell script to remove the commands for the accounting features that you do not want.
`monacct`	This shell script creates monthly summary accounting files. You can modify the `monacct` shell script to remove the commands for the accounting features that you do not want.
`ac`	This command displays connect-time records. You can direct the output to a file. Also, you can add this command to the `runacct` shell script.
`pac`	This command displays printer accounting records. You can direct the output to a file. See Section 10.8 for information on enabling printer accounting.

You can include the `dodisk` shell script in the `/usr/spool/cron/crontabs/root` file. The `dodisk` shell script creates disk usage accounting records and should be run once during nonprime time each week.

See Chapter 3 and `crontab`(1) for more information on submitting commands with the `crontab` command.

The following example shows part of a `/usr/spool/cron/crontabs/adm` file that includes accounting commands and shell scripts:

```
0 2 * * 1-6 /usr/sbin/acct/runacct  > /usr/adm/acct/nite/fd2log&
5 * * * * /usr/sbin/acct/ckpacct&
0 4 1 * *  /usr/sbin/acct/monacct&
10 3 * * * /usr/sbin/ac -p > /var/adm/timelog&
40 2 * * * /usr/sbin/pac -s&
```

The following example shows part of a `/usr/spool/cron/crontabs/root` file that includes the `dodisk` shell script:

```
0 3 * * 4 /usr/sbin/acct/dodisk  > /var/adm/diskdiag&
```

10.3 Starting Up and Stopping Accounting

The `startup` and `shutacct` shell scripts enable and disable the various accounting processes. The scripts invoke the `acctwtmp` program, which adds a record to the `/var/adm/wtmp` file by using the system name as the login name.

The `startup` shell script initializes the accounting functions and has the following syntax:

/usr/sbin/acct/startup

Note

You must ensure that the `pacct` file, which is created by the `startup` script, is owned by group `adm` and user `adm` and has 664 protection. If it does not have the correct ownership, the `accton` command does not work, and the following message is displayed:

```
accton: uid/gid not adm
```

The `shutacct` script turns process accounting off and ensures that the accounting functions are halted before the system shuts down. The `shutacct` shell script has the following syntax:

/usr/sbin/acct/shutacct [*Reason*]

The *Reason* string is a user-defined reason for invoking the command. If the `shutacct` shell script is invoked, the *Reason* message is written to the `ut_line` field in the `/var/adm/wtmp` file shutdown record. Then, the `turnacct off` shell script is invoked to tell the kernel that its active accounting functions should be disabled.

10.4 Connect Session Accounting

When a user logs in or logs out, the `login` and `init` commands write the user login and logout history to records in the `/var/adm/wtmp` binary database file. The `/var/adm/utmp` binary database file is the active connect session file. All hangups, terminations of the `login` command, and terminations of the login shell cause the system to write logout records, so the number of logouts is often more than the number of sessions.

Note

If you have accounting records that date back to versions of Version 4.0A of the operating system, see `wtmpconvert`(8) for information on converting the files.

Connect session commands can convert the /var/adm/wtmp file records to useful connect session accounting records. You can obtain connect session accounting only if the /var/adm/wtmp file exists.

The formatted records in the /var/adm/wtmp file provide the following information about each connect session:

- User login name (from the /etc/passwd file)
- Line identification number (from the /etc/inittab file)
- The device name (for example, console or tty23)
- Type of entry
- Process identification number
- Process termination status
- Process exit status
- Time entry was made
- Host machine name

You can use the following two shell scripts and seven commands to obtain or modify information about system connect sessions:

Command	Description
ac	This command displays connect session records for the entire system and for each user.
acctcon1	This command summarizes connect session records and displays those records in ASCII format, using one line for each connect session.
acctcon2	This command uses the output of the acctcon1 command to produce an accounting record file of the total connect session in ASCII format.
acctwtmp	This command enables you to write records to the wtmp file by entering them from the keyboard.
fwtmp	This command displays records from files with the utmp.h file structure.
last	This command displays login information.
lastlogin	This shell script updates the /var/adm/acct/sum/loginlog file to show the last date that each user logged in.

Command	Description
prctmp	This shell script displays the contents of the session-record file (usually /var/adm/acct/nite/ctmp) that the acctcon1 command created.
wtmpfix	This command corrects the wtmp connect session records that are affected by a date modification and validates login names written to the login name field in the wtmp file.

The /usr/include/utmp.h header file structure is the record format for the following connect session files:

- /var/adm/wtmp
- /var/adm/utmp
- /var/adm/acct/nite/wtmp.*mmdd*
- /var/adm/acct/nite/ctmp

The /usr/include/utmp.h header file structure includes nine fields. Table 10–6 shows the utmp ASCII conversion format for the field number, member name in the header file structure, its description and, if necessary, character length.

Table 10–6: The utmp ASCII Conversion Structure Members

Field	Member	Description
1	ut_user	The user login name, which must have exactly sizeof(ut_user) characters.
2	ut_id	The inittab ID, which must have exactly sizeof(ut_id) characters.
3	ut_line	A memory location, where information used to describe the type of record (for example, the device name) is stored. It must have exactly sizeof(ut_line) characters.
4	ut_pid	The process identification number.
5	ut_type	The type of entry, which can specify several symbolic constant values. The symbolic constants are defined in the /usr/include/utmp.h header file.
6	ut_exit.e_termination	The process termination status.
7	ut_exit.e_exit	The process exit status.
8	ut_time	The starting time (in seconds).
9	ut_host	The host name, which must have exactly sizeof(ut_host) characters.

10.4.1 The wtmpfix Command

The /usr/sbin/acct/wtmpfix command corrects date and time stamp inconsistencies in files with the utmp.h header file structure and displays the records. The runacct script invokes the wtmpfix command.

Each time a date is entered in the /var/adm/wtmp file (for example, at system startup or by using the date command), a pair of date-change records is also written to the wtmp file. The first date-change record is the old date, which is specified in the ut_line and ut_type fields. The second date-change record is the new date, which is also specified in the ut_line and ut_type fields. The wtmpfix command uses these records to synchronize all date and time stamps in the /var/adm/wtmp file, and then the date-change record pair is removed. The date-change records never appear in an output file.

The wtmpfix command also verifies the validity of the user name field (the ut_user field) to ensure that the name consists only of alphanumeric characters, a dollar sign ($), or spaces. If an invalid name is detected, the wtmpfix command changes the login name to INVALID and displays a diagnostic message.

The wtmpfix command has the following syntax:

/usr/sbin/acct/wtmpfix [*filename*]...

The *filename* variable specifies the name of the input file. The default input file is the /var/adm/wtmp binary file.

10.4.2 The fwtmp Command

The fwtmp command allows you to correct wtmp files. The command converts binary records from files with the utmp.h header file structure to formatted ASCII records. You can edit the ASCII version of a wtmp file to repair bad records or for general file maintenance. Table 10–6 shows the ASCII structure you should use.

During system operation, date changes and reboots occur, and the records are written to the /var/adm/wtmp file. The wtmpfix command adjusts the time stamps in the /var/adm/wtmp file; however, some corrections can evade detection by the wtmpfix command and cause the acctcon command to fail. In this case, you can correct the /var/adm/wtmp file by using the fwtmp command.

The fwtmp command has the following syntax:

/usr/sbin/acct/fwtmp [-ic]

The fwtmp file uses standard input, or you can direct a file to the command.

If no options are specified with the fwtmp command, binary records are converted to ASCII records. See fwtmp(8) for information on command options.

If you want to enter /usr/include/utmp.h header file records manually, you must enter data in each of the nine fields in the order used by the utmp ASCII structure members, as shown in Table 10–6. All record-field entries that you enter from the keyboard must be separated by a space. Also, you must specify all the string fields by using blank characters, if necessary, up to the maximum string size. All decimal values must be specified with the required number of decimal places, using preceding zeros (0) to indicate the empty digit positions.

The following example converts the /var/adm/wtmp binary file records to ASCII records:

```
# /usr/sbin/acct/fwtmp  <  /var/adm/wtmp
           system boot  0 20000 0000 652547412 Jan 5 11:10:12 1994
           system boot  0 10062 0123 652547412 Jan 5 11:10:12 1994
bcheck  bl              6 80000 0000 652547413 Jan 5 11:10:13 1994
cat     cr             16 80000 0000 652547414 Jan 5 11:10:14 1994
rc      rc             17 80000 0000 652547485 Jan 5 11:11:25 1994
hoffman co console    147 70000 0001 652547495 Jan 5 11:11:35 1994
hoffman p4 pty/ttyp4 2156 80000 0002 652650095 Jan 6 15:41:35 1994
LOGIN   p4 pty/ttyp4 2140 60000 0000 652649075 Jan 6 15:24:35 1994
LOGIN   p4 pty/ttyp4 2140 80000 0000 652649086 Jan 6 15:24:46 1994
```

To correct a /var/adm/wtmp file:

1. Change your working directory to /var/adm/acct/nite.
2. Use the fwtmp command to create an ASCII version of the wtmp file.

   ```
   # fwtmp < wtmp.0617 > wtmp_temp
   ```

3. Edit the temporary file and remove the corrupted records.
4. Use the fwtmp command to recreate the wtmp file.

   ```
   # fwtmp –ic < wtmp_temp > wtmp.0617
   ```

10.4.3 The acctwtmp Command

The acctwtmp command allows you to add a string specifying the reason for invoking the command, and the current time and date to a utmp.h structured file, usually the /var/adm/wtmp file. The runacct, startup, and shutacct shell scripts invoke the acctwtmp command to record when the runacct script is invoked and when system accounting is turned on and off.

The acctwtmp command has the following syntax:

/usr/sbin/acct/acctwtmp *reason*

The *reason* variable must have a maximum of `sizeof(ut_line)` characters and be enclosed in quotation marks (" ").

10.4.4 The ac Command

The `ac` command displays connect session records from files with the `utmp` file structure shown in Table 10–6. You can use the command to perform system diagnostics and determine user charges. The `ac` command displays the total connect time for all users or the total connect time for the specified users. The connect time is given in hours rounded to the nearest hundredth. To automatically generate total user connect session files, you can include the `ac` command in the `/usr/spool/cron/crontab/adm` file or modify the `runacct` shell script and include the `ac` command. See Section 10.2.4 for information on setting up automatic accounting.

The `ac` command has the following syntax:

/usr/sbin/ac [–d] [–p] [–w *filename*] [username...]

See `ac`(8) for information on command options.

The default behavior displays the sum of the system connect time for all users. For example:

```
# /usr/sbin/ac
 "total 48804.26"
```

The following command displays the total connect time according to user name:

```
# /usr/sbin/ac  –p
buckler     61.44
fujimori   530.94
newsnug    122.38
dara         0.10
root       185.98
buchman    339.33
russell     53.96
hoff       200.43
hermi      157.81
total     1968.02
```

The total connect time for all users listed is shown in the last line.

10.4.5 The acctcon1 Command

The `acctcon1` command converts binary session records from a file with the `utmp.h` header file structure to ASCII format. A single record is produced for each connect session. The `runacct` shell script uses the `acctcon1`

command to create the `lineuse` and `reboots` files, which are included in the `/var/adm/acct/sum/rprt`*mmdd* daily report.

The `acctcon1` command has the following syntax:

/usr/sbin/acct/acctcon1 [-l *file*] [–o *file*] [–pt]

You must direct a file as input to the command. See `acctcon1`(8) for information on command options.

The following command line provides an example of a `/var/adm/acct/nite/lineuse` file. It writes records to the specified file in ASCII line-usage format, which helps you to track line usage and to identify bad lines; and it includes the reference designation of the ports that the user logged in to and the date and time stamp of the currently active connect session.

```
# acctcon1 –l line_file < /var/adm/wtmp | more line_file
TOTAL DURATION IS 57 MINUTES
LINE         MINUTES      PERCENT    # SESS   # ON   # OFF
pty/ttyp4         37           64         3      3       7
console           26           45         2      2       4
pty/ttyp5          7           11         1      1       3
pty/ttyp6          0            0         0      0       2
TOTALS            69            -         6      6      16
```

In the previous example, the ASCII line-usage format specifies the following:

- Total number of minutes that the system was in multiuser state
- The line name
- The number of session minutes used during the accounting period
- The ratio of minutes in use to the total duration
- The number of times the port was accessed (fourth and fifth columns)
- The number of logouts and any other interrupts on the line

You can compare the last column to the fourth column to determine if a line is bad.

The following example produces a sample `/var/adm/acct/reboots` file. It writes records to a file in ASCII overall-record format, which specifies a starting time, an ending time, the number of restarts, and the number of date changes.

```
# acctcon1 –o overall_file < /var/adm/wtmp | more overall_file
from Thu Jun  13 17:20:12 2002 EDT
to   Fri Jun  14 09:56:42 2002 EDT
2   date changes
2   acctg off
0   run-level S
2   system boot
```

```
2   acctg on
1   acctcon1
```

The overall-record format includes the `from` and `to` fields, which specify the time that the last accounting report was generated and the time of the current report. These fields are followed by a list of records from the `/var/adm/wtmp` file.

10.4.6 The acctcon2 Command

The `runacct` shell script invokes the `acctcon2` command to convert the `/var/adm/acct/nite/ctmp` connect session file, which is produced by the `acctcon1` command, from ASCII format into binary format.

10.4.7 The prctmp Shell Script

The `prctmp` shell script writes column headings on a connect session database file that has the `utmp.h` header file structure, such as the `/var/adm/acct/nite/ctmp` file, which is created by the `acctcon1` command. The `prctmp` shell script has the following syntax:

/usr/sbin/acct/prctmp [*filename*]

See `prctmp`(8) for more information.

10.4.8 The lastlogin Shell Script

The `lastlogin` shell script writes the last date that a user logged in to the system to the `/var/adm/acct/sum/loginlog` file. The script invokes the `printpw` command to access the login names and user identification numbers in the `/etc/passwd` file.

The `runacct` shell script invokes the `lastlogin` shell script during its CMS state. You can invoke the `lastlogin` shell script manually to update the `/var/adm/acct/sum/loginlog` file, which is included in the `/var/adm/acct/sum/rprt`*`mmdd`* daily report.

The `lastlogin` shell script has the following syntax:

/usr/sbin/acct/lastlogin

10.4.9 The last Command

The `last` command displays, in reverse chronological order, all login records in the `/var/adm/wtmp` file. For each login session, the following information is provided:

- Time that the session began
- Duration of the session

- Terminal on which the session took place

The following information is included when applicable:

- Terminations when rebooting
- Continuing sessions

The `last` command has the following syntax:

/usr/bin/last [-#] [*username*...] [*tty*...]

By default, all records are displayed. You can specify a user name and a terminal for which you want to display records.

The following example displays information only about the three previous `root` logins:

```
# last -3 root
root    ttyp1     shout     Fri Jun 21 10:56   still logged in
root    ttyp1     raven     Fri Jun 21 08:59 - 09:00  (00:00)
root    ttyp0     raven     Thu Jun 20 15:29 - 15:54  (00:24)
```

10.5 Process Accounting

Process accounting occurs when a command, shell script, or program is executed in the system. When a process exits, the kernel writes the process accounting record to the `pacct` database file. Process accounting records enable you to monitor program execution statistics. You can use the `ps` command to get information about running processes. The `accton` command creates the `/var/adm/pacct` file and turns on process accounting.

The `pacct` file grows in size. The `ckpacct` command verifies the size of the `pacct` file and creates a `pacct`*n* file if the `pacct` file is larger than a specified size.

The `pacct` database file includes the following process information:

- Process type (for example, child process)
- Exit status indicating how the process terminated
- User identification number
- Group identification number
- Terminal from which the process originated
- Start, user, system, and CPU time
- Amount of memory used
- Number of I/O characters transferred
- Number of 1024-byte blocks read or written

- Name of the command used to start the process

The record format for the process accounting files is `tacct` format and is established by the `acct` header file structure. The `acct` header file structure is defined in the `/usr/include/sys/acct.h` header file and includes up to 18 columns of accounting information. The `tacct` structure members are defined in the private `tacct.h` header file.

Table 10–7 specifies the column number, heading, and description for files with the `tacct` format.

Table 10–7: The tacct File Format

Column	Heading	Description
1	UID	Specifies the user identification number, which is obtained from the `/etc/passwd` file.
2	LOGNAME	Specifies the user login name, which is obtained from the `/etc/passwd` file.
3	PRI_CPU	Specifies the prime time CPU run time, which is the total time (in seconds) that prime time CPU run time was charged to the user.
4	NPRI_CPU	Specifies the nonprime time CPU run time, which is the total time (in seconds) that nonprime time CPU run time was charged to the user.
5	PRI_MEM	Specifies the prime time memory kcore minutes, which is the total CPU time (in minutes) multiplied by the mean size of the memory used.
6	NPRI_MEM	Specifies the nonprime time memory kcore minutes, which is the total CPU time (in minutes) multiplied by the mean size of the memory used.
7	PRI_RD/WR	Specifies the total number of characters transferred during prime time operation.
8	NPRI_RD/WR	Specifies the total number of characters transferred during nonprime time operation.
9	PRI_BLKIO	Specifies the total number of I/O blocks transferred during prime time read and write operations. The number of bytes in an I/O block depends on how it was implemented.
10	NPRI_BLKIO	Specifies the total number of I/O blocks transferred during nonprime time read and write operations. The number of bytes in an I/O block depends on how it was implemented.
11	PRI_CONNECT	Specifies the total number of prime time seconds that a connection existed.

Table 10–7: The tacct File Format (cont.)

Column	Heading	Description
12	NPRI_CONNECT	Specifies the total number of nonprime time seconds that a connection existed.
13	DSK_BLOCKS	Specifies the total number of disk blocks used.
14	PRINT	Specifies the total number of pages queued to any printer in the system.
15	FEES	Specifies the number of units charged. This value is specified with the /usr/sbin/acct/charge-fee shell script.
16	PROCESSES	Specifies the total number of processes spawned by the user during the accounting period.
17	SESS	Specifies the total number of times the user logged in during the accounting period.
18	DSAMPS	Specifies the total number of times that the disk accounting command was used to get the total number of disk blocks specified in the DSK_BLOCKS column. You can divide the value in the DSK_BLOCKS column by the value in the DSAMPS column to obtain the average number of disk blocks used during the accounting period.

Process accounting shell scripts and commands allow you to combine information about commands and the resources used to process the commands. The following sections describe the process accounting shell scripts and commands.

10.5.1 The accton Command

The accton command enables and disables process accounting. The accton command has the following syntax:

/usr/sbin/acct/accton [*filename*]

If you do not specify the *filename* variable, process accounting is disabled. If you specify the *filename* variable, process accounting is turned on and the kernel writes process accounting records to the specified file. Usually, this file is the pacct file; however, you can specify a different process accounting database file. The file must exist in the /var/adm directory, be owned by user adm, and be a member of the adm login group.

Note

The `runacct` and `turnacct` shell scripts use the `pacct` process accounting database file. If you specify a process accounting database file other than the `pacct` file, the `runacct` and `turnacct` shell scripts are affected.

10.5.2 The turnacct Shell Script

The `turnacct` shell script controls the process accounting functions and creates process accounting files. You must be superuser to use the shell script. The `turnacct` script has the following syntax:

turnacct [on | off | switch]

The `turnacct on` shell script turns on process accounting by invoking the `accton` shell script with the `pacct` file argument.

The `turnacct off` shell script turns off process accounting by invoking the `accton` command without an argument to disable process accounting.

The `turnacct switch` shell script moves the contents of the `pacct` file to the `pacct`*n* file and then creates a new `pacct` file.

10.5.3 The ckpacct Shell Script

The `pacct` file can grow in size. If the `pacct` file is larger than a specified limit and if enough disk space is available, the `ckpacct` script invokes the `turnacct switch` shell script to move the contents of the `pacct` file to the `pacct`*n* file and create a new `pacct` file.

You can set up your `cron` daemon to invoke the `ckpacct` script periodically. See Section 10.2.4 for more information.

The `ckpacct` shell script has the following syntax:

ckpacct [*blocksize*]

The *blocksize* variable specifies the size limit (in disk blocks) for the `pacct` file. The default size is 500 disk blocks.

If you invoke the `ckpacct` shell script, the script reports the number of disk blocks that are available in the `/var/adm` directory. If the number of available blocks is less than the size limit, process accounting is disabled by invoking the `turnacct off` shell script. A diagnostic message is displayed and mailed to the address that is specified with the `MAILCOM` environment variable. Use the `putenv` function to set the `MAILCOM` environment variable to the following command:

```
mail root adm
```

The following diagnostic message shows that there are 224 disk blocks remaining in the /var/adm directory:

```
ckpacct: /var/adm too low on space  (224 blocks)
  "turning acctg off"
```

The ckpacct shell script continues to display diagnostic messages until adequate space exists in the /var/adm directory.

10.5.4 The acctcom Command

The acctcom command displays summaries of process accounting records. Command options allow you to specify the type and format of the output. You do not have to be superuser to use the acctcom command.

The acctcom command displays information only about processes that have terminated; use the ps command to display information about active processes. The acctcom command has the following syntax:

/usr/bin/acctcom [*option*...] [*filename*...]

If you do not specify the *filename* variable, the command uses the pacct file to obtain the process accounting records. You can use the *filename* variable to specify a different process accounting file that has the acct.h header file structure. If you specify more than one *filename* variable, the acctcom command reads the files in chronological order.

If you do not specify any command options, the default output includes the following information in a column heading format:

- Time and date that accounting was enabled
- Command name
- User name
- Terminal name
- Process start time
- Process end time
- Real seconds
- CPU seconds
- Mean memory size (in kilobytes)

See acctcom(8) for information on the command options.

The following is an example of the default process accounting summary output:

```
# /usr/bin/acctcom  /var/adm/pacct1
ACCOUNTING RECORDS FROM: Mon Jun 17 02:00:00 2002
COMMAND                  START    END        REAL   CPU    MEAN
NAME     USER   TTYNAME  TIME     TIME      (SECS) (SECS) SIZE(K)
#sa      root   ttyp1    11:59:00 11:59:00   0.77   0.01   0.00
ls       root   ttyp1    11:59:04 11:59:04   0.11   0.01   0.00
uugetty  root   ?        11:58:39 11:59:48  69.53   0.01   0.00
#ls      root   ttyp1    11:59:55 11:59:55   0.30   0.01   0.00
uugetty  root   ?        11:59:49 12:00:58  69.48   0.01   0.00
cp       adm    ?        12:05:01 12:05:01   0.33   0.01   0.00
chmod    adm    ?        12:05:01 12:05:01   0.27   0.01   0.00
#df      adm    ?        12:05:02 12:05:02   0.38   0.01   0.00
awk      adm    ?        12:05:02 12:05:02   0.58   0.01   0.00
sed      adm    ?        12:05:02 12:05:02   0.56   0.01   0.00
```

10.5.5 The sa Command

The sa command summarizes process accounting information. This command helps you to manage the large volume of accounting information. The files produced by the sa command include all the available process accounting information. The sa command has the following syntax:

/usr/sbin/sa [*options*...] [*filename*]

The *filename* variable specifies a process accounting file with the acct.h header file structure. If the *filename* variable is not specified, the pacct file is used.

If you invoke the sa command with no options, the default output consists of six unheaded columns. Certain command options allow you to expand the six columns to include more information. You can specify options to change the format and to output additional information that includes an identifying suffix. See sa(8) for information on the command options.

The following example shows the default format of the output of the sa command:

```
# /usr/sbin/sa
798    277.24re     0.08cpu  3248790avio    0k
  7     33.42re     0.08cpu   103424avio    0k    csh
 14      0.08re     0.00cpu   127703avio    0k    mv
 40      0.34re     0.00cpu   159968avio    0k    cp
  2      0.01re     0.00cpu   132448avio    0k    acctwtmp
 34      0.13re     0.00cpu   133517avio    0k    chmod
 23      0.10re     0.00cpu   139136avio    0k    chgrp
 25      0.11re     0.00cpu   144768avio    0k    chown
 36      0.15re     0.00cpu   133945avio    0k    dspmsg
 32      0.18re     0.00cpu   134206avio    0k    cat
[1]    [2]        [3]        [4]          [5]  [6]
```

[1] Shows information about the number of command executions. An additional column is added to show the command percentage if you specify the `-c` option.

[2] Shows information about the amount of real time used. An additional column is added to show the real-time percentage if you specify the `-c` option.

[3] Shows information about CPU time used. Depending on the options specified, the column can show the total system and user CPU time, the user CPU time, the system CPU time, or the ratio of user CPU time to system CPU time. An additional column is added to show the real-time percentage if you specify the `-c` option. Also, an additional column is added to show the ratio of real time to total user and system CPU time if you specify the `-t` option.

[4] Shows information about disk I/O operations, either the average number of I/O operations or the total number of I/O operations.

[5] Shows information about kiloblocks (number of blocks multiplied by 1024) used or the memory time integral.

[6] Shows the command name.

The following example adds three columns to the default format to display the following percentages:

```
# /usr/sbin/sa -c
645 100.00% 324.10re 100.00% 0.02cpu 100.00% 6171050avio 0k
  2   0.31%  25.70re   7.93% 0.02cpu 100.00%  107392avio 0k csh
  6   0.93%   0.04re   0.01% 0.00cpu   0.00%  132928avio 0k mv
 38   5.89%   0.33re   0.10% 0.00cpu   0.00%  163357avio 0k cp
  2   0.31%   0.01re   0.00% 0.00cpu   0.00%  132992avio 0k cat
 26   4.03%   0.11re   0.03% 0.00cpu   0.00%  136832avio 0k chmod
 24   3.72%   0.10re   0.03% 0.00cpu   0.00%  139824avio 0k chgrp
     [1]         [2]           [3]
```

The additional columns show the following information:

1. Indicates the number of times each command was executed with respect to the total number of times all commands were executed.
2. Indicates the amount of real time needed to execute the command the number of times specified in column one with respect to the total real time required to execute all the commands.
3. Indicates the amount of CPU time needed to execute the command the number of times specified in column 1 with respect to the total CPU time required to execute all commands.

10.5.6 The acctcms Command

The `acctcms` command produces ASCII and binary total command summary files from process accounting records. You specify process accounting files that have the `/usr/include/sys/acct.h` header file structure, such as the `pacct` file. The `acctcms` command sorts the records and combines the statistics for each command used during the accounting period into a single record. The records allow you to identify the commands used most and the commands that use the most system time.

The `runacct` shell script invokes the `acctcms` command during its CMS state. You can invoke this command manually to create a command summary report.

The `acctcms` command has the following syntax:

/usr/sbin/acct/acctcms [-acjnopst] *filename*...

If you invoke the `acctcms` command with no options, the command sorts the output in descending order according to total kcore minutes, which is the number of kilobytes of memory used by the process multiplied by the buffer time used. Binary output is the default. Use the following calculation to obtain the kcore minutes:

```
kcoremin=[(CPU time in seconds)*(mean memory size in kbyte)]/60
```

See `acctcms`(8) for information on the command options.

Note

If you use the `acctcms` command to produce a total summary file in ASCII format, each command record consists of more than 80 characters, and the entire width of 8.5 x 11-inch paper could be used if the 10-character per inch constant-width font is specified. If part of a record exceeds the column width, it is moved to the next line.

The following example produces ASCII output that includes the statistics for commands that were invoked only once in a row specifying `***other` in the COMMAND NAME column:

```
# acctcms -a -j /var/adm/pacct1
                          TOTAL COMMAND SUMMARY
COMMAND NUMBER TOTAL  TOTAL  TOTAL   MEAN  MEAN   HOG    CHARS  BLOCKS
NAME    CMDS KCOREMIN CPUMIN REALMIN SIZEK CPUMIN FACTOR TRNSFD READ

TOTALS  9377  0.00    0.36  26632.67 0.00  0.00   0.00 17768213 100529

chmod     34  0.00    0.00       .15 0.00  0.00   0.07  5785856    64
ln         4  0.00    0.00      0.01 0.00  0.00   0.78   422016    16
xterm      9  0.00    0.03    537.41 0.00  0.00   0.00 22948288   536
getcons    8  0.00    0.00      0.14 0.00  0.00   0.07 26636992   102
cfe2.20    4  0.00    0.00      0.09 0.00  0.00   0.12   182464   155
dump      22  0.00    0.00     14.91 0.00  0.00   0.00 69402112   128
whoami     4  0.00    0.00      0.03 0.00  0.00   0.36  7405952    27
restore   40  0.00    0.00     49.16 0.00  0.00   0.00 34247488  1316
***other  25  0.00    0.00   3546.88 0.00  0.00   0.00 35904984   737
hostname   2  0.00    0.00      0.01 0.00  0.01   0.94   223104    14
```

The `HOG FACTOR` is the total CPU time divided by the total real time.

10.5.7 The acctprc1 Command

The `acctprc1` command reads process accounting records from files with the `/usr/include/sys/acct.h` header file structure, adds the login names that correspond to the user identification numbers, and displays the records in ASCII format. Login session records are sorted according to user identification number and login name.

If your system has users with the same user identification number, you should use a process accounting file in the `/var/adm/acct/nite` directory instead of the `pacct` file.

The `runacct` shell script invokes the `acctprc1` command during its PROCESS state. You can invoke the command manually. The `acctprc1` command has the following syntax:

/usr/sbin/acct/acctprc1 [*filename*]

The *filename* variable specifies a file that contains a list of login sessions in a format defined by the `/usr/include/utmp.h` header file structure. If the *filename* variable is not specified, login names are obtained from the `/etc/passwd` file.

The command output specifies information in a format with seven unheaded columns that specify the following:

- User identification number
- Login name
- Number of CPU seconds the process used during prime time

- Number of CPU seconds the process used during nonprime time
- Total number of characters transferred
- Total number of blocks read from and written to
- Mean memory size (in kilobytes)

The following is an example of the `acctprc1` command and its output:

```
# /usr/sbin/acct/acctprc1  < /usr/adm/pacct
   0   root      0     1   17228   172    6
   4   adm       0     6   46782    46   16
   0   root      0    22  123941   132   28
9261   hoffmann  6     0   17223    22   20
   9   lp        2     0   20345    27   11
9261   hoffmann  0   554   16554    20  234
```

10.5.8 The acctprc2 Command

The `acctprc2` command reads records produced by the `acctprc1` command, summarizes them according to user identification number and login name, and then uses the `tacct` file format to display the sorted summaries as total accounting binary records. You can merge the binary file produced by the `acctprc2` command with other total accounting files by using the `acctmerg` command to produce a daily summary accounting record file.

The `runacct` shell script invokes the `acctprc2` command during its PROCESS state. You can invoke the command manually also.

10.5.9 The lastcomm Command

The `lastcomm` command displays command execution information from the `pacct` file in reverse chronological order.

The following information is displayed for each process:

- Command name
- Either the `S` flag, which specifies that the command was invoked by the superuser; or the `F` flag, which specifies that the command ran after a fork but was not followed by an `exec` system call
- Name of the user who issued the command
- Terminal from which the command was started
- Number of seconds of CPU time used
- Time the process started

The `lastcomm` command has the following syntax:

/usr/bin/lastcomm [*command*] [*username*] [*tty*]

The following example displays information about the sed commands executed by root:

```
# lastcomm sed root
sed      S  root     ttyp0      0.01 secs Fri Jan 21 11:34
sed      S  root     ttyp0      0.01 secs Fri Jan 21 11:34
```

10.6 Disk Usage Accounting

Disk usage accounting is performed by the dodisk shell script. The dodisk shell script uses either the diskusg or the acctdusg command to write information to the intermediate ASCII file /var/adm/dtmp. The shell script then uses the intermediate file as input to the acctdisk command to create a binary total accounting database file, /var/adm/acct/nite/dacct. The dodisk script performs disk accounting on all or selected file systems specified in the /etc/fstab file system database file.

You can combine the total accounting information in the /var/adm/acct/nite/dacct file with other accounting information to create complete accounting reports. For example:

```
# /usr/sbin/acct/dodisk
# /usr/sbin/acct/prtacct /var/adm/acct/nite/dacct
```

10.6.1 The dodisk Shell Script

Use the dodisk shell script to obtain disk usage accounting. You can set up your cron daemon to run the dodisk script automatically, or you can invoke the command manually. The dodisk shell script has the following syntax:

/usr/sbin/acct/dodisk [–o] [*filesystem*…]

/usr/sbin/acct/dodisk [*device special file*…]

Using the -o option, you can specify the file system variable to perform disk usage accounting on the mount point of a UFS file system or an AdvFS fileset. If the -o option is not specified, the variable must be the raw or character device special file. For example:

```
# /usr/sbin/acct/dodisk /dev/rdisk/dsk3c
```

If you do not specify any arguments, disk accounting is performed on the UFS device special files decribed in the /etc/fstab database file. See fstab(4) for more information.

Note

If you have a swap space specified in the /etc/fstab file, the dodisk shell script does not execute correctly. In this case, you can edit the dodisk shell script to use only specific file systems

or you can invoke the dodisk shell script and specify the file systems for which you want accounting.

If you specify the -o option, the dodisk shell script uses the acctdusg command instead of the diskusg command to perform a more thorough but slower version of disk accounting. If you specify the -o option and a *filesystem* variable, specify the mount point instead of the device special file name.

10.6.2 The diskusg Command

The diskusg command displays disk accounting records. The diskusg command obtains user login names and identification numbers from the /etc/passwd file. The diskusg command has the following syntax:

/usr/sbin/acct/diskusg [–options] [*filesystems*]

See diskusg(8) for information on the command options.

The diskusg command produces ASCII output, which is directed to the /var/adm/dtmp file. This file is used as input to the acctdisk command, which converts the ASCII records to binary total accounting records in the /var/adm/acct/nite/dacct file. You can merge these records with other accounting records to create a daily total accounting report.

Each output record produced by the diskusg command contains the user identification number, login name, and the total number of disk blocks allocated to the user. Because the diskusg command verifies user inode records, all disk space is accounted for, including empty directories.

The following is an example of the diskusg command:

```
# /usr/sbin/acct/diskusg  /dev/disk/dsk3c
   0   root        63652
   1   daemon         84
   2   bin         71144
   4   adm           976
   5   uucp         3324
 322   homer           2
 521   whistler        2
 943   cellini       363
1016   pollock        92
1098   hopper        317
```

You must specify the raw device special file for a file system (for example, /dev/rdisk/dsk3c). A file system must exist on the target device.

10.6.3 The acctdusg Command

The acctdusg command performs more thorough disk accounting than the diskusg command. If dodisk is invoked with the -o option, the acctdusg command is used to create the /var/adm/dtmp file.

The acctdusg command has the following syntax:

acctdusg [–u *filename*] [–p *filename*]

See acctdusg(8) for information on the command options.

You must direct a binary disk usage file, usually /var/adm/dtmp, to the command. If the dodisk shell script invokes the command, the acctdusg command uses the file systems specified with the dodisk script as input.

The input to the acctdusg command is usually a list of files piped from a find / -print command. The command compares the file pathnames to the users' login directories ($HOME). If a file pathname is the same as a user's login directory, that user is charged for the file. Therefore, the directory in which the file is located is the determining factor in charging users for disk space. You can use the -u option to display the number of disk blocks used by files in directories other than the login directories.

For each file, the acctdusg command calculates the computed value, which is the number of disk blocks (including hidden or indirect blocks) that are allocated to the file divided by the number of hard links. If two or more users have links to the same file, the acctdusg command charges each user an equal percentage of the file's total disk space.

The acctdusg command output displays the user identification number, the user name, and the sum of the computed values of all the files owned by the user in three columns and adds leading 0s (zeros) to the user identification number. The acctdusg command does not display the disk-block count for empty directories.

10.6.4 The acctdisk Command

The acctdisk command creates a binary total accounting file. If it is invoked from the dodisk script, the acctdisk command reads the /var/adm/dtmp file that is produced by either the diskusg or acctdusg command. It then writes converted binary records to a temporary file, which is then moved to the /var/adm/acct/nite/dacct file.

The disk usage accounting records produced by the acctdisk command are merged usually with other accounting records to produce a total accounting report.

10.7 System Administration Service Accounting

You can charge users for system administration services. For example, you could charge for the following services:

- Backing up files to disk
- Recovering files from disk
- Backing up files to tape
- Recovering files from tape
- Providing software technical assistance by phone
- Providing software technical assistance in person

The `chargefee` shell script allows you to charge users according to the work performed. You should determine how much you want to charge for each service. Services can have different charge rates according to the time it takes to perform the task.

Charge units are collected in the `fee` file. You can use the number of units charged to a user name to determine the fees for the system administration tasks. The `chargefee` shell script creates the `fee` file, if necessary, and adds a record that includes the user identification number, user name, and charge units.

The `chargefee` shell script has the following syntax:

/usr/sbin/acct/chargefee *user_name units*

You can subtract units by specifying a dash (–) with the *`units`* variable.

The following example charges 7 units to user `josh`:

```
# chargefee josh 7
```

If the previous command is issued, the following record is written to the `/fee` file:

```
1114 josh 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0
```

10.8 Printer Accounting

When you use a printer that has accounting enabled, a record is written to the printer accounting file. Printer accounting records have a specific syntax and provide the following information:

- Name of the host and user that issued the print request
- Number of pages or feet of medium printed
- Number of times the printer was used

- Price per unit of printed output

The printer accounting records enable you to charge users for the system printing resources and to track printer usage.

The two printer accounting files are located in either the /var/adm or the /var/adm/printer directory. The *printer.*acct printer user file lists the amount and cost of print media used, according to machine and user name. The *printer.*acct_sum printer summary file lists a summary of media produced according to machine and user name. The *printer* variable specifies the printer name. See Chapter 8 for information on creating the printer accounting files.

Use the pac command to create a report of your printer activity. The pac command can obtain information only for printers that have accounting enabled. The pac command has the following syntax:

pac [-cmrs] [-p *price*] [-P *printer*] [*user*...]

See pac(8) for information on the command options.

10.9 Creating Daily, Summary, and Monthly Report Files

There are four shell scripts and one command that you can use to create daily, summary, and monthly report files in the /var/adm/acct/nite, /var/adm/acct/sum, and /var/adm/acct/fiscal directories, as shown in the following table:

Command	Description
runacct	This shell script creates the daily and summary files in the /var/adm/acct/nite and /var/adm/acct/sum directories.
acctmerg	This command merges total accounting record files and allows you to combine process connect time, fee, disk usage, and print queue accounting records into files whose format you specify. The output can be in either the default binary format or ASCII format and can include up to 18 columns of accounting information.
prtacct	This shell script formats and displays accounting files that have the /usr/include/sys/acct.h header file structure. Each record includes information about the user identification number, connect time, process time, disk usage, and printer usage.

Command	Description
prdaily	This shell script creates an ASCII file that contains the accounting data from the previous day. When this script is invoked from the runacct script, it creates the /var/adm/acct/sum/rprt*mmdd* file.
monacct	This shell script creates cumulative process and total accounting files in the /var/adm/acct/fiscal directory.

The following sections describe the shell scripts and the command in detail.

10.9.1 The runacct Shell Script

The runacct shell script uses accounting shell scripts and commands to process the connect time, fee, disk usage, queue, and process accounting database files to create the daily and summary files in the /var/adm/acct/nite and /var/adm/acct/sum directories.

The /var/adm/acct/nite directory contains files that are reused daily by the runacct script. Some of these files have binary counterparts in the /var/adm/acct/sum directory, which contains the cumulative summary files that are updated by the runacct shell script and used by the monacct shell script to produce monthly reports.

You can set up the cron daemon to invoke the runacct shell script each day, or you can invoke the runacct shell script manually. You may have to invoke the command manually if the runacct shell script does not run to completion or if a file created by the script becomes corrupted or lost.

When you invoke the runacct shell script it creates the /var/adm/acct/nite/lock temporary file. If the /var/adm/acct/nite/lock file exists, the runacct shell script does not run.

The runacct shell script executes in the following 13 states, in the order listed, and can be restarted at any of the 13 states:

State	Description
SETUP	Sets up some of the accounting files.
WTMPFIX	Fixes corrupted date and time stamp entries that can cause commands such as the acctcon1 command to fail.
CONNECT1	Writes connect session records.
CONNECT2	Uses the connect session records to create a binary total accounting record that are merged with other records to create a daily report.
PROCESS	Produces process accounting report files.

State	Description
MERGE	Uses the acctmerg command to create the binary total accounting file.
FEES	Uses the acctmerg command to merge records from the fee file into the binary total accounting file.
DISK	Uses the acctmerg command to merge disk-usage records into the binary total accounting file.
QUEUEACCT	Uses the acctmerg command to merge print queue accounting records into the binary total accounting file.
MERGEACCT	Copies the binary total accounting file to the daily total accounting file, which is used as input to the acctmerg command to create the cumulative total daily accounting file.
CMS	Produces command usage summaries.
USEREXIT	Invokes any site-specific shell scripts.
CLEANUP	Removes the temporary files.

10.9.1.1 Correcting runacct Shell Script Errors

If a runacct shell script error occurs, a message is written to the console device, the lock file is removed, the diagnostic files and error messages are saved, and processing is halted. Use the following information to determine if a runacct shell script error has occurred:

- The /var/adm/acct/nite/active file is created if the script has completed successfully. The runacct shell script logs messages to this file. You can use this file to determine which tasks have been completed successfully. The following is an example of an active file:

```
Fri Jul 5 11:02:56 EST 2002
-rw-r--r-- 1 adm  adm      0 Jul 01 03:00 /var/adm/acct/nite/dacct
-rw-rw-r-- 1 root system 924 Jun 05 10:45 /var/adm/wtmp
-rw-rw-r-- 1 adm  adm      0 Jun 08 13:46 fee
-rw-rw-r-- 1 adm  adm      0 Jun 07 02:00 pacct
-rw-rw-r-- 1 adm  adm   8904 Jun 02 11:02 pacct1
files setups complete
wtmp processing complete
connect acctg complete
process acctg complete for /var/adm/Spacct1.1101
process acctg complete for /var/adm/Spacct2.1101
all process acctg complete for 1101
tacct merge to create daytacct complete
no fees
no disk records
no queueing system records
updated sum/tacct
command summaries complete
system accounting completed at Fri
```

- The /var/adm/acct/nite/active*mmdd* file is created if the script has not successfully completed. This file contains information about the script execution; you can use it to determine where the script failed.
- The /var/adm/acct/nite/statefile file contains the name of the last state that the runacct shell script executed. The runacct shell script may not have completed this state successfully.
- The /var/adm/acct/nite/lastdate file contains the date of the last runacct shell script execution. If the date specified in the file is the current date, the shell script does not run.

If the runacct shell script fails or terminates before it is completed, you must restart the script from its last successfully completed state. The /var/adm/acct/nite/statefile file contains the name of the state that was last executed.

The runacct shell script has the following syntax:

/usr/sbin/acct/runacct [*mmdd*] [*state*]

The *mmdd* variable specifies the date for which you want to run the runacct shell script. Use the *state* variable to specify the state from which you want the runacct script to start processing.

If the runacct shell script fails on more than one successive day, invoke the SETUP state commands manually.

Before you restart the runacct shell script, you should remove the /var/adm/acct/nite/lock file and the /var/adm/acct/nite/lastdate file.

In the following example, the runacct shell script is invoked at its MERGE state and uses the accounting database files from January 26:

```
# runacct 0126 MERGE > /var/adm/nite/fd2log&
```

The following example invokes the runacct shell script, which uses the accounting database files from January 26 and specifies the nohup command so that signals, hangups, logouts, and quits are disregarded; any error messages generated during its execution are written to the fd2log file:

```
# nohup runacct 0126 > /var/adm/acct/nite/fd2log&
```

10.9.1.2 Examples of Errors and Corrective Actions

The following list provides examples of errors and the actions you can take to correct problems:

ERROR: locks found. run aborted

A /var/adm/acct/nite/lock file exists. Remove the file and restart the runacct shell script from its last completed state.

`ERROR: acctg already run for Fri :  check Jan`

The current date is the same as the date specified in the `/var/adm/acct/nite/lastdate` file. Remove the file and restart the `runacct` shell script from its last completed state.

`ERROR: runacct called with invalid arguments`

You have specified invalid arguments with the `runacct` shell script.

`ERROR: turnacct switch returned rc=?`

The `accton` command failed when it was invoked by the `turnacct switch` shell script. Verify the `accton` command protections and ensure that user `adm` can invoke the command.

`ERROR: Spacct?.mmdd already exists run setup manually`

You must invoke the `runacct` shell script manually from the `MERGE` state. The question mark (?) specifies a single-character wildcard. The actual file name includes a version number and a date such as `var/adm/Spacct1.1101`.

`ERROR: wtmpfix errors see nite/wtmperror`

An unrepairable `wtmp` file was found during the `WTMPFIX` state. Use the `fwtmp` command to correct the file.

`ERROR: invalid state, check /usr/var/adm/nite/active`

During processing, the `runacct` shell script may have detected a corrupted `active` file. Examine the `/var/adm/acct/nite/active*` and `statefile` files.

10.9.2 The acctmerg Command

The `acctmerg` command combines process, connect time, fee, disk-usage, and queue total accounting record files with the `tacct` file format. For example, you can merge the total accounting records for a particular login name and user identification number to provide a single group of records for that login name and user identification number. Usually, file records are merged according to the user identification number or the user login name.

The default command output is in binary format, but you can produce ASCII output. The default `acctmerg` command output has the `/usr/include/sys/acct.h` header file structure and includes up to 18 columns of accounting information. Records with the `/usr/include/sys/acct.h` header file structure that include data types

specified as an array of two double elements can have both prime time and nonprime time values.

The `runacct` shell script invokes the `acctmerg` command. You can invoke the command manually to produce reports. The `acctmerg` command has the following syntax:

/usr/sbin/acct/acctmerg [-ahiptuv] [#] [*file*...]

You can specify up to nine total accounting record files. If you do not specify a file, records are read from standard input.

See `acctmerg`(8) for information on command options.

The following example reads the UID, LOGNAME, DSK_BLOCKS, and DSAMPS column entries from the `/var/adm/acct/nite/dacct` ASCII disk accounting file. It then merges them into binary records in the `/var/adm/acct/sum/tacct` total accounting file.

```
# acctmerg  -i1-2, 13, 18 < nite/dacct | sum/tacct
```

You can use the `acctmerg` command to correct errors in the `/var/adm/sum/tacct` file. Errors that can occur in the file include negative numbers and duplicate user identification numbers.

To correct errors in the current `/var/adm/sum/tacct` file:

1. Change your directory to `/var/adm/sum`.
2. Enter the `prtacct` command to display the `/var/adm/sum/tacctprev` file. If the file is correct, then the problem probably is located in the `/var/adm/sum/tacct`*mmdd* file. This example assumes that the `/var/adm/sum/tacct`*mmdd* file needs to be fixed.
3. To obtain an ASCII version of the `/var/adm/sum/tacct`*mmdd* file, enter:

   ```
   # acctmerg –v < tacct.0617 > tacct_temp
   ```
4. Edit the temporary file and correct the records as necessary.
5. To recreate the `/var/adm/sum/tacct`*mmdd* file, enter:

   ```
   # acctmerg –i < tacct_temp > tacct.0617
   ```
6. To recreate the `/var/adm/sum/tacct` file, enter:

   ```
   # acctmerg tacctprev < tacct.0617 > tacct
   ```

10.9.3 The prtacct Shell Script

The `prtacct` shell script displays a binary total accounting file with the `tacct` file format in ASCII format. The script allows you to produce a connect time, process time, disk usage, or printer usage report file.

The monacct and prdaily shell scripts invoke the prtacct shell script. The runacct shell script invokes the prdaily shell script during its CLEANUP state. The prtacct shell script has the following syntax:

/usr/sbin/acct/prtacct [–f *column*] [–v] *file*

See prtacct(8) for information on the command options.

10.9.4 The prdaily Shell Script

The prdaily shell script creates an ASCII report of the accounting data from the previous day. The runacct shell script invokes the prdaily shell script during its CLEANUP state to create the /var/adm/acct/sum/rprt*mmdd* file. You can invoke the command manually to produce a report.

The prdaily script combines information from the following six accounting files:

- /var/adm/acct/nite/reboots
- /var/adm/acct/nite/lineuse
- /var/adm/acct/sum/tacct*mmdd*
- /var/adm/acct/nite/daycms
- /var/adm/acct/nite/cms
- /var/adm/acct/sum/loginlog

The prdaily shell script has the following syntax:

prdaily [-l[*mmdd*]] | [-c]

See prdaily(8) for more information on command options.

10.9.5 The monacct Shell Script

The monacct shell script uses the binary accounting files to create cumulative summary files in the /var/adm/acct/fiscal directory. After the summary files are produced, the command removes the old accounting files from the /var/adm/acct/sum directory and creates new files.

Usually, you run the monacct script once each month to produce monthly report files. You can set up your cron daemon to run the shell script automatically. See Section 10.2.4 for more information. The monacct shell script has the following syntax:

/usr/sbin/acct/monacct [*number*]

The *number* variable specifies an integer that is within the range 1 to 12 and that specifies the month for which you want to create the summary report. The default is the current month.

The `monacct` shell script creates the following files in the `/var/adm/acct/fiscal` directory:

`tacct`*mm*
: Specifies the binary total accounting file for the month preceding the month specified by the *mm* variable.

`cms`*mm*
: Specifies the binary cumulative command summary file for the month preceding the month specified by the *mm* variable.

`fiscrpt`*mm*
: Specifies the ASCII total monthly accounting report file. This file has a format that is similar to the `/var/adm/acct/sum/rprt`*mmdd* report file and is created from the following files:

  - `/var/adm/acct/fiscal/tacct`*mm*
  - `/var/adm/acct/fiscal/cms`*mm*
  - `/var/adm/acct/sum/loginlog`

11

Monitoring and Testing the System

The following topics are covered in this chapter:

- An overview of the basic monitoring guidelines and utilities, and pointers to related topics (Section 11.1)
- A detailed discussion of some of these monitoring utilities (Section 11.2)
- A discussion of environmental monitoring, which monitors aspects of system hardware status such as the temperature and whether the cooling fan is working; this feature depends on whether the hardware contains sensors that support such monitoring and not all systems support this feature (Section 11.3)
- A discussion on the use of the system component test utilities; your system hardware also provides test routines; see the Owner's Manual for more information (Section 11.4)

  If you need to obtain detailed information on the characteristics of system devices (such as disks and tapes) see the `hwmgr` command, documented in the *Hardware Management* manual.

11.1 Overview of Monitoring and Testing

System monitoring involves the use of basic commands and optional utilities to obtain baselines of operating parameters, such as the CPU workload or I/O throughput. Use these baselines to monitor, record, and compare ongoing system activity and ensure that the system does not deviate too far from your operational requirements.

Monitoring the system also enables you to predict and prevent problems that may make the system or its peripherals unavailable to users. Information from monitoring utilities enables you to react quickly to unexpected events such as system panics and disk crashes so that you can resolve problems quickly and bring the system back on line.

The topic of monitoring is related closely to your technical support needs. Some of the utilities described in this chapter have a dual function. Apart from realtime system monitoring, they also collect historical and event-specific data that is used by your technical support representative. This data can be critical for getting your system up and running quickly after

a fault in the operating system or hardware. Therefore, it is recommended that you follow the monitoring guidelines in Section 11.1.1 at the very least.

Testing involves the use of commands and utilities to exercise parts of the system or peripheral devices such as disks. The available test utilities are documented in this chapter. Your system hardware also provides test utilities that you run at the console prompt. See your Owner's manual for information on hardware test commands.

Section 11.1.1 provides general guidelines for monitoring your system, and Section 11.1.2 gives a brief overview of all the utilities that the operating system provides.

11.1.1 Guidelines for Monitoring Systems

Use the following procedure after you configure your system exactly as required for its intended operation:

1. Choose the utilities to monitor your system on a daily basis.

   Review the overview of monitoring utilities described in Section 11.1.2. Based on the system configuration, select utilities that satisfy the requirements of the configuration and your monitoring needs. For example, if you have a graphics terminal and you want to monitor several distributed systems, consider setting up the SysMan Station. If you want to monitor a single local server, the `dxsysinfo` window may be adequate.

   If applicable, set any attributes that trigger warnings and messages. For example, you may choose to set a limit of 85% full on all file systems to prevent loss of data because of a full device.

   Note

   Many optional subsystems provide their own monitoring utilities. Familiarize yourself with these interfaces and decide whether they are more appropriate than the generic utilities.

2. Establish a baseline.

   Run the `sys_check` utility with the `-all` option:

   - To establish a no-load baseline.
   - To determine whether any system attributes should be tuned.

   If necessary, use the information from the `sys_check` utility to tune system attributes. See the *System Configuration and Tuning* manual for information on Tuning your system. Store the baseline data where it

can be accessed easily later, such as on another system. Also, print a copy of the report.

3. Run the `sys_check` utility under load.

   At an appropriate time, run the `sys_check` utility when the system is under a reasonable workload. Choose only those options, such as `-perf`, that you want to monitor. This may have a small affect on system performance, so you may not want to run it during peak end-user demand.

   Analyze the output from the `sys_check` utility and perform any additional recommended changes that satisfy your operational requirements. This may involve further tuning of system attributes or configuration changes, such as the reallocation of system resources using a utility like the Class Scheduler. See Section 11.2.2 for information on using the `sys_check` utility.

4. Set up the Event Manager.

   Configure the event management logging and reporting strategy for the system in conjunction with whatever monitoring strategy you employ. See Chapter 13 and Chapter 12 for information on how to configure the Event Manager.

5. Configure monitoring utilities.

   Set up any other monitoring utilities that you want to use. For example:

   - Configure the `sys_check` utility to run regularly during off-peak hours by using the `runsyscheck` script with the `cron` utility as described in Section 11.2.2. In the event of a system problem, the regularly-updated report is useful when analyzing and troubleshooting the problem.

     Note

     Crash dump data may be required for diagnosing system problems. See Chapter 14 for information on configuring the crash dump environment.

   - Install and configure any optional performance utilities. If supported by the target system, configure environmental monitoring, as described in Section 11.3.

11.1.2 Summary of Commands and Utilities

The operating system provides a number of monitoring commands and utilities. Some commands return a simple snapshot of system data in numerical format, while others have many options for selecting and filtering

information. Also provided are complex graphical user interfaces that filter and track system data in real time and display it on a graphics terminal.

Choose monitoring utilities that suit your local environment and monitoring needs and consider the following:

- Using monitoring utilities can affect system performance.
  - To help diagnose problems in performance, such as I/O bottlenecks, a simple command, like `iostat`, may be adequate.
  - To provide a quick visual examination of resources on a single-user system, the X11 System Information interface (`dxsysinfo`) may be adequate.
- Some utilities are restricted to the root user while others are accessible by all system users.
- For enterprise-wide monitoring, the SysMan Station can display the health of many systems simultaneously on a single screen.
- To track assets across an enterprise or verify what options are installed in what systems (and verify whether they are functioning correctly), the web-based HP Insight Manager utility can be used for both UNIX servers and client PC systems.
- You may need to provide output from a monitoring utility to your technical support site during problem diagnosis. It reduces your system downtime greatly if you take a system baseline and establish a routine monitoring and data collection schedule before any problems occur.

The following sections describe the monitoring utilities.

11.1.2.1 Command Line Utilities

Use the following commands to display a snapshot of various system statistics:

`vmstat`

The `vmstat` command displays system statistics for virtual memory, processes, trap, and CPU activity. An example of `vmstat` output is:

```
bigrig> vmstat
Virtual Memory Statistics: (pagesize = 8192)
procs   memory         pages                            intr       cpu
r  w  u act  free wire fault cow zero react pin pout in  sy  cs us sy id
2 97 20 8821  50K 4434 653K 231K 166K 1149 142K    0 76 250 194  1  1 98
```

See `vmstat`(1) for more information.

iostat

The iostat command reports input and output information for terminals and disks and the percentage of time the CPU has spent performing various operations. An example of iostat output is:

```
bigrig> iostat
    tty      floppy0          dsk0          cpu
 tin tout    bps    tps    bps    tps  us ni sy id
   0    1      0      0      3      0   0  0  1 98
```

See iostat(1) for more information.

who

The who command reports information about users and processes on the local system. An example of who output is:

```
bigrig> who
# who
root        console     Jan  3 09:55
root        :0          Jan  3 09:55
root        pts/1       Jan  3 09:55
bender      pts/2       Jan  3 14:59
root        pts/3       Jan  3 15:43
```

There is a similar command, users, that displays a compact list of the users logged in.

See who(1) and users(1) for more information.

uptime

The uptime command reports how long the system has been running.

```
bigrig> uptime
16:20  up 167 days, 14:33,  4 users,  load average: 0.23, 0.24, 0.24
```

See uptime(1) for more information.

There is a similar command, w, that displays the same information as uptime, but also displays information for the users logged in. See w(1) for more information.

netstat

The netstat command displays network-related statistics in various formats.

See the netstat command and the *Network Administration: Connections* manual for information on monitoring your network.

11.1.2.2 SysMan Menu Monitoring and Tuning Tasks

The SysMan Menu provides options for several monitoring tasks. See Chapter 1 for general information on using the SysMan Menu. The following options are provided under the Monitoring and Tuning menu item:

View Events [event_viewer]

This option invokes the EVM event viewer, which is described in Chapter 13.

Set up HP Insight Manager [imconfig]

This option invokes the interface that enables you to configure HP Insight Manager and start the HP Insight Manager daemon. See Chapter 1 for information on configuring HP Insight Manager.

View Virtual Memory (VM) Statistics [vmstat]

This is a SysMan Menu interface to the `vmstat` command, described in Section 11.1.2.1.

View Input/Output (I/O) Statistics [iostat]

This is a SysMan Menu interface to the `iostat` command, described in Section 11.1.2.1.

View Uptime Statistics [uptime]

This is a SysMan Menu interface to the `uptime` command, described in Section 11.1.2.1.

In addition, the following options are provided under the Support and Services menu item:

Create escalation Report [escalation]

This option invokes the escalation report feature of the `sys_check` utility. The escalation report is used only in conjunction with diagnostic services, and is requested by your technical support organization. See Section 11.2.2 for more information on using the escalation options in the `sys_check` utility.

Create configuration Report [config_report]

This option invokes the system configuration report feature of the `sys_check` utility. Use this option to create a baseline record of your system configuration and to update the baseline at regular intervals. Using this option creates a full default report which can take

many minutes to complete and can affect system performance. See Section 11.2.2 for more information on using the `sys_check` utility.

11.1.2.3 SysMan Station

The SysMan Station provides a graphical view of one or more systems and also enables you to launch applications to perform administrative operations on any component. See Chapter 1 for information on using the SysMan Station.

11.1.2.4 X11-Compliant Graphical User Interfaces

The operating system provides several graphical user interfaces (GUIs) that are used typically under the default Common Desktop Environment (CDE) windowing environment; they are located in the System Management folders.

You can invoke these interfaces from the SysMan Applications panel on the CDE Front Panel; Figure 1–5 shows the SysMan Applications Panel. There are icons that link to the Monitoring/Tuning folder, the Tools folder, and the Daily Admin folder.

Monitoring/Tuning Folder

This folder provides icons that invoke the following SysMan Menu items:

Configuration Report

This icon invokes a graphical user interface to the system configuration report feature of the `sys_check` utility.

Escalation Report

This icon invokes a graphical user interface to the escalation report feature of the `sys_check` utility.

HP Insight Manager

This icon invokes the interface that enables you to configure HP Insight Manager and start the HP Insight Manager daemon.

The remaining applications in this folder relate to system tuning. See the *System Configuration and Tuning* manual for information on tuning using the Process Tuner (a graphical user interface to the `nice` command) and the Kernel Tuner (`dxkerneltuner`).

Tools Folder

The Tools folder provides graphical user interfaces to the commands such as `vmstat`. Invoke these interfaces from the CDE Front Panel by selecting the Application Manager icon to display the Application Manager folder. From this folder, select the System Admin icon, and then the Tools icon. This folder provides the following interfaces:

I/O Statistics

> This is a graphical user interface to the `iostat` command, described previously in Section 11.1.2.1.

Network Statistics

> This is a graphical user interface to the `netstat` command. See the *Network Administration: Connections* manual for information on monitoring your network.

System Messages

> This is a graphical user interface to the `/var/adm/messages` log file, which stores certain system messages according to the current configuration of system event management. For information on events, the messages they generate, and the message log files, see Chapter 12 and Chapter 13.

Virtual Memory Statistics

> This is a graphical user interface to the `vmstat` command, described in Section 11.1.2.1.

Who?

> This is a graphical user interface to the `who` command, described in Section 11.1.2.1.

Daily Admin Folder

The remaining X11–compliant monitoring application is System Information, which is located in the Application Manager – DailyAdmin folder.

System Information

> Select the System Information (`dxsysinfo`) icon to launch the interface. This interface provides you with a quick view of the following system resources and data:
>
> - A brief description of the number and type of processors (CPUs).

- The UNIX operating system version and the amount of available system memory.
- Three dials indicating approximate amount of CPU activity, in-use memory, and in-use virtual memory (swap). This information can be obtained also by using commands such as `vmstat`.
- Two warning indicators for files and swap. These indicators change color when a file system is nearly full or if the amount of swap space is too low.
- The current available space status of all local and remotely-mounted file systems. You can set a percentage limit here to trigger the warning indicators if available space falls below a certain percentage. See Chapter 6 and Chapter 9 for information on increasing the available file system space.

11.1.2.5 Advanced Monitoring Utilities

The following utilities provide options that enable you to view and record many different operating parameters:

Collect

The `collect` utility enables you to sample many different kinds of system and process data simultaneously over a predetermined sampling time. You can collect information to data files and play the files back at the terminal.

The `collect` utility can assist you in diagnosing performance problems and its report output may be requested by your technical support service when they are assisting you in solving system problems. Using the `collect` utility is described in Section 11.2.1.

See the *Collect User's Guide*, which is included with the installation kit, for more information.

The `sys_check` utility

The `sys_check` utility is a command line interface that you use to create a permanent record of the system configuration and the current settings of many system attributes. This utility is described in detail in Section 11.2.2.

The Monitoring Performance History (MPH) Utility

The Monitoring Performance History (MPH) utility is a suite of shell scripts that gathers information on the reliability and availability of the operating system and its hardware environment such as crash data files. This utility is described in detail in Section 11.2.3.

11.1.3 Related Documentation

The following topics in this manual are related closely to system monitoring and testing:

- See Chapter 10 for information on administering the system accounting services, which enables you to monitor and record access to resources such as printers.
- See Chapter 12 for instructions on configuring and using basic system event logging by using the basic `binlogd` and `syslogd` event channels. This chapter also describes how you access system log files, where events and errors are recorded.
- See Chapter 13 for information on configuring and using the Event Manager (EVM), which provides sophisticated management of system events, including automated response to certain types of event.

These manuals also provide additional information related to system monitoring and testing:

- See the *System Configuration and Tuning* manual for information on tuning your system in response to information gathered during monitoring and testing.
- See the *Network Administration: Connections* manual for information on monitoring the system's networking components.

11.2 Configuring and Using Monitoring Utilities

The following sections introduce some of the monitoring utilities and describes their setup and use. See the documentation and reference pages supplied with each application for more information. See Chapter 1 for information on configuring and using the SysMan Station to monitor systems that have a graphics environment.

A closely related topic is event management and error logging. See Chapter 12 and Chapter 13 for information on these topics.

11.2.1 Using the collect Utility to Record System Data

The `/usr/sbin/collect` command line utility collects data that describes the current system status. It enables you to select from many parameters and sort them and to time the data collection period. The data is displayed in real time or recorded to a file for future analysis or playback. Using the `collect` utility has a low CPU overhead because you can focus on the exact aspects of system behavior that you need to record and therefore it should not adversely effect system performance.

The output from the unqualified /usr/sbin/collect command is similar to the output from monitoring commands such as vmstat, iostat, or netstat.

The command synopsis is defined fully in collect(8). Important features provided by the collect utility are:

- Controlling the duration of, and rate at which data is sampled. Sorting the output according to processor usage.
- Extracting a time slice of data from a data record file. For example, if you want to look at certain system parameters during the busiest time of use, you can extract that data from the data file by using the -C option.
- Specifying a particular device using its device special file name. For example the following command identifies that data is collected from the named devices:

  ```
  # collect -sd -Ddsk1,dsk10
  ```

- Specifying a particular subsystem such as the CPU or the network. For example, the following command specifies that data is collected only for the CPUs, and a sample of data is shown:

  ```
  # collect -e cf
  CPU SUMMARY
  USER SYS IDLE WAIT INTR SYSC CS RUNQ AVG5 AVG30 AVG60 FORK VFORK
    13  16   71    0  149  492 725   0 0.13 0.05  0.01 0.30 0.00
  SINGLE CPU STATISTICS
    CPU USER  SYS IDLE WAIT
      0   13   16   71    0
  ```

- Recording and preserving a series of data files by using the -H (history) option.
- Compressing data files for economical storage.
- Specifying specific users, groups, and processes for which data is to be sampled.
- Playing back data files with the -p option. The -f option lets you combine multiple binary input files into one binary output file.

The collect utility locks itself into memory by using the page locking function plock(), and cannot be swapped out by the system. It also raises its priority by using the priority function nice(). If required, page locking can be disabled by using the -ol command option and the priority setting can be disabled by using the -on command option. However, using the collect utility should have minimal affect on a system under high load.

11.2.2 Using the sys_check Utility

The sys_check utility provides you with the following:

- The ability to establish a baseline of system configuration information, both for software and hardware and record it in an easily accessible HTML report for web browsing. You can recreate this report regularly or as your system configuration changes.
- The opportunity to perform automated examination of many system attributes (such as tuning parameters) and receive feedback on settings that may be more appropriate to the current use of the system.

  The `sys_check` utility examines and reports recommended maintenance suggestions, such as installing patch kits and maintaining swap space.
- The ability to generate a problem escalation report that can be used by your technical support service to diagnose and correct system problems.

In addition to recording the current hardware and software configuration, the `sys_check` utility produces an extensive dump of system performance parameters. This feature enables you to record many system attribute values, providing a useful baseline of system data. Such a baseline is particularly useful before you undertake major changes or perform troubleshooting procedures.

When you run the `sys_check` utility, it produces an HTML document on standard output. Used with the `-escalate` flag, the script produces `$TMPDIR/escalate*` output files by default, where the environmental variable `$TMPDIR` determines the temporary directory. These files can be forwarded to your technical support organization and used for diagnosing system problems and errors.

Use the following command to obtain a complete list of command options.

```
# /usr/sbin/sys_check -help
```

The output produced by the `sys_check` utility typically varies between 0.5MB and 3MB in size and it can take from 30 minutes to an hour to complete the examination. See `sys_check`(8) for more details of the various command options. You can reduce the run time greatly by excluding items from the run. For example, the `sys_check` utility runs `setld` to record the installed software. Excluding the `setld` operation can greatly reduce the `sys_check` run duration.

You can invoke standard `sys_check` run tasks as follows:

- Using CDE, open the Application Manager from the CDE front panel. Select System_Admin and then MonitoringTuning. There are icons for two standard `sys_check` run tasks, Configuration Report and Escalation Report.
- Using the SysMan Menu, expand the Support and Services menu item and choose from the following options:
  - Create escalation report

– Create configuration report

For information on using the SysMan Menu, see Chapter 1.

You can run `sys_check` tasks automatically by enabling an option in the root `crontabs` file. In the `/var/spool/cron/crontabs` directory, the `root` file contains a list of default tasks that are run by `cron` on a regular basis. Remove the comment (#) command from the following line:

```
#0 3 * * 0 /usr/share/sysman/bin/runsyscheck
```

When this option is enabled, the resulting report is referenced by HP Insight Manager and can be read from the Tru64 UNIX Configuration Report icon on the HP Insight Manager Device Home Page. See Chapter 1 for information on using HP Insight Manager.

11.2.3 Using the Monitoring Performance History Utility

The Monitoring Performance History (MPH) utility is a suite of shell scripts that gathers information, such as crash data files, on the reliability and availability of the operating system and its hardware environment. The information is copied automatically to your systems vendor by Internet mail or a DSN link, if available. Using this data, performance analysis reports are created and distributed to development and support groups. This information is used internally only by your systems vendor to improve the design of reliable and highly available systems.

The MPH run process is automatic, requiring no user intervention. Initial configuration requires approximately ten minutes. MPH does not affect or degrade your system's performance because it runs as a background task, using negligible CPU resources. The disk space required for the collected data and the application is approximately 300 blocks per system. This could be slightly higher in the case of a high number of errors and is considerably larger for the initial run, when a baseline is established; this is a one-time event.

The MPH utility operates as follows:

- Every 10 minutes it records a timestamp indicating that the system is running.
- Daily at 2:00 A.M., it extracts any new events records from the default event log `/var/adm/binary.errlog`.
- Every day at 3:00 A.M. it transfers the event and time stamp data and any new `crashdc` data files in `/var/adm/crash` to the system vendor. The average transfer is 150 blocks of data.

Before running MPH, review the following information:

- The Standard Programmer Commands (Software Development) OSFPGMR400 subset must be installed. Use the `setld -i` command to verify that the subset is installed.
- The MPH software kit is contained in the mandatory base software subset OSFHWBASE400. This subset is installed automatically during the operating system installation. Full documentation is located in `/usr/field/mph/unix_installation_guide.ps`. A text file is also supplied.
- The disk space requirement for the MPH software subset is approximately 100 blocks.

To configure MPH on your system, you must be the root user and the principal administrator of the target system. You need to supply your name, telephone number, and e-mail address. Complete the following steps:

1. Find the serial number (SN) of the target system, which is generally located on the rear of the system box. You need this number to complete the installation script.
2. Enter the following command to run the MPH script:

   ```
   # /usr/field/mph/MPH_UNIX***.CSH
   ```

   In this example, ***** denotes the version number, such as 025.
3. Enter the remaining information requested by the script. When the script is complete, MPH starts automatically.

If the operating system needs to be shut down for any reason, an orderly shutdown process must be followed. Otherwise, you need to restart the MPH script as described in the MPH documentation. See `mph`(1) for more information.

11.3 Environmental Monitoring and envmond/envconfig

On any system, thermal levels can increase because of poor ventilation, overheating, or fan failure. Without detection, an unscheduled shutdown could occur, causing a loss of data, damage to the system, or both. By monitoring the system's environment, you can be forewarned in time to recover or perform a gradual and orderly system shutdown.

Environmental Monitoring from the Command Line

You can monitor the environmental status of your system by using the `sysconfig` command line utility, as follows:

```
# /sbin/sysconfig -q envmon
```

This command reports the current temperature, fan status, and other general environmental information.

Environmental Sensor Monitoring

On a limited number of recent hardware platforms there also exists the capability to monitor thermal environment sensors for processor temperature and fan speed thresholds. Issue the following command as root to determine if your hardware platform supports these features.

```
# /sbin/hwmgr view hier | grep sensor
  52:     sensor systhermal-cpu0
  53:     sensor systhermal-cpu1
  54:     sensor systhermal-cpu2
  55:     sensor systhermal-pci_zone-1
  56:     sensor systhermal-pci_zone-2
  57:     sensor systhermal-pci_zone-3
  58:     sensor sysfan-pci_zone-1/2
  59:     sensor sysfan-power_supply-3/4
  60:     sensor sysfan-cpu_memory-5/6
  61:     sensor syspower-ps-0
  62:     sensor syspower-ps-1
  63:     sensor syspower-ps-2
```

The output returned by this command indicates the individual active sensors.

If there is no output, these features are not supported on your platform.

These sensors post events to the Event Manager when a threshold is reached. By reading the Event Manager log file and using the Hardware Manager command line program, you can determine which component is becoming critical. See Chapter 13 for information on Event Manager; see the *Hardware Management* manual and `hwmgr`(8) for more information on the Hardware Manager command line utility.

HP Insight Manager

HP Insight Manager also provides a method for monitoring the environmental conditions through the Recovery->Environment display. See *HP Management Agents for AlphaServers for Tru64 UNIX* for more information.

envmond/envconfig Framework

There exists an Environmental Monitoring framework that consists of four components:

- The loadable kernel module and its associated APIs
- The Server System MIB subagent daemon
- The `envmond` daemon

  The `envmond` daemon is used to monitor the system environment. See `envmond`(8) for more information.
- The `envconfig` utility

The `envconfig` utility lets you customize the `envmond` daemon. See `envconfig`(8) for more information.

These components are described in the following sections.

11.3.1 Loadable Kernel Module

The loadable kernel module and its associated APIs contain the parameters needed to monitor and return status on your system's threshold levels. The kernel module exports server management attributes as described in Section 11.3.1.1 through the kernel configuration manager (CFG) interface only. It works across all platforms that support server management, and provides compatibility for other server management systems under development.

The loadable kernel module does not include platform-specific code (such as the location of status registers). It is transparent to the kernel module which options are supported by a platform. That is, the kernel module and platform are designed to return valid data if an option is supported, a fixed constant for unsupported options, or null.

11.3.1.1 Specifying Loadable Kernel Attributes

The loadable kernel module exports the parameters listed in Table 11–1 to the kernel configuration manager (CFG).

Table 11–1: Parameters Defined in the Kernel Module

Parameter	Purpose
`env_current_temp`	Specifies the system's current temperature. If a system is configured with the KCRCM module, the temperature returned is in Celsius. If a system does not support temperature readings and a temperature threshold is not exceeded, a value of -1 is returned. If a system does not support temperature readings and a temperature threshold is exceeded, a value of -2 is returned.
`env_high_temp_thresh`	Provides a system-specific operating temperature threshold. The value returned is a hardcoded, platform-specific temperature in Celsius.
`env_fan_status`	Specifies a noncritical fan status. The value returned is a bit value of zero (0). This value differs when the hardware support is provided for this feature.

Table 11–1: Parameters Defined in the Kernel Module (cont.)

Parameter	Purpose
env_ps_status	Provides the status of the redundant power supply. On platforms that provide interrupts for redundant power supply failures, the corresponding error status bits are read to determine the return value. A value of 1 is returned on error; otherwise, a value of zero (0) is returned.
env_supported	Indicates whether or not the platform supports server management and environmental monitoring.

11.3.1.2 Obtaining Platform-Specific Functions

The loadable kernel module must return environmental status based on the platform being queried. To obtain environmental status, the get_info() function is used. Calls to the get_info() function are filtered through the platform_callsw[] table.

The get_info() function obtains dynamic environmental data by using the function types described in Table 11–2.

Table 11–2: get_info() Function Types

Function Type	Use of Function
GET_SYS_TEMP	Reads the system's internal temperature on platforms that have a KCRCM module configured.
GET_FAN_STATUS	Reads fan status from error registers.
GET_PS_STATUS	Reads redundant power supply status from error registers.

The get_info() function obtains static data by using the HIGH_TEMP_THRESH function type, which reads the platform-specific upper threshold operational temperature.

11.3.2 Server System MIB Subagent

The Server System MIB Agent (an eSNMP subagent) exports a subset of the Environmental Monitoring parameters specified in the Server System MIB. The Server System MIB exports a common set of hardware-specific parameters across all server platforms, depending on the operating system installed.

Table 11–3 maps the subset of Server System MIB variables that support Environmental Monitoring to the kernel parameters described in Section 11.3.1.1.

Table 11–3: Mapping of Server Subsystem Variables

Server System MIB Variable Name	Kernel Module Parameter
svrThSensorReading	env_current_temp
svrThSensorStatus	env_current_temp
svrThSensorHighThresh	env_high_temp_thresh
svrPowerSupplyStatus	env_ps_temp
svrFanStatus	env_fan_status

An SNMP MIB compiler and other utilities are used to compile the MIB description into code for a skeletal subagent daemon. Communication between the subagent daemon and the master agent eSNMP daemon, `snmpd`, is handled by interfaces in the eSNMP shared library (`libesnmp.so`). The subagent daemon must be started when the system boots and after the eSNMP daemon has started.

The subagent daemon contains code for Server System MIB variable listed in Table 11–3. The daemon accesses the appropriate parameter from the kernel module through the CFG interface.

11.3.3 Environmental Monitoring Daemon

Use the Environmental Monitoring daemon, `envmond`, to examine threshold levels and take corrective action before damage occurs to your system. Then the `envmond` daemon performs the following tasks:

- It queries the system for threshold levels.
- It begins a system shutdown when a cooling fan fails.

  On the AlphaServer 1000A fails, the kernel logs the error and synchronizes the disks before it powers down the system.

  On all other fan failures, a hard shutdown occurs.
- It notifies users when a high temperature threshold condition is resolved.
- It notifies all users that an orderly shutdown is in progress if recovery is not possible.

To query the system, the `envmond` daemon uses the base operating system command `/usr/sbin/snmp_request` to obtain the current values of the environment variables specified in the Server System MIB.

To enable Environmental Monitoring, the eSNMP and Server System MIB agents must be started during system boot followed by the enabling of the the envmond daemon, also during system boot.

See envmond(8) for more information.

11.3.4 Using envconfig to Configure the envmond Daemon

You can use the envconfig utility to customize how the envmond daemon queries the environment. These customizations are stored in the /etc/rc.config file, which is read by the envmond daemon during startup.

Use the envconfig utility to perform the following tasks:

- Turning environmental monitoring on or off during the system boot.
- Starting or stopping the envmond daemon after the system boot.
- Specifying the frequency between system queries by the envmond daemon.
- Setting the highest threshold level that can be encountered before a temperature event is signaled by the envmond daemon. Specify the path of a user-defined script that you want the envmond daemon to execute when a high threshold level is encountered.
- Specifying a grace period for saving data if a shutdown message is broadcast.
- Displaying the values of the Environmental Monitoring variables.

See envconfig(8) for more information.

11.3.5 User-Definable Messages

You can modify messages broadcasted or logged by the Environmental Monitoring utility. The messages are located in the following file:

```
/usr/share/sysman/envmon/EnvMon_UserDefinable_Msg.tcl
```

You must be root to edit this file; you can edit any message text included in braces ({}). The instructions for editing each section of the file are included in the comment fields, preceded by the # symbol.

For example, the following message provides samples of possible causes for the high temperature condition:

```
set EnvMon_Ovstr(ENVMON_SHUTDOWN_1_MSG){System has reached a \
high temperature condition. Possible problem source: Clogged \
air filter or high ambient room temperature.}
```

You could modify this message text as follows:

```
set EnvMon_Ovstr(ENVMON_SHUTDOWN_1_MSG) {System \
has reached a high temperature condition. Check the air \
conditioning unit}
```

Do not alter any data in this file other that the text strings between the braces ({}).

11.4 Using System Exercisers

The operating system provides a set of exercisers that you can use to troubleshoot your system. The exercisers test specific areas of your system, such as file systems or system memory. The following sections provides information on the system exercisers:

- Running the system exercisers (Section 11.4.1)
- Using exerciser diagnostics (Section 11.4.2)
- Exercising file systems by using the `fsx` command (Section 11.4.3)
- Exercising system memory by using the `memx` command (Section 11.4.4)
- Exercising shared memory by using the `shmx` command (Section 11.4.5)
- Exercising communications systems by using the `cmx` command (Section 11.4.6)

Additionally, you can exercise disk drives by using the `diskx` command and tape drives by using the `tapex` command. For more information, see `diskx`(8) and `tapex`(8), respectively, for more information.

In addition to the exercisers documented in this chapter, your system may support the Verifier and Exerciser Tool (VET), which provides a similar set of exercisers. See the documentation that came with your latest firmware CD-ROM for information on VET.

11.4.1 Running System Exercisers

To run a system exerciser, you must be logged in as superuser and your current directory must be `/usr/field`.

The commands that invoke the system exercisers provide an option for saving the diagnostic output into a specified file when the exerciser completes its task.

Most of the exerciser commands have an online help option that displays a description of how to use that exerciser. To access online help, use the `-h` option with a command. For example, to access help for the `diskx` exerciser, use the following command:

```
# diskx –h
```

You can run the exercisers in the foreground or the background; you can cancel them at any time by entering Ctrl/c in the foreground. You can run more than one exerciser at the same time; keep in mind, however, that the more processes you have running, the slower the system performs. Thus, before exercising the system extensively, make sure that no other users are on the system.

There are some restrictions when you run a system exerciser over an NFS link or on a diskless system. Exercisers, such as `fsx`, need to write to a file system, so the target file system must be writable by root. Also, the directory from which an exerciser is executed must be writable by root because temporary files are written to the directory.

These restrictions can be difficult to adhere to because NFS file systems are often mounted in a way that prevents root from writing to them. You can overcome some of these problems by copying the exerciser into another directory and running it from the new directory.

11.4.2 Using Exerciser Diagnostics

When an exerciser is halted (either by entering Ctrl/c or by timing out), diagnostics are displayed and are stored in the exerciser's most recent log file. The diagnostics inform you of the test results.

Each time an exerciser is invoked, a new log file is created in the `/usr/field` directory. For example, when you execute the `fsx` command for the first time, a log file named `#LOG_FSX_01` is created. The log files contain records of each exerciser's results and consist of the starting and stopping times, and error and statistical information. The starting and stopping times are also logged into the default `/var/adm/binary.errlog` system error log file. This file also contains information on errors reported by the device drivers or by the system.

The log files provide a record of the diagnostics. However, be sure to delete a log file after reading it because an exerciser can have only nine log files. If you attempt to run an exerciser that has accumulated nine log files, the exerciser tells you to remove some of the old log files so that it can create a new one.

If an exerciser finds errors, you can determine which device or area of the system is experiencing difficulty by looking at the `/var/adm/binary.errlog` file, using either the `dia` command (this is preferred) or the `uerf` command. For information on the error logger, see the Section 12.1. For the meanings of the error numbers and signal numbers, see `intro`(2) and `sigvec`(2).

11.4.3 Exercising a File System

Use the `fsx` command to exercise the local file systems. The `fsx` command exercises the specified local file system by initiating multiple processes, each of which creates, writes, closes, opens, reads, validates, and unlinks a test file of random data.

Note

Do not test NFS file systems with the `fsx` command.

The `fsx` command has the following syntax:

fsx [-f*path*] [–h] [–o*file*] [–p*num*] [–t*min*]

See `fsx`(8) for a description of the command options.

The following example of the `fsx` command tests the `/usr` file system with five `fsxr` processes running for 60 minutes in the background:

```
# fsx -p5 -f/usr -t60 &
```

11.4.4 Exercising System Memory

Use the `memx` command to exercise the system memory. The `memx` command exercises the system memory by initiating multiple processes. By default, the size of each process is defined as the total system memory in bytes divided by 20. The minimum allowable number of bytes per process is 4,095. The `memx` command writes and reads ones and zeroes, zeroes and ones, and random data patterns in the allocated memory being tested.

The files that you need to run the `memx` exerciser include the following:

- `memx`
- `memxr`

The `memx` command is restricted by the amount of available swap space. The size of the swap space and the available internal memory determine how many processes can run simultaneously on your system. For example, if there are 16 MB of swap space and 16 MB of memory, all the swap space is used if all 20 initiated processes (the default) run simultaneously. This prevents execution of other process. Therefore, on systems with large amounts of memory and small amounts of swap space, you must use the `-p` or `-m` option, or both, to restrict the number of `memx` processes or to restrict the size of the memory being tested.

The `memx` command has the following syntax:

memx -s [-h] [-m*size*] [-o*file*] [-p*num*] [-t*min*]

See memx(8) for a description of the command options.

The following example of the memx command initiates five memxr processes that test 4,095 bytes of memory and runs in the background for 60 minutes:

```
# memx –m4095 –p5 –t60 &
```

11.4.5 Exercising Shared Memory

Use the shmx command to exercise the shared memory segments. The shmx command spawns a background process called shmxb. The shmx command writes and reads the shmxb data in the segments, and the shmxb process writes and reads the shmx data in the segments.

Using shmx, you can test the number and the size of memory segments and shmxb processes. The shmx exerciser runs until the process is killed or until the time specified with the -t option is exhausted.

You invoke the shmx exerciser automatically when you start the memx exerciser, unless you specify the memx command with the -s option. You can also invoke the shmx exerciser manually. The shmx command has the following syntax:

/usr/field/shmx [-h] [-o*file*] [-v] [-t*time*] [-m*size*] [-s*n*]

See shmx(8) for a description of the command options.

The following example tests the default number of memory segments, each with a default segment size:

```
# shmx &
```

The following example runs three memory segments of 100,000 bytes for 180 minutes:

```
# shmx -t180 -m100000 -s3 &
```

11.4.6 Exercising the Terminal Communication System

Use the cmx command to exercise the terminal communications system. The cmx command writes, reads, and validates random data and packet lengths on the specified communications lines.

The lines you exercise must have a loopback connector attached to the distribution panel or the cable. Also, the line must be disabled in the /etc/inittab file and in a nonmodem line; that is, the CLOCAL option must be set to on. Otherwise, the cmx command repeatedly displays error messages on the terminal screen until its time expires or until you enter Ctrl/c.

You cannot test pseudodevice lines or `lta` device lines. Pseudodevices have `p`, `q`, `r`, `s`, `t`, `u`, `v`, `w`, `x`, `y`, or `z` as the first character after `tty`, for example, `ttyp3`.

The `cmx` command has the following syntax:

/usr/field/cmx [-h] [-o *file*] [-t *min*] [-l *line*]

See `cmx`(8) for a description of the command options.

The following example exercises communication lines `tty22` and `tty34` for 45 minutes in the background:

```
# cmx -l 22 34 -t45 &
```

The following example exercises lines `tty00` through `tty07` until you enter Ctrl/c:

```
# cmx -l 00-07
```

12

Administering the Basic System Event Channels

This chapter explains how system events are logged and describes how to configure the basic system event logging channels. Information on managing log files is included.

The following topics are discussed in this chapter:

- Options for monitoring system events (Section 12.1)
- Setting up event monitoring (Section 12.2)
- How to recover and read event logs after the system has crashed (Section 12.3)
- Options for managing log files (Section 12.4)

12.1 Understanding the Basic Event-Logging Facilities

The operating system uses three mechanisms to log system events:

- The system event-logging facility, `syslogd`. See `syslogd`(8) for information on the initialization options and `syslog.conf`(4) for information on configuration options. See `syslog.auth`(4) for information on remote logging.
- The binary event-logging facility, `binlogd`. See `binlogd`(8) for information on the initialization options and `binlog.conf`(4) for information on the `binlog.conf` file. See `binlog.auth`(4) for information on remote logging.
- The Event Manager provides an integrated approach to administering system events and errors. See `EVM`(5) for an introduction to the Event Manager, and see Chapter 13 for information on configuring and using Event Manager.

You can review events detected and recorded by `syslogd` and `binlogd` using the Event Manager, DECevent, or the error report formatter, `uerf`.

The Event Manager is the recommended method of administering system events. See Chapter 13 for information on configuring Event Manager. The Event Manager viewer, `evmviewer`, provides a graphical user interface for selecting, filtering, and displaying system events. See `EVM`(5) and `evmviewer`(8) for more information.

System events are often returned in a binary format. To render such events in a readable text format, use a translation tool such as:

- The service tools provided with your service contract contain event analysis tools such as Compaq Analyze. See http://www.support.compaq.com/svctools/webes/index.html for more information. Recent processor models produce `binlogd` events using a header format that differs from the format produced by earlier platforms. The newer format events are known as Common Event Header (CEH) events. If your system does not produce CEH events you cannot use Compaq Analyze to translate them, and you must install the DECevent formatter utility, `/usr/sbin/dia`.
- A limited use license for DECevent is provided in the distribution kit as described in the *Installation Guide*. See the *DECevent Translation and Reporting Utility* manual and `dia`(8) for more information.
- The `uerf` command utility. See `uerf`(8) for more information.

Note

The `uerf` command utility does not support CEH events and will be retired in a future release. You should migrate your event management procedures to Event Manager as soon as possible.

The log files created by the event-logging facilities are protected and owned by `root`, and belong to the `adm` group. You must have the proper authority to examine the files.

The following sections describe the event-logging facilities.

12.1.1 System Event Logging

The primary event-logging facility uses the `syslog` function to log system-wide events in ASCII format. The `syslog` function uses the `syslogd` daemon to collect the messages that are logged by the various kernel, command, utility, and application programs. The `syslogd` daemon logs the messages to a local file or forwards the messages to a remote system, as specified in the `/etc/syslog.conf` file.

When you install the operating system, the `/etc/syslog.conf` file is created and specifies the default event-logging configuration. The `/etc/syslog.conf` file specifies the file names that are the destination for the event messages, which are in ASCII format. Section 12.2.1.1 discusses the `/etc/syslog.conf` file. See `syslog.conf`(4) for more information.

The /etc/syslog.auth file specifies which remote hosts are allowed to forward syslog messages to the local host. For system security, only messages coming from remote hosts listed in this file are logged by the syslogd daemon. If the /etc/syslog.auth file is not present, then event forwarding from all remote hosts is enabled.

The /etc/syslog_evm.conf file specifies which syslogd messages are forwarded from the syslogd daemon to the Event Manager, in the form of Event Manager events. Those syslogd messages are posted to the Event Manager daemon, evmd, by syslogd if the syslogd forwarding function is turned on with the -e option. Event forwarding is always on by default. Use the -E option to turn it off if required. Events are posted with the Event Manager name of sys.unix.syslog.*facility*.

See syslog.auth(4) and syslog_evm.conf(4) for more information.

12.1.2 Binary Event Logging

The binary event-logging facility detects hardware and software events in the kernel and logs the detailed information in binary format records. Some events that are logged by the binary event-logging facility are logged by the syslog function also, in a less detailed message.

The binary event-logging facility uses the binlogd daemon to collect various event-log records. The binlogd daemon logs these records to a local file or forwards them to a remote system, as specified in the /etc/binlog.conf default configuration file, which is created when you install your system. Section 12.2.1.3 discusses the /etc/binlog.conf file.

DECevent (or Compaq Analyze) translates binary events to ASCII reports from entries in the system's binary event log files. Invoke DECevent by entering the dia command at the command line. Entering the command without options immediately causes DECevent to access and translate the contents of the event log files, displaying the events as shown in Example 12–1. Events scroll up the terminal screen until all events are displayed or you press Ctrl/c.

Example 12–1: Sample Translated Event

```
**** V3.3  ****************** ENTRY  4
************************   1
Logging OS                              2[OS]    2
System Architecture                     2.
Alpha Event sequence number             440.
Timestamp of occurrence                 22-AUG-2002 18:24:31    3
Host name                               Host Name

System type register      x0000001B     AlphaServer 800 or 1000A
```

Example 12–1: Sample Translated Event (cont.)

```
Number of CPUs (mpnum)    x00000001
CPU logging event (mperr) x00000000

Event validity    4                  1. O/S claims event is valid
Event severity                        5. Low Priority
Entry type                            301. Shutdown ASCII
Message Type                          -1.    - (minor class)
SWI Minor class                       9.
ASCII Message SWI Minor sub class     2. Shutdown  ASCII Message
System halted by root:  System going down @ 6:24PM on 22 Aug
Please log off in good time 5
```

1 The number of the event in the translated log. This number may be based on the selection or filtering of events.

2 Identification of the operating system (*[OS]*) and system architecture.

3 The time stamp (date and system clock time) that indicates when the event occurred and the name of the system on which it occurred (<host name>).

4 Information about the validity, severity, and type of event. In this case, an informational message that the system shut down.

5 The actual message logged by the event, which may have been displayed to a terminal or console also at the time the event occurred.

For information about administering the DECevent utility, see the following documentation:

- *DECevent Translation and Reporting Guide*
- `dia`(8)

Compaq Analyze is a rules-based hardware fault management diagnostic application that provides error event analysis and translation. The multi-event correlation analysis feature of Compaq Analyze provides you with the capability to analyze events stored in the binary system event log or other specified binary log files. When Compaq Analyze is installed, you can launch its GUI interface directly from the SysMan Station by selecting the Host Icon and selecting Compaq Analyze from the Tools menu.

12.2 Configuring Event Logging

You can change the default configuration by modifying the configuration files as described in this section. For example, you can change the configuration so that only important, system-critical events are logged and informational

events are ignored. You can choose to concentrate on certain subsystems, such as mail or print services, and control how and where event messages are logged. The optimum method of monitoring system events is to use Event Manager, as described in Chapter 13. Event Manager enables you to consolidate and filter events.

To enable system and binary event-logging, the special files must exist and the event-logging daemons must be running. See Section 12.2.3 and Section 12.2.4 for more information.

The file `/var/adm/syslog.dated` and other files in `/var/adm` directory are context-dependent symbolic links (CDSLs), which facilitate joining single systems into clusters. The CDSL for the `syslog` directory is `/var/cluster/members/member0/adm/syslog.dated`. Take care not to break symbolic links when working with these files. See Chapter 6 for more information on CDSLs.

12.2.1 Editing the Configuration Files

If you do not want to use the default system or binary event-logging configuration, you can edit the `/etc/syslog.conf` or `/etc/binlog.conf` configuration file to specify how the system should log events. Specify the following data in the files:

- The facility, which is the source of a message or the part of the system that generates a message
- The priority, which is the message's level of severity
- The destination for messages.

The following sections describe how to edit the configuration files.

12.2.1.1 Editing the syslog.conf File

If you want the `syslogd` daemon to use a configuration file other than the default, you must specify the file name in the `syslogd` command, for example:

```
# syslogd -f config_file
```

The following is an example of the default `/etc/syslog.conf` file:

```
#
# syslogd config file
#
# facilities: kern user mail daemon auth syslog lpr binary
# priorities: emerg alert crit err warning notice info debug
#
#[1]  [2]                          [3]
kern.debug              /var/adm/syslog.dated/kern.log
```

```
user.debug                /var/adm/syslog.dated/user.log
daemon.debug              /var/adm/syslog.dated/daemon.log
auth.crit;syslog.debug    /var/adm/syslog.dated/syslog.log
mail,lpr.debug            /var/adm/syslog.dated/misc.log
msgbuf.err                /var/adm/crash.dated/msgbuf.savecore
kern.debug                /var/adm/messages
kern.debug                /dev/console
*.emerg                   *
```

Each `/etc/syslog.conf` file entry has the following entry syntax:

1. Specifies the facility, which is the part of the system that generates the message.
2. Specifies the severity level. The `syslogd` daemon logs all messages of the specified severity level plus all messages of greater severity. For example, if you specify level `err`, all messages of levels `err`, `crit`, `alert`, and `emerg` or `panic` are logged.
3. Specifies the destination where the messages are logged. This can be a log file or a device such as `/dev/console`.

The `syslogd` daemon ignores blank lines and lines that begin with a number sign (#). Specify # as the first character in a line to include comments in the `/etc/syslog.conf` file or to disable an entry.

The facility and severity level are separated from the destination by one or more tab characters or spaces.

You can specify more than one facility and its severity level by separating them with semicolons. In the preceding example, messages from the `auth` facility of `crit` severity level and higher and messages from the `syslog` facility of `debug` severity level and higher are logged to the `/var/adm/syslog.dated/syslog.log` file.

You can specify more than one facility by separating them with commas. In the preceding example, messages from the `mail` and `lpr` facilities of `debug` severity level and higher are logged to the `/var/adm/syslog.dated/misc.log` file.

Facilities

You can specify the following facilities:

`kern` Messages generated by the kernel. These messages cannot be generated by any user process.

`user` Messages generated by user processes. This is the default facility.

`mail`	Messages generated by the mail system.
`daemon`	Messages generated by the system daemons.
`auth`	Messages generated by the authorization system, for example, `login`, `su`, and `getty`.
`lpr`	Messages generated by the line printer spooling system, for example, `lpr`, `lpc`, and `lpd`.
`local0`, `local1`, through `local7`	Reserved for local use.
`mark`	Receives a message of priority `info` every 20 minutes, unless a different interval is specified with the `syslogd -m` command.
`msgbuf`	Kernel syslog message buffer recovered from a system crash. The `savecore` command and the `syslogd` daemon use the `msgbuf` facility to recover system event messages from a crash.
*	Messages generated by all parts of the system.

Severity Levels

You can specify the following severity levels, which are listed in order of highest to lowest severity:

`emerg` or `panic`	A panic condition. These messages are broadcast to all users.
`alert`	A condition, such as a corrupted system database, that you should correct immediately.
`crit`	A critical condition, such as a hard device error.
`err`	An error message.
`warning` or `warn`	A warning message.
`notice`	A condition that is not an error condition, but is handled as a special case.

`info`	An informational message.
`debug`	A message containing information that is used to debug a program.
`none`	A mechanism to disable a specific facility's messages.

Destinations

You can specify the following message destinations:

Full pathname	Appends messages to the specified file. Direct each facility's messages to separate files; for example: `kern.log`, `mail.log`, or `lpr.log`.
Host name preceded by an at sign (@)	Forwards messages to the `syslogd` daemon on the specified host. Messages are not forwarded if the `-R` option is specified when the `syslogd` daemon is started. See Section 12.2.2 for more information.
List of users separated by commas	Writes messages to the specified users if they are logged in.
*	Writes messages to all the users who are logged in.

Daily Log Files

You can specify that the `syslogd` daemon create daily log files by using the following syntax to specify the pathname of the message destination:

```
/var/adm/syslog.dated/{file}
```

The *file* variable specifies the name of the log file, for example, `mail.log` or `kern.log`.

If you specify a `/var/adm/syslog.dated/`*file* pathname destination, each day the `syslogd` daemon creates a subdirectory under the `/var/adm/syslog.dated` directory and a log file in the subdirectory using the following syntax:

```
/var/adm/syslog.dated/date/file
```

- The *date* variable specifies the day, month, and time that the log file was created.

- The *file* variable specifies the name of the log file you specified in the /etc/syslog.conf file.

The syslogd daemon automatically creates a new *date* directory every 24 hours, when you boot the system, or when the syslogd daemon is restarted or reconfigured. You can find the latest logs in the /var/adm/syslog.dated/current directory. The current directory is a symbolic link to the latest *date* directory.

For example, to create a daily log file of all mail messages of level info or higher, edit the /etc/syslog.conf file and include a line similar to the following:

```
mail.info  /var/adm/syslog.dated/mail.log
```

If you specify the previous line in the /etc/syslog.conf, the syslogd daemon creates the following daily directory and file:

```
/var/adm/syslog.dated/11-Jan-12:10/mail.log
```

12.2.1.2 Configuring syslog to Use Event Manager

By default, syslogd is configured with the -e option to forward events to Event Manager. (See Section 12.2.4). You can select which syslog events are forwarded to the Event Manager by modifying the syslog_evm.conf file. If the file does not exist, or if it exists but contains no subscription entries, no syslog messages are posted to the Event Manager.

The default syslog_evm.conf file contains entries similar to those shown in Example 12–2, which excludes the informational file header.

Example 12–2: Sample syslog_evm.conf File Entries

```
1   2
*.emerg
# above forwards all emergency events to EVM  3
kern.info+   4
user.notice+
mail.notice+
daemon.notice+
auth.notice+
syslog.notice+
```

1 The first part of each line item specifies which facility generated the message, such as kern for kernel. An asterisk (*) indicates that all facilities are selected. In this case, *.emerg ensures that all messages of emergency priority are forwarded to Event Manager.

You can choose which events are forwarded by creating an entry for a facility, or removing an existing entry. Entries are based on the keywords in the facility table in Section 12.2.1.1.

2 The second part of each item specifies the priority of messages, based on the keywords in the severity level table in Section 12.2.1.1.

3 You can add comments, preceded by a number sign (#). However, you cannot mix forwarding entries and comments in the same line

4 The plus sign (+) appended to a priority indicates that the specified priority and all higher priority messages are forwarded. If you want to choose individual severity levels for a facility, such as warning, critical and emergency, create a line for each priority.

Events are posted with the Event Manager name of `sys.unix.syslog.`*`facility`*.

See `syslog_evm.conf`(4) and Chapter 13 for more information.

12.2.1.3 Editing the binlog.conf File

If you want the `binlogd` daemon to use a configuration file other than the default, specify the file name with the `binlogd -f`*`config_file`* command. The `binlogd` daemon forwards all events to the Event Manager. You can filter and select `binlog` events using Event Manager utilities, as described in Chapter 13.

You can forward `binlogd` events to a remote host. See `binlogd`(8) for information on the remote logging options. The `-R` and `-r` options are important because they control the creation of an inet port for remote access.

The following is an example of a `/etc/binlog.conf` file:

```
#
# binlogd configuration file
#
# format of a line:   event_code.priority         destination
#
# where:
# event_code - see codes in binlog.h and man page, * = all events
# priority   - severe, high, low, * = all priorities
# destination - local file pathname or remote system hostname
#
#
*.*   /usr/adm/binary.errlog
dumpfile  /usr/adm/crash/binlogdumpfile
102.high  /usr/adm/disk.errlog
 1   2                   3
```

Each entry in the `/etc/binlog.conf` file, except the `dumpfile` event class entry, contains three fields:

1. Specifies the event class code that indicates the part of the system generating the event.
2. Specifies the severity level of the event. Do not specify a severity level if you specify `dumpfile` for an event class.
3. Specifies the destination where the binary event records are logged.

The `binlogd` daemon ignores blank lines and lines that begin with a number sign (#). You can specify # as the first character in a line to include comments in the file or to disable an entry.

The event class and severity level are separated from the destination by one or more tab characters or spaces.

You can specify the following event class codes:

Class Code	Description
General	
*	Specifies all event classes.
`dumpfile`	Specifies the recovery of the kernel binary event log buffer from a crash dump. A severity level cannot be specified.
Hardware-Detected Events	
100	CPU machine checks and exceptions, or generalized exception fault
101	Memory
102	Disk
103	Tape
104	Device controller
105	Adapter
106	Bus
107	Stray interrupt
108	Console event
109	Stack dump
110	Generalized machine state
113	Double error halt
115	(Un)correctable environmental
120	Reporting of correctables disabled
195	StorageWorks Command Console (SWCC)
196	I2O block storage
198	SWXCR RAID controller

Class Code	Description
199	SCSI CAM
Software-Detected Events	
201	CI port-to-port-driver
202	System communications services
203	LSM note
204	LSM warning
205	LSM continuation
206	AdvFS domain panic
Informational ASCII Messages	
250	Generic informational ASCII message
Operational Events	
300	Startup ASCII message
301	Shutdown ASCII message
302	ASCII Panic message
310	Time stamp
350	Diagnostic status ASCII message
351	Repair and maintenance ASCII message
400	Filterlog event. (Use only with filterlog)

Severity Levels

You can specify the following severity levels:

*	All severity levels
`severe`	Unrecoverable events that are usually fatal to system operation
`high`	Recoverable events or unrecoverable events that are not fatal to system operation
`low`	Informational events

Destinations

You can specify the following destinations:

Full pathname	Specifies the file name to which the `binlogd` daemon appends the binary event records.

@hostname	Specifies the name of the host, preceded by an at sign (@), to which the `binlogd` daemon forwards the binary event records. If you specify `dumpfile` for an event class, you cannot forward records to a host. Operational timestamp (310) events are not forwarded automatically.

12.2.2 Remote Messages and syslog Security

Unless the domain host name of a remote host is entered in the local `/etc/syslog.auth` file, the local system does not log any `syslog` messages from that remote host. If you intend to make `syslogd` secure on your system, and you have configured or intend to configure other hosts to forward `syslog` messages to the system, complete the following steps:

1. Use the `su` command to become the superuser (root).
2. Create the `/etc/syslog.auth` file using a text editor.
3. Add the names of any remote hosts that are allowed to forward `syslog` messages to the local system to the `/etc/syslog.auth` file. Host names must meet the following criteria:
   - Each remote host name should appear in a separate line in the `/etc/syslog.auth` file. Lines beginning with the # character are comments and are ignored.
   - A host name must be a complete domain name such as `trout.fin.huk.com`.
   - If a domain host name is given, it must either appear in the local `/etc/hosts` file or the local system must resolve it through a name server (such as BIND).
   - A host name can have at most as many characters as defined by the `MAXHOSTNAMELEN` constant in the `/sys/include/sys/param.h` file, although each line in the `/etc/syslog.auth` file is limited to 512 characters.
   - A plus sign (+) by itself allows event forwarding from all hosts. Also, a host name can be preceded by a minus sign (-) to prohibit that host from forwarding events. If the `/etc/syslog.auth` file is not present on the system, then forwarding from all hosts is enabled.
4. Ensure that the `/etc/syslog.auth` is owned by root.

   ```
   # chown root /etc/syslog.auth
   ```

5. Ensure that the `/etc/syslog.auth` have its permissions set to 0600.

```
# chmod 0600 /etc/syslog.auth
```

Specify the -R option when starting the daemon if you do not want the syslogd daemon to create an inet port to listen for events being sent by remote hosts. To make this the default mode of operation, edit the startup command line in the /sbin/init.d/syslog file. Using the -R option means that the syslogd daemon cannot forward events to other systems.

See syslog.auth(4) and syslogd(8) for more information.

12.2.3 Creating the Special Files

The syslogd daemon cannot log kernel messages unless the /dev/klog character special file exists. If the /dev/klog file does not exist, create it as follows:

```
# /dev/MAKEDEV /dev/klog
```

Also, the binlogd daemon cannot log local system events unless the /dev/kbinlog character special file exists. If the /dev/kbinlog file does not exist, create it as follows:

```
# /dev/MAKEDEV /dev/kbinlog
```

See MAKEDEV(8) for more information.

12.2.4 Starting and Stopping the Event-Logging Daemons

The syslogd and binlogd daemons are started automatically by the init program during system startup. However, you must ensure that the daemons are started. You can specify options when you start the daemons also.

12.2.4.1 The syslogd Daemon

You must ensure that the init program starts syslogd daemon. If the syslogd daemon does not start, or if you want to specify options with the command that starts the syslogd daemon, you must edit the /sbin/init.d/syslog file. When you edit the file, you either must include or modify the syslogd command line. You also can invoke the command manually.

The command that starts the syslogd daemon has the following syntax:

/usr/sbin/syslogd [-b *rcvbufsz*] [-d] [-e | -E] [-f *config_file*] [-m *mark_interval*] [-p *path*] [-r | -R] [-s]

The initialization of the daemon uses only the -e option by default. The -e option configures the daemon to forward events to the Event Manager automatically. You can verify the current syslogd configuration using the ps command as follows:

```
# /sbin/ps agx | grep syslogd
261 ??    S    0:00:10  usr/sbin/syslogd -e
```

See syslogd(8) for information on the command options.

Note

You must ensure that the /var/adm directory is mounted, or the syslogd daemon does not work correctly.

The syslogd daemon reads messages from the following:

- The domain socket /dev/log file, which is created automatically by the syslogd daemon.
- An Internet domain (UDP) socket, which is specified in the /etc/services file. For security reasons, you may want to either disable this socket using the -R option or specify authorized hosts in the /etc/syslog.conf file.
- The device special /dev/klog file, which logs only kernel messages.

Messages from other programs use the openlog, syslog, and closelog calls.

When the syslogd daemon is started, it creates the /var/run/syslog.pid file, where the syslogd daemon stores its process identification number. Use the process identification number to stop the syslogd daemon before you shut down the system.

During normal system operation, the syslogd daemon is called if data is put in the kernel syslog message buffer, located in physical memory. The syslogd daemon reads the /dev/klog file and gets a copy of the kernel syslog message buffer. The syslogd daemon starts at the beginning of the buffer and sequentially processes each message that it finds. Each message is prefixed by facility and priority codes, which are the same as those specified in the /etc/syslog.conf file. The syslogd daemon then sends the messages to the destinations specified in the file.

To stop the syslogd event-logging daemon, use the following command:

```
# kill ‘cat /var/run/syslog.pid‘
```

Using the following command, you can apply changes to the /etc/syslog.conf configuration file without restarting the daemon:

```
# kill -HUP ‘cat /var/run/syslog.pid‘
```

12.2.4.2 The binlogd Daemon

You must ensure that the `init` program starts the `binlogd` daemon. If the `binlogd` daemon does not start, or if you want to specify options with the command that starts the `binlogd` daemon, edit the `/sbin/init.d/binlog` file and either include or modify the `binlogd` command line. You can invoke the command manually also. The `binlogd` command supports the following options

/usr/sbin/syslogd [-d] [-f *config_file*] [-r | -R]

See `binlogd`(8) for information on command options.

The `binlogd` daemon reads binary event records from the following:

- An Internet domain socket (`binlogd`, `706/udp`), which is specified in the `/etc/services` file. For reasons of security, you may want to disable this socket using the `-R` option. You can specify authorized hosts in the `/etc/binlog.conf` file also.
- The `/dev/kbinlog` special file.

When the `binlogd` daemon starts, it creates the `/var/run/binlogd.pid` file, where the `binlogd` daemon stores its process identification number. Use the process identification number to stop or reconfigure the `binlogd` daemon.

During normal system operation, the `binlogd` daemon is called if data is put into the kernel's binary event-log buffer or if data is received on the Internet domain socket. The `binlogd` daemon then reads the data from the `/dev/kbinlog` special file or from the socket. Each record contains an event class code and a severity level code. The `binlogd` daemon processes each binary event record and logs it to the destination specified in the `/etc/binlog.conf` file.

To stop the `binlogd` daemon, use the following command:

```
# kill 'cat /var/run/binlogd.pid'
```

You can apply changes to the `/etc/binlog.conf` configuration file without restarting the daemon by using the following command:

```
# kill -HUP 'cat /var/run/binlogd.pid'
```

12.2.5 Configuring the Kernel Binary Event Logger

To configure the kernel binary event logger, modify the default keywords and rebuild the kernel. You can:

- Scale the size of the kernel binary event-log buffer to fulfill your system needs.

- Enable and disable the binary event logger and the logging of kernel ASCII messages into the binary event log.

The /sys/data/binlog_data.c file defines the binary event-logger configuration. The default configuration specifies a buffer size of 24K bytes, enables binary event logging, and disables the logging of kernel ASCII messages. You can modify the configuration by changing the values of the binlog_bufsize and binlog_status keywords in the file.

The binlog_bufsize keyword specifies the size of the kernel buffer that the binary event logger uses. The size of the buffer can be between 8 kB (8,192 bytes) and 1 MB (1,048,576 bytes). Small system configurations, such as workstations, can use a small buffer. Large server systems that use many disks may need a large buffer.

The binlog_status keyword specifies the behavior of the binary event logger. You can specify the following values for the binlog_status keyword:

0 (zero)	Disables the binary event logger.
BINLOG_ON	Enables the binary event logger.
BINLOG_ASCIION	Enables the logging of kernel ASCII messages into the binary event log if the binary event logger is enabled. This value must be specified with the BINLOG_ON value as follows: `int binlog_status = BINLOG_ON | BINLOG_ASCII;`

You must rebuild and boot the new kernel after you modify the /sys/data/binlog_data.c file.

12.3 Recovering Event Logs After a System Crash

You can recover unprocessed messages and binary event-log records from a system crash when you reboot the system.

The msgbuf.err entry in the /etc/syslog.conf file specifies the destination of the kernel syslog message buffer msgbuf that is recovered from the dump file. The default /etc/syslog.conf file entry for the kernel syslog message buffer file follows:

```
msgbuf.err              /var/adm/crash/msgbuf.savecore
```

The dumpfile entry in the /etc/binlog.conf file specifies the file name destination for the kernel binary event-log buffer that is recovered from the dump file. The default /etc/binlog.conf file entry for the kernel binary event-log buffer file follows:

```
dumpfile              /usr/adm/crash/binlogdumpfile
```

If a crash occurs, the `syslogd` and `binlogd` daemons cannot read the `/dev/klog` and `/dev/kbinlog` special files and process the messages and binary event records. When you reboot the system, the `savecore` command runs and, if a dump file exists, recovers the kernel `syslog` message and binary event-log buffers from the dump file. After `savecore` runs, the `syslogd` and `binlogd` daemons are started.

The `syslogd` daemon reads the `syslog` message buffer file, verifies that its data is valid, and then processes it in the same way that it normally processes data from the `/dev/klog` file, using the information in the `/etc/syslog.conf` file.

The `binlogd` daemon reads the binary event-log buffer file, verifies that its data is valid, and then processes the file in the same way that it processes data from the `/dev/kbinlog` special file, using the information in the `/etc/binlog.conf` file.

After the `syslogd` and `binlogd` daemons are finished with the buffer files, the files are deleted.

12.4 Managing Log Files

On a well maintained system, the size of the various log files should not become a problem when you:

- Select carefully only those events that you want to log
- Monitor the logs for error conditions that result in many postings
- Archive and back up your important event logs regularly

The `/var/spool/cron/crontabs/root` file contains the following model entry for managing log files:

```
0 2 * * 0 /usr/lbin/logclean /var/adm/wtmp > /dev/null
```

You can use the `cron` daemon to specify that other log files be deleted. However, you should take care that important log files are stored or archived according to your local site requirements.

The following is an example of a `crontab` file entry that cleans up the older logs in the `/var/adm/syslog.dated` directory:

```
40 4 * * * find /var/adm/syslog.dated/* -depth -type d -ctime +7 -exec rm -rf {}\;
```

This entry causes all directories under the `/var/adm/syslog.dated` directory (and their contents) that were created more than seven days ago to be deleted every day at 4:40. See Chapter 3 and `crontab`(1) for more information.

12.5 Startup Log Messages in /var/adm/messages

The number of messages stored depends on the size of the message buffer used to store boot-log messages, which is controlled by the `msgbuf_size` kernel attribute. The minimum default value for this attribute is 8K bytes, for systems with up to 128 MB of physical memory. For systems with greater than 128 MB of physical memory, the value of `msgbuf_size` is calculated and set automatically at 64 bytes for every 1MB of memory. For example, in a system with 512 MB, the value is 512 * 64 bytes, or 32,768 bytes , which is equivalent to 32K bytes.

For large systems with many adapters and devices, the default value may be insufficient, causing messages to be dropped from the `/var/adm/messages` file. For large-memory systems that have few devices, the value can be too high and you may want to reclaim the buffer space.

If your system's boot-log record is incomplete, or if you want to reduce the assigned value to reclaim the buffer space, use the following procedure to modify the value of the `msgbuf_size` attribute:

1. Invoke the `dxkerneltuner` graphical user interface from the command line.
2. Select the `generic` subsystem and select `Select Subsystem`. The Subsystem Attributes dialog box labeled generic opens.
3. Locate the `msgbuf_size` and enter the new value for the `Boot Time Value`.
4. Select `OK` to apply the changes and close the dialog box.
5. Select `Reset All` from the `Option` menu in the main window.
6. Select `Exit` from the `File` menu in the main window.

You can use the `sysconfig` and `sysconfigdb` commands to implement this change instead, as described in Chapter 4.

13

Using the Event Manager

The Event Manager is a comprehensive event management system. In addition to providing traditional event handling facilities, it unifies its own events and events from other channels to provide a single source of information, simplifying the task of monitoring system activity. The Event Manager includes a graphical event viewer and a full set of command line tools. It is integrated into the SysMan Menu application suite and the SysMan Station.

The following topics are covered in this chapter:

- An overview of Event Manager (Section 13.1)
- How to set up and customize Event Manager (Section 13.2)
- How to use Event Manager to assist in the administration of your system (Section 13.3)
- Troubleshooting common Event Manager problems (Section 13.4)

13.1 Event Manager Overview

A critical part of a UNIX system administrator's job is to monitor the state of the system, and to be ready to take action when certain unusual conditions occur. Examples of such conditions are when a disk fills or a processor reports hardware errors. It is also important to verify that certain routine tasks run successfully each day, and to review certain system configuration values. Such conditions or task completions are known as system events.

An **event** is an indication that something interesting has occurred – an action has been taken, some condition has been met, or it is time to confirm that an application is still operational. A particular event may be interesting to the administrator or to some other class of system user. A system event could also be significant to other system entities, such as:

- System monitoring software
- Operating system software
- End-user application programs
- Hardware components

Entities interested in events can be part of either the local system or a remote system.

When a system component has something interesting to report, it makes the information available through an **event channel**, which is any facility used to publish or retrieve event information. Examples of event channels are:

- Log files, where messages are stored in a file that is usually in ASCII text format
- Event management systems
- Programs that you run to obtain a snapshot of status information

An event management system is an active event channel and as such, it provides services for distributing, storing, and retrieving event information.

The operating system supports a number of channels through which system components can report event and status information. Verify the information available at each channel regularly to be sure that the system is operating normally. The system logger, `syslog`, and the binary error logger, `binlog`, are familiar examples of event management systems. They provide simple event distribution facilities for other components to use, and their daemons actively manage the event information they receive.

By contrast, the `cron` daemon's log file, `/var/adm/cron/log`, is an example of a passive event channel. The `cron` daemon writes new event information to the end of its file, and takes no special action to notify interested entities when it does so.

Apart from `syslog` and `binlog`, there are several other log files stored in various locations on the system. To facilitate management of these log files, the Event Manager provides a single point of focus for multiple event channels by combining events from all sources into a single event stream. The system administrator can either monitor the combined stream in real time or view historical events retrieved from storage. The Event Manager viewing facilities include a graphical event viewer, which is integrated with the SysMan Menu and SysMan Station, and a full set of command line utilities, which enable you to filter, sort, and format events as needed. You can configure Event Manager to automatically notify you (or other system entities) of selected conditions.

The Event Manager encapsulates `syslog` and `binlog` instead of replacing them. These channels remain in place, and continue to handle the same set of events as they always did. However, with Event Manager the other channels are much more accessible.

13.1.1 Features of the Event Manager

Event Manager provides the following features:

- Facilities for users and applications to post and monitor events

- Support for other event channels, including `syslog` and `binlog`
- Support for encapsuling custom event channels
- Integration with DECevent and Compaq Analyze for translation of binary error log events

  Compaq Analyze is a rules-based hardware fault management diagnostic tool that provides error event analysis and translation
- Integration of a graphical event viewer with the SysMan application suite
- Choice of summary or detailed event data, including online explanations
- A full set of command line utilities that you can use to post and handle events from shell scripts and from the command line
- Configurable event logger that allows full control over which events are logged and optimizes storage space used by identical events
- Configurable event forwarding that enables you to automatically notify other system entities of selected events
- Log file management that automatically archives and purges log files daily
- Support for the application programming interface (API) library
- Centralized access to event information
- Configurable authorization for posting or accessing events

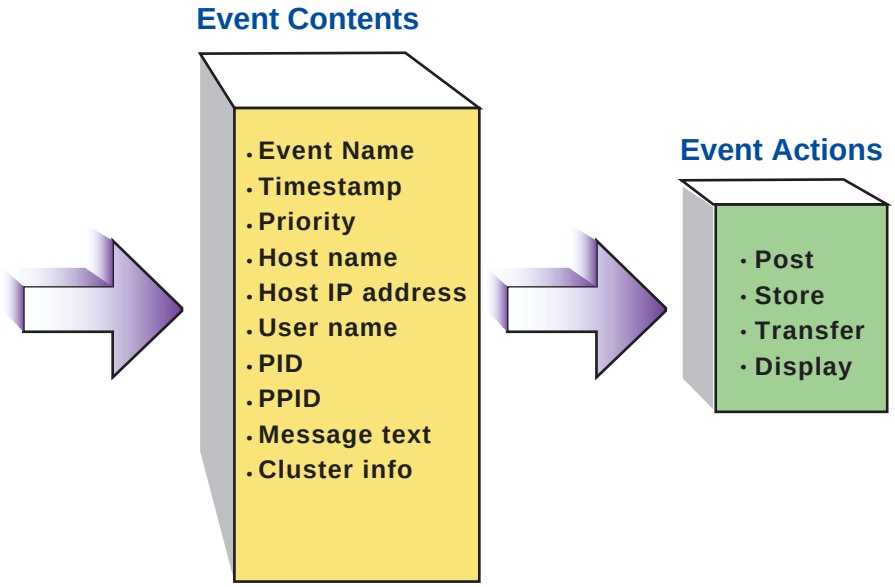
13.1.2 Understanding Event Manager Events

An Event Manager event is a binary package of data that contains a set of standard data items, including a name, a timestamp, and information about the poster. An event may contain variable data, which is named and supplied by the poster. For example, an event reporting the failure of a device may hold variables containing the path name and type of the device. Events are created and posted by an Event Manager posting client, and distributed to other clients by the Event Manager daemon. Then, a receiving process can extract and process the information contained in the event.

Although the Event Manager logger captures posted events and stores them in a system log file, you can easily capture your own set of events and store them in your own file for later analysis. You use the `evmwatch` monitoring utility, or reconfigure the logger to capture your own events.

Figure 13–1 shows a graphical representation of an event. The Event Contents box shows items, such as the process identifier (PID) and the name of the host system on which the event was generated, that may be included in the event. The Event Actions box shows some of the possible actions performed on any event.

Figure 13–1: Event Model



The Event Manager includes command line utilities that understand the format of the event, and which you use to perform basic operations at the command prompt or in shell scripts; you cannot view an event directly with a text viewer (for example, `more`) because an event is a package of binary data. You can use Event Manager commands to:

- Retrieve events from storage, sort them into a preferred order, and format them for display
- Watch for new events being posted
- Post new events

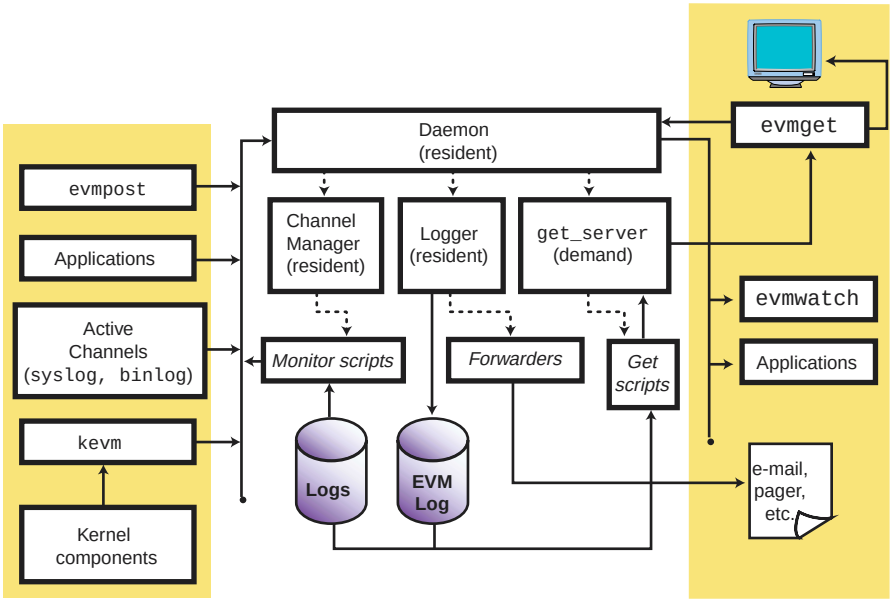
The Event Manager command line utilities are designed to be used together in pipelines. For example, you may pipe a set of events from a file into the sort utility, pipe the output into the formatting utility, then pipe the output of that command into the `more` command, or redirect it to a file. Section 13.3 provides examples of using Event Manager commands to monitor and review event activity.

After the event file is converted to text form, you can use other standard utilities to analyze it. For example, you may display just the event names, and then pipe the display into the `sort -u` and `wc -l` commands to determine how many different types of events are in the file.

13.1.3 Event Manager Components

This section describes how the different parts of Event Manager interact. It also describes the system files used to run Event Manager and any files created by Event Manager during normal operations. Figure 13–2 shows a model of the system.

Figure 13–2: Event Manager Component Model



In Figure 13–2, client components involved in posting events are shown at the left, Event Manager system components are in the center, and client components involved in subscribing to and retrieving of events are at the right. Active event channels post events directly to Event Manager. Passive event channels do not post events and must be polled for information. These channels are depicted by the log files handled by the monitor scripts.

The primary component of the Event Manager is the `evmd` daemon, which is initialized when the system is booted to run level 2; see Chapter 2 for information on run levels. For event management to function during system startup, the initialization of the daemon and its child processes is synchronized as follows:

- When you boot the system, some kernel components post events as part of their initialization sequences. Because the Event Manager daemon is not yet running, these events are queued in kernel memory until the daemon is ready to accept them.

- The Event Manager daemon starts early in the run level 2 initialization sequence of system startup; see Chapter 3 for information on the system run levels. The daemon then:
  - Starts the logger
  - Starts the channel manager
  - Listens for connection requests from clients
- After the logger establishes its listening connection and is ready to log events, the daemon begins accepting posted events from kernel and user-level posters.

The Event Manager logger program, `evmlogger`, runs as a resident process. It is configured to subscribe to a selected set of events, and to store them in managed log files for later retrieval. The logger is also configured by default to:

- Write high-priority events to the system console
- Send mail to the system administrator when high-priority events occur

The Event Manager logger, `evmlogger`, is an essential system component and should never be deconfigured from the system because some system components rely on its operation.

The resident channel manager process, `evmchmgr`, is configured to run periodic channel-monitoring scripts, which post events when they detect noteworthy activity in the channel. The channel manager also runs the daily log cleanup functions.

The get server process, `evmget_srv`, is a transient (demand) process that executes event retrieval scripts for the various event channels. The `evmd` daemon runs an instance of `evmget_srv` whenever a user runs the `evmget` command.

Entities on the left side of the model create posting connections to the daemon in order to post events. After it receives events from the posters, the daemon merges them with corresponding event templates from its template database, and distributes them to its subscribing clients.

The following occur on the right side of the model:

- The `evmwatch` and other application programs that need to receive event information as it happens create subscribing connections to the daemon and pass filter strings to it to specify their event subscriptions.
- The `evmget` command, which a user can run to retrieve historical event information from log files, creates a service connection and passes a filter string to specify the set of events to be retrieved. The daemon then runs an instance of the `get` server to handle the request.

- The e-mail and pager actions are examples of forwarding commands, which the logger may execute in response to the occurrence of certain events.

13.1.3.1 Event Manager Command Line Utilities

Event Manager provides a number of command line utilities both for administering the Event Manager system itself and for use in posting or obtaining events. Table 13–1 describes the general user commands. Detailed information is available from the reference pages. See Section 13.3 for examples of how to use these commands to monitor and review event activity.

Table 13–1: Event Manager Command Line Utilities

Command	Description
`evmget`	Retrieves stored events from a configured set of log files and event channels, using channel-specific retrieval functions
`evmpost`	Accepts a file or stream of text event sources and posts them to the Event Manager daemon for distribution
`evmshow`	Accepts one or more Event Manager events and outputs them in the requested format
`evmsort`	Reads a stream of events and sorts them according to supplied criteria
`evmwatch`	Subscribes to events specified and outputs them as they arrive

Table 13–2 lists the Event Manager administrative commands, which are usually invoked during system initialization. The individual command reference pages discuss other conditions under which the command is used.

Table 13–2: Event Manager Administrative Utilities

Command	Description
`evmchmgr`	The Event Manager daemon automatically starts the Event Manager channel manager. It executes the periodic functions defined for any channel.
`evmd`	The Event Manager daemon receives events from posting clients and distributes them to subscribing clients, that is, clients that have indicated they want to receive the events. The daemon is a critical system facility that starts automatically at system boot. Do not terminate it. The Essential Services Monitor (ESM) daemon, `esmd`, maintains the availability of essential system daemons, including `evmd`, by automatically restarting them. See `esmd`(8) for information on the ESM daemon.

Table 13–2: Event Manager Administrative Utilities (cont.)

Command	Description
`evmlogger`	The Event Manager daemon automatically starts the Event Manager logger. The logger receives events from the daemon and writes them to each of the logs whose filter string they match. The `evmlogger` also serves as an event forwarding agent that you can configure to take an action when required.
`evmreload`	This command posts control events, which instruct the Event Manager components to reload their configuration files. When you modify an Event Manager configuration file you must use this command to load the new configuration.
`evmstart`	This command starts the Event Manager daemon. It is intended for use by the system startup scripts, but you can also use it to restart Event Manager should it terminate for any reason. Under normal operation, the esmd daemon restarts the Event Manager daemon automatically.
`evmstop`	This command stops the Event Manager daemon, preventing entities from posting or subscribing for events. It is intended for use by the system shutdown scripts. Do not use this command under normal circumstances, because Event Manager is required for many system functions to operate correctly. In most circumstances, the esmd daemon restarts the Event Manager daemon automatically.

13.1.3.2 Event Manager Application Programming Interface

The Event Manager API library, `libevm.so`, contains an extensive range of event management functions. This library enables programmers to design programs that interface with the Event Manager. The API functions enable programs to post events, send requests and notifications to the daemon, or receive responses and information from the daemon. The use of these interfaces is described in the *Programmer's Guide*; see `EVM`(5) for a list of individual API reference pages.

13.1.3.3 Event Manager System Files

Event Manager creates or uses the following system files; they are described in terms of executable files, configuration files, and log files.

Executable Files

Executable files for Event Manager administrative commands are located in the `/usr/sbin` directory.

General (that is, user) command executable files are located in the /usr/bin directory.

Initialization files are located in the /sbin/init.d directory.

Configuration Files

Base Event Manager configuration files are located in the /etc directory; they are listed here.

/etc/evmdaemon.conf

This file is a text file that contains commands used to configure and start the Event Manager. See Section 13.2.2.1 and evmdaemon.conf(4) for a complete description of this file.

/etc/evmchannel.conf

The event channel configuration file, which is read by the channel manager, evmchmgr, and the evmshow command. This file describes all the channels through which events can be posted and retrieved. See Section 13.2.2.2 and evmchannel.conf(4) for a complete description of this file.

/etc/evmlogger.conf

The configuration file for the logger, evmlogger. It contains commands used to direct the display, forwarding, or storage of events. See Section 13.2.2.3 and evmlogger.conf(4) for a complete description of this file.

/etc/evm.auth

This file is used to control access to events and event services. See Section 13.2.3.2 and evm.auth(4) for a complete description of this file.

Log Files, Working Files, and Local Installation Files

Log files, working files, and local installation files are located in the following subdirectories of /var/evm.

/var/evm/sockets

This CDSL directory contains a domain socket node, evmd, and a related lock file, evmd.lck. Local clients use this socket for connection.

/var/evm/evmlog

This CDSL directory contains the event logs created by the default Event Manager logger configuration. Log files in this directory have names in the format evmlog.yyyymmdd[_nn], where yyyymmdd

is the date of the log, and _nn is a sequential generation number. A new log generation starts if the log reaches its configured maximum size during the course of the day, or if the logger finds an error in the current file. The day's first log file has no generation number. A new log file is started automatically when it receives the first event after midnight, system time.

This directory also contains a lock file, `evmlog.dated.lck`, and a generation control file, `evmlog.dated.gen`, the latter containing information about the current generation number. See Section 13.2.4 for more information on managing log files.

`/var/evm/adm/logfiles`

This CDSL directory contains output message logs created by the resident components of Event Manager: the daemon, logger, and channel manager. New files are created each time Event Manager starts. Old files are renamed by appending the suffix "`.old`" to their names, overwriting any previous old files. These message logs are encapsulated by Event Manager's `misclog` event channel, so their contents are visible through `evmget` and the event viewer.

`/var/evm/shared`

This directory is a work directory that holds temporary files required for client authentication.

`/var/evm/adm/templates`

The directory is provided for installation of local and third-party event template subdirectories. This directory is connected to the system template directory by a symbolic link.

`/var/evm/adm/channels`

The directory is provided for installation of local and third-party event channel scripts.

`/var/evm/config`

This directory and its subdirectories contain secondary configuration files for various Event Manager components. In this release, only the logger supports secondary configuration files; see `evmlogger.conf`(4) for more information.

`/var/evm/adm/filters`

The directory is provided for installation of local and third-party event filter files.

`/var/run/evmd.pid`

This file contains the daemon process identifier (PID), that is saved by the `evmd` daemon for future actions, such as stopping Event Manager.

`/var/run/evmlogger.info`

This file contains the logger's PID and information about the log files being managed. The `evmlog` channel retrieval and daily cleanup functions use this information.

System-supplied Definition Files

System-supplied definition files for templates, channels, and filters are located in the following subdirectories of the `/usr/share/evm` directory.

Do not modify these files.

`/usr/share/evm/channels`

This directory contains a subdirectory for system-supplied event channels such as `binlog`, `syslog`, and `evmlog`. Each subdirectory contains scripts that define the services available for that channel.

`/usr/share/evm/filters`

This directory contains system filter files.

`/usr/share/evm/templates`

This directory contains system event template files and subdirectories.

13.1.4 Related Utilities

The following subsystems or optional components also provide event handling capabilities:

System logger (`syslogd`)

The system logger logs text messages on behalf of the kernel and many user-level system components. In addition to storing events in its own log files, the default configuration of the `syslogd` daemon forwards selected events to Event Manager for further storage and distribution. Event Manager stores `syslog` events in the `evmlog` files to reduce the overhead of retrieval from potentially very large text files. See `syslogd`(8) for more information.

Binary error logger (`binlogd`)

The binary error logger logs system errors and configuration information in binary format. Events are translated by the DECevent translation facility (`dia`), or by Compaq Analyze (`ca`) depending on the system type. In addition to storing events in its own log files and distributing them to its own clients, the `binlogd` daemon forwards events to Event Manager for further distribution. Event Manager retrieves binary error log events from storage through the `binlog` event channel functions. See `binlogd`(8) for more information.

DECevent and Compaq Analyze

DECevent is a rules-based translation and reporting utility that provides event translation for binary error log events. Event Manager uses DECevent's translation facility, `dia`, to translate binary error log events into human-readable form. Compaq Analyze performs a similar role on most EV6 series processors. See `ca`(8) and other Compaq Analyze documentation for more information.

13.2 Administering Event Manager

The role of the administrator in running Event Manager involves the following principal activities:

- Starting and stopping Event Manager, described in Section 13.2.1
- Configuring Event Manager, described in Section 13.2.2
- Controlling who is allowed to post or access events, described in Section 13.2.3
- Managing log files, described in Section 13.2.4
- Providing event reporting facilities for other system users, described in Section 13.2.5
- Installing new products that use Event Manager capabilities, described in Section 13.2.6

For information on using the Event Manager, see Section 13.3.

13.2.1 Starting and Stopping Event Manager

The Event Manager is started automatically at system startup and is stopped when the system is shut down.

The Essential Services Monitor (ESM) daemon, esmd, maintains the availability of essential system daemons, including the Event Manager

daemons, by automatically restarting them. See esmd(8) for more information.

To stop Event Manager, it is necessary to acquire the process identifier of the ESM daemon. Use the following procedure to stop the Event Manager:

1. Acquire the process identifier (PID) of the ESM daemon.

```
# ps -aef | grep esmd | grep -v grep
1. root    48  1  0.0  Apr 22 ??  0:00.09 /usr/sbin/esmd
```

   In this example the PID is 48.

2. Use the kill command to stop the ESM daemon.

```
# kill -STOP PID
```

3. Use the evmstop command to stop the Event Manager.

```
# /usr/sbin/evmstop
```

Use the following procedure to start the Event Manager and ESM daemon.

1. Use the evmstart command to start the Event Manager.

```
# /usr/sbin/evmstart
```

2. Use the kill command to restore the operation of the ESM daemon; be sure to use the same *PID* that you used to stop ESM daemon.

```
# kill -CONT PID
```

You do not need to stop and start Event Manager when you want to change the Event Manager configuration. In this instance, change the configuration, then issue the evmreload command. See evmreload(8) for more information.

13.2.2 Configuring Event Manager

Configuring Event Manager means establishing and maintaining its configurable resident components:

- The Event Manager daemon, evmd
- The channel manager, evmchmgr
- The logger, evmlogger

Each component recognizes a configuration file that directs its operations.

When you install the operating system, Event Manager is configured to run with default configuration options that are suitable for most installations automatically. However, you can change the configuration for your system if, for example:

- An event channel is to be added or modified
- The log file archive and expiration options need to be changed

- An alternate logging directory is established
- Remote access to Event Manager facilities is to be enabled

Event Manager is preconfigured to use both DECevent and Compaq Analyze to translate binary logger (`binlogd`) events.

Whenever the configuration changes because a new file is loaded or because a change is made, the configuration must be reestablished by running the `evmreload` command. See `evmreload`(8) for information on this command.

Configuration files are described in the following sections and in the corresponding reference pages.

13.2.2.1 Event Manager Daemon Configuration

The Event Manager daemon reads the `/etc/evmdaemon.conf` configuration file at system startup and whenever you issue a reload request by using the `evmreload` command. For a complete description of the contents and syntax of the configuration file, see `evmdaemon.conf`(4). Example 13–1 shows some sample entries in the Event Manager daemon configuration file.

Example 13–1: Sample Event Manager Daemon Configuration File Entries

```
# Event template directory:
sourcedir "/usr/share/evm/templates"    1

# Start the Event Manager Logger      2
start_sync "/usr/sbin/evmlogger -o /var/run/evmlogger.info \
            -l /var/evm/adm/logfiles/evmlogger.log"

# Start the Event Manager Channel Manager     2
start_sync "/usr/sbin/evmchmgr -l \/var/evm/adm/logfiles/evmchmgr.log"

# Event retrieval service definition:
service      3
 {
   name          event_get
   command       "/usr/sbin/evmget_srv"
 }

# Set up an activity monitor.
activity_monitor      4
  {
   name            event_count
   period          10
   threshold       500
   holdoff         240
  }
```

Example 13–1: Sample Event Manager Daemon Configuration File Entries (cont.)

```
remote_connection false        5
```

1 This statement identifies the top of the directory hierarchy for all event template files.

2 These commands start the `evmlogger` and the `evmchmgr` components as synchronized clients, ensuring that both clients complete their subscription requests before the daemon accepts any events from posting clients. The command line options for these commands define the clients' log files and, in the case of the logger, an output file that is used to make operational details available to the `evmlog` event channel functions.

3 These statements define the `event_get` event retrieval service, which the `evmget` command uses to retrieve events.

4 These statements define an activity monitor. In this example, if 500 or more events are received during any ten minute period, the daemon posts a high-priority event to alert the system administrator. Activity monitoring (counting of events) is then suspended for the hold-off period of four hours (240 minutes).

5 This line sets the `remote_connection` to `false` to disable connection to this system by remote Event Manager clients. See `evmdaemon.conf`(4) and to Section 13.2.3 for information about the security implications of changing this value.

If you make any changes to the configuration file you must run the `evmreload` command to make the Event Manager daemon aware of these changes. See `evmreload`(8) for more information.

13.2.2.2 Event Manager Channel Configuration

An event channel is a source of event information. The channel configuration file, `/etc/evmchannel.conf`, defines a set of event channels and the functions that operate on them, for use by the channel manager, the `evmshow` command, and the event retrieval process. For a complete description of the contents and syntax of the channel configuration file, see `evmchannel.conf`(4). Example 13–2 shows sample channel configuration file entries.

Example 13–2: Sample Event Manager Channel Configuration File

```
# Global path for channel functions
path   /usr/share/evm/channels     1

# Time-of-day at which daily cleanup function will run
cleanup_time 02:00:00   2

# ================================
# Event channel:  EVM log
# ================================
channel
  {   3
    name        evmlog   4
    path        /usr/share/evm/channels/evmlog      5
    events      *      6
    fn_get      "evmlog_get"  7
    fn_details  "evmlog_details"
    fn_explain  "evmlog_explain"
    fn_monitor  "evmlog_mon"
    fn_cleanup  "evmlog_cleanup 7 31"       8
    mon_period  15:00  # Monitor every 15 minutes      9
   }
```

1 This line declares the `/usr/share/evm/channels` directory as the default path for all channel functions. This path is prefixed to the names of any channel functions defined in this file that do not begin with a slash (`/`) character, unless the channel group supplies its own path value.

2 This line defines a daily 2:00 am cleanup for all channels.

3 This line specifies a configuration group that defines an event channel.

4 This line specifies that the name of the channel is `evmlog`.

5 This line overrides the default path `/usr/share/evm/channels` defined at the global level.

6 In this line, the asterisk (`*`) indicates that the channel provides default event handling, meaning that its functions are invoked to provide details and explanations for any events whose names do not match the events value of any other channel.

7 Any line beginning with `fn_` defines a script that runs for each function.

8 The argument values on this line are passed to the cleanup program to control its operation. In this example, log files older than 7 days are compressed and those older than 31 days are deleted. The meanings of

the arguments are specific to individual channel functions, and may not be the same in all cases.

9 This line sets the monitoring period, which causes the /usr/share/evm/channels/evmlog/evmlog_mon function to be invoked every 15 minutes.

13.2.2.3 Event Manager Logger Configuration

The Event Manager logger handles storage and forwarding of events, according to entries in the /etc/evmlogger.conf configuration file. For a complete description of the contents and syntax of this file, see evmlogger.conf(4). Example 13–3 shows sample entries in a logger configuration file. An example of possible customization of the logger is to direct output to a terminal in addition to a log file.

Example 13–3: Sample Event Manager Logger Configuration File Entries

```
# Main log file:
eventlog {   1
    name        evmlog   2
    logfile     /var/evm/evmlog/evmlog.dated   3
    type        binary   4
    maxsize     512  # Kbytes    5

    # Uncomment the following "alternate" line and set the
    # logfile path to specify an alternate logfile in case
    # of write failures.
    # The path must specify an existing directory.
    #alternate  /your_alternate_fs/evmlog/evmlog.dated   6

    # Log all events with priority >= 200, except binlog events:
    filter       "[prio >= 200] & (! [name @SYS_VP@.binlog])"  7

    # Suppress logging of duplicate events:
    suppress    8
    {   filter      "[name *]"
        period      30    # minutes
        threshold   3     # No. of duplicates before suppression
    }
}
# Forward details of high-priority events to root:
forward {    9
    name      priority_alert    10
    maxqueue  200   11

    # Don't forward mail events through mail
    filter  "[prio >= 600] & ![name @SYS_VP@.syslog.mail]"   12

    suppress    13
```

Example 13–3: Sample Event Manager Logger Configuration File Entries (cont.)

```
    {   filter  "[name *]"
        period  120    # minutes
        threshold   1  # No. of duplicates before suppression
    }

    # This evmshow command writes a subject line as the first
    # line of output, followed by a detailed display of the
    # contents of the event.
    # The resulting message is distributed by mail(1).
    command "evmshow -d -t 'Subject: EVM ALERT [@priority]: @@' |
    mail root"   14

    # Limit the number of events that can be queued for this
    # command:
    maxqueue        100
}
# Secondary configuration files can be placed in the following
# directory.  See the evmlogger.conf(5) reference page for
# information about secondary configuration files.
configdir       /var/evm/adm/config/logger
```

1 This line begins an event log configuration group.

2 This line provides a `name` for the the event log. Other portions of the configuration file may reference this name.

3 This line specifies that the log files are stored in the `/var/evm/evmlog` directory. Each day, when the log for that day is first written, the `dated` suffix is replaced by the date in the format `yyyymmdd`.

4 This line specifies that the `type` of events written to this log are binary Event Manager events, rather than formatted (ASCII text) events.

5 This line specifies the maximum size of the log file in kilobytes (KB). In this case, if the size of the current log file exceeds 512 KB the logger closes it and begins a new log file, with a sequentially numbered suffix (for example, _2) appended to the file name.

6 If this line is not commented out (by #) and the sample path is replaced by the path name of an existing write-enabled directory, an alternate log file is opened in this directory if the primary directory becomes write-disabled.

7 This line establishes the filtering conditions for events, determining which events are logged by this event log. See `EvmFilter`(5) for details

of Event Manager filter syntax. The @SYS_VP@ entry is a macro that is replaced with sys.unix when the file is read.

8 These statements define the suppression parameters for this event log. In this case, suppression of a particular event begins if three or more duplicate events are received within 30 minutes. Suppression of duplicate events saves space in the log file. See evmlogger.conf(4) for a detailed description of event suppression.

9 This line establishes conditions for forwarding events to the root user. An event forwarder executes a specified command string when selected events occur. It is useful for notifying the system administrator when a significant error occurs.

10 In this line, name identifies the forwarder.

11 The maxqueue *queue_limit* keyword limits the number of events that a forwarder can queue while a previous event is being handled. If the maximum number of events is already queued when a new event arrives, the new event is ignored by this forwarder. If not specified, this keyword has a default value of 100 events. If you specify a value greater than 1000 events, the logger automatically limits it to 1000 events.

12 This line establishes filtering for the events. As with an event log definition, the filter string specifies the set of events that are handled by this forwarder. To prevent an event loop from occurring if the mailer posts high-priority events, signifying a possible problem in the mail subsystem, mail events are explicitly excluded from this forwarder.

13 These lines suppress multiple forwarding of events. The suppression mechanism for a forwarder is similar to that for an event log. Here, the purpose is to prevent the command from being sent multiple times in a short period because of the same event being posted repeatedly. In the example, a particular event is forwarded once every two hours at most.

14 This line defines the command that executes when an event is handled by the forwarder. The event is piped into the command's stdin stream. The result of this command is shown in the comments preceding the command line.

If you make any changes to the logger configuration file you must run the evmreload command to make the changes known to the logger; see evmreload(8) for more information.

See Section 13.3.13.3 for details of remote logging configuration.

13.2.2.4 Secondary Logger Configuration Files

Secondary logger configuration files enable you to add event logs or forwarders without modifying the primary configuration file,

`/etc/evmlogger.conf`. This feature ensures that any problems with secondary files do not affect the primary configuration. It enables you to safely experiment with different logger configurations. Should the logger encounter a syntax error in a secondary configuration file, it displays an error message and rejects the file. The primary configuration file and any additional (and correct) secondary files are processed and Event Manager functions correctly. The secondary configuration directory feature also allows individual system components, products and applications to install or change logfiles and forwarders by installing or replacing files, rather than having to insert or maintain lines in the primary configuration file. You can uninstall entries by removing the file.

The default and recommended location of secondary configuration files is the `/var/evm/adm/config/logger` directory, or a subdirectory of that directory. You can place the configuration file elsewhere and create a symbolic link to it from the default directory. Although supported, it is recommended that you avoid adding `configdir` lines to the primary configuration file. Your secondary configuration files must have file name suffix `.conf` and the file syntax must follow the rules stated in Section 13.2.2.3.

It is important that you give appropriate permissions to the secondary logger configuration files and directories. The logger runs with superuser privileges and can execute commands specified in any secondary configuration file. For this reason, the logger rejects any configuration files that do not have the correct permissions and posts a warning event. See `evmlogger.conf`(4) for the correct file permissions.

In a cluster environment, the logger configuration files are shared by all the cluster members. If you require a member-specific event log or forwarder, you can specify it in a secondary configuration file. Create a context-dependent symbolic link (CDSL) in the secondary configuration directory to reference the file. See `mkcdsl`(8) for instructions on creating a CDSL.

13.2.2.5 Changing the Buffer Size to Prevent Missed Events

If missing events becomes a problem, then you can increase the receive buffer size by changing a system parameter.

The receive buffer size is set to the default system socket buffer maximum. Enter the following command to determine the current size of this parameter:

```
# /sbin/sysconfig -q socket sb_max
```

To change the runtime value of this parameter, enter the following command:

```
# /sbin/sysconfig -r socket sb_max=new-value
```

This change remains in effect until the next reboot and affects only new Event Manager connections.

Use the `sysconfigdb` or `dxkerneltuner` utilities to effect the change on a permanent basis. See `sysconfigdb`(8) or `dxkerneltuner`(8) respectively for more information.

13.2.3 Security Considerations

Security is an important consideration when dealing with events, for the following reasons:

- Uncontrolled access to certain event information may provide an unauthorized user with sensitive information about system operation.
- Posting certain events may cause critical system actions, for example, application failover or system shut down, to occur.

Traditionally, event information security is maintained by restricting read access to log files and limiting certain posting operations to the root user. Because the Event Manager daemon and event retrieval facilities provide alternate means of access to all events, both as they are posted and after they are logged, the daemons also provide a way to limit access, so that events are seen only by authorized users. You can enable access control by providing authorization facilities and using authentication techniques.

You must be careful to avoid compromising security when writing executable functions to be used in the Event Manager environment. See the *Programmer's Guide* manual for more information about protecting channel functions.

As described in Section 13.2.3.3, the Event Manager can be accessible remotely to specified users.

13.2.3.1 User Authentication

The Event Manager daemon authenticates the identities of all local system users before accepting any connection request. In a cluster, users requesting a connection from another node of the same cluster are also authenticated. See Section 13.2.3.3 for information about remote connections, including authentication of remote users.

13.2.3.2 User Authorization

Access to events is controlled by the Event Manager authorization file, `/etc/evm.auth`.

The root user can authorize individual users or groups of users to do the following:

- Post selected events
- Access (subscribe to or retrieve from storage) selected events

- Execute selected services

By default, all events are protected. Event rights are granted by supplying, for each event class, a list of users who have the specified right or who are explicitly denied rights. A plus sign (+) that is not followed by a user list implicitly grants the right to all users. A minus sign (-) that is not followed by a user list implicitly denies the right to all users. The root user has implicit posting and access rights to all events unless explicitly denied them. Example 13–4 shows sample entries in an authorization file. See `evm.auth`(4) for more information.

Example 13–4: Sample Event Manager Authorization File Entries

```
# ===================
#      EVENTS
# ===================

event_rights {   1
   class        @SYS_VP@.evm.control   # EVM control events
   post         root
   access       +
}

event_rights {   2
   class        @SYS_VP@.evm.msg.admin  # EVM admin message
   post         root
   access       "root, group=adm"
}

event_rights {   3
   class        @SYS_VP@.evm.msg.user   # EVM user message
   post         +
   access       +
}

# ===================
#      SERVICES
# ===================

service_rights {   4
   service      event_get
   execute      +
}
```

1 Only the root user can post the class of events that have names beginning with `sys.unix.evm.control`. Such events are accessible by all users. The @SYS_VP@ entry is a macro that is replaced with `sys.unix` when the file is read.

2 Only the root user can post the class of events that have names beginning with `sys.unix.evm.msg.admin`. Such events can be accessed by root or other users in the `admin` group.

3 All users can post or access the class of events that have names beginning with `sys.unix.evm.msg.user`.

4 All users can execute the `event_get` service.

If you make any changes to the authorization file you must run the `evmreload` command to make the Event Manager daemon aware of the changes.

13.2.3.3 Remote Access with Authentication

Event Manager can be accessible to clients that are running on remote systems, allowing you to monitor and retrieve events from a central system. This access can be universal, or limited to specific systems. Also, individual users on remote systems can be allowed, or given or denied permission, to subscribe to events, post events, and retrieve events.

You can make a remote connection by specifying a host name or IP address by using the `-h` option with the `evmwatch`, `evmget` and `evmpost` command line utilities. Alternatively, you can specify a remote host name in the event viewer's Get Events From... dialog box. See Section 13.3.11 for information on the event viewer.

There are two files that control remote access:

`/etc/evmdaemon.conf`

If the value for `remote_connection` is `true`, remote access is allowed. If this value is `false` (the default value), remote access is denied.

`/etc/evm.auth`

The `remote_host` settings in this file control which remote systems are allowed access, which users on those remote systems are allowed or denied access, and the authentication method, `callback` or `open`, to be used.

Setting Remote Hosts, Users, and Authentication

After setting the value of `remote_connection` in the `/etc/evmdaemon.conf` file to `true`, you may need to refine access by remote host systems and users. The `remote_host` settings in the `/etc/evm.auth` file allow you to do so.

The `remote_host` settings have the following format.

```
remote_host  {
             host           "list of one or more host names"
             authentication authentication-type
             port           port-number
             users          "user specification"
             }
```

These parameters may appear in any order, but the host specification is usually first. These parameters are described further.

`host`

The specification for the host parameter is a list of one or more hosts that are permitted access to the Event Manager on the local host. You can give the host name or IP address for the host in this list; however, the host may not be a cluster alias. The list of hosts can be separated by space characters or commas if the list is enclosed in double-quotes. Otherwise, use commas to separate the host names. The following are examples of acceptable host lists:

```
host "hostA hostB hostC"
host "hostM, hostN"
host hostX,hostY,hostZ
```

`authentication`

This argument specifies the type of authentication to be used when accessing the hosts listed. There are two types: `evm_open` and `evm_callback`. Specifying `evm_open` allows open access to the remote host. Specifying `evm_callback` means that authentication is performed and the remote host is contacted to verify the user's identity.

Both authentication types may be specified. In this case, the local host determines if the remote host can use `evm_callback`. If so, `evm_callback` is used; otherwise, `evm_open` is used. This is illustrated in an example at the end of this section.

`port`

The port number specifies the TCP/IP port.

`users`

This argument determines which users on a remote system can access the local host and which local authorizations they have. There are several variations of this specification. The simplest form is the individual login of the remote user, as follows:

```
users    adam
```

This means that remote user adam has the same authorization on the local host as the local user adam.

You can use the equals sign (=) to map a remote user to a local user. Thus, the remote user remo has the same authorization as local user lcal.

```
users    remo=lcal
```

The hyphen or minus sign (-) before the remote user name indicates that this user is denied access.

```
users    -eve
```

The plus sign (+) means all users. Here, it means that all remote users have the same authorization as the local login nobody.

```
users    +=nobody
```

In another form, the plus sign maps every remote user to their local equivalent:

```
users    +=+
```

These user specifications may be used in combination. For example, the following specification means that root has root authorization, and everyone else, except cain, has the same authorization as the local user nobody; the remote user cain is denied access.

```
users   root
users   cain-
users   +=nobody
```

Examples

In the following example, systems systemA, systemB, and systemC are allowed remote access to this machine using callback authentication. Every user on these remote systems maps to user1 on the local system.

```
remote_host {
            host  "systemA, systemB,systemC"
            users  +=user1
            authentication  evm_callback
            }
```

In this example, systems systemA, systemB, and systemC are allowed remote access to this machine using callback authentication. Every user on these remote systems, except for user2 and user3, maps to user1 on the local system; user2 and user3 are denied access explicitly.

```
remote_host {
            host  "systemA, systemB,systemC"
            users  +=user1
            users  -user2
            users  -user3
            authentication  evm_callback
            }
```

In this example, systems systemA, systemB, and systemC are also allowed remote access to this machine using callback authentication. Every user on

these remote systems is mapped to his or her equivalent on the host system, except user2 who is denied access.

```
remote_host {
            host  "systemA, systemB,systemC"
            users  +=+
            users  -user2
            authentication  evm_callback
            }
```

All systems are allowed remote access with callback authentication in the following example. While every other remote user maps to the local "special_user", root is denied access.

```
remote_host {
            host +
            users  +=special_user
            users  -root
            authentication  evm_callback
            }
```

In this example, systemA and systemB are allowed remote access using open authentication. All users are able to access the Event Manager.

```
remote_host {
            host  "systemA, systemB"
            authentication evm_open
            }
```

In the following example, five remote systems may access the Event Manager on the local host, but the extent of access differs. Only the root and adm users on the remote systems julius, augustus, and caesar have access to the Event Manager; when any of these systems attempts to establish a connection for those users, the local Event Manager determines whether they can use `evm_callback` authentication. If so, that is used; otherwise, `evm_open` is used. All users on the remote systems plato and socrates map to themselves, and only the `evm_callback` authentication is used.

```
remote_host {
        host  "julius augustus caesar"
        users  root
        users  adm
        authentication  evm_callback
        authentication  evm_open
        }

remote_host {
        host  plato,socrates
        users  +=+
        authentication  evm_callback
        }
```

13.2.4 Managing Log Files

The Event Manager channel manager, `evmchmgr`, provides log management capability through the channel `fn_cleanup` function. You can define

this capability for any channel through the channel configuration file, `evmchannel.conf`. See Section 13.2.2.2 for more information on this file.

By default, channel cleanup functions run when Event Manager starts, and then run at 2:00 am each day. You can change the time of day by editing the `cleanup_time` value in the channel configuration file. When a cleanup is scheduled, the channel manager scans the event channel list, and executes the `fn_cleanup` command for each channel identified in the file.

The `evmlog` cleanup function, `evmlog_cleanup`, takes two arguments:

- The archive period, which has a default value of 7 days.
- The delete period, which has a default value of 31 days.

The function uses the `find` utility to locate and compress (zip) all logs older than the archive period, and to delete any archived files older than the delete period. You can change the period values by editing the function definition in the channel configuration file. Setting either of these values to zero disables the corresponding function.

The default channel configuration also provides a similar cleanup function for the SysMan Station message log files, through the `misclog` event channel. You can manage the `syslog` and `binary` error log channels by using entries in the `crontab` file. The binary error log file typically is not managed on a daily basis; the channel's cleanup function posts a daily Event Manager event reporting the size of the log. If the log is growing significantly, review the log entries; if necessary, use the cleanup options in `binlogd` to initiate a cleanup. See `binlogd`(8) for more information.

The `evmget` command does not retrieve `evmlog` events that are stored in archived (zipped) logs. To retrieve events from archived logs you must first uncompress them with the `gunzip` command; see `gunzip`(1) for information on unzipping archive files.

13.2.5 Event Templates

An event template is a centrally held description of an event. The template is used:

- To register the event with the Event Manager daemon, so that the event is posted
- To hold centralized information, avoiding the need to have it hard-coded into an application

Event template definitions are held in template files, which are text files stored in directories subordinate to (or linked to) the system template

directory, /usr/share/evm/templates. If you have installation-specific or third-party event templates, load them as follows:

1. Create an appropriately-named subdirectory of the local template directory, /var/evm/adm/templates, and copy the event templates into it.
2. Run the evmreload command, specifying the -d option to signal the Event Manager daemon to reload its internal template database.

To be recognized by Event Manager, template files require specific ownership and permissions. See evmtemplate(4) for more information. See the *Programmer's Guide* for more information on installing new event template files.

Each time an event is posted, the Event Manager daemon looks in its internal template database for a template event whose name matches the posted event. It then retrieves any centralized data items held in the template event, and combines them with the items the program supplied when it posted the event, to yield a merged event for distribution to subscribers.

13.2.6 Installing New Event Manager Clients

You can add new events to the event set as new applications are installed and as new administrative scripts are developed to use the facilities. As events are added it may be necessary to modify Event Manager configuration and authorization files, and to add new templates. See Section 13.2.2 for a discussion of the various configuration files. See Section 13.2.3.2 for information on changing the authorization for new users.

Add new event templates as follows:

1. Create new template files as described in Section 13.2.5.
2. Copy the template files to the /var/evm/adm/templates directory or to a subdirectory.
3. Run the evmreload command, specifying the -d option, to signal the Event Manager daemon to reload its internal template database.

See evmtemplate(4) for details of the required ownership and permissions of a template file.

See the *Programmer's Guide* for more information about developing Event Manager client applications.

13.2.7 Configuring binlog Event Translation Utilities

There are two utilities that provide event translation, Compaq Analyze and DECevent. Newer processors do not support DECevent; they support only Compaq Analyze.

Compaq Analyze is a rules-based hardware fault management diagnostic tool that provides error event analysis and translation. The multi-event correlation analysis feature of Compaq Analyze provides the capability to analyze events that are stored in the system's event log file and to analyze events from other systems, including other operating systems such as OpenVMS and Windows NT.

DECevent is a rules-based translation and reporting utility that provides event translation for binary error log events. Event Manager uses DECevent's translation facility, `dia`, to translate binary error log events into human-readable form.

Although the Event Manager infrastructure directly recognizes events only in its own Event Manager format, events are posted through other channels, such as the `binlogd` daemon. These events can be passed to Event Manager within a wrapper Event Manager event by inserting the lower-level event into the Event Manager event as variable data. The whole package is then passed to Event Manager without Event Manager having any knowledge of the content or format of the variable.

The binary logger daemon, `binlogd`, uses this approach to make its own events available through Event Manager. When the `binlogd` daemon receives an event from the operating system it first stores the event in its own log file and distributes it to its own clients. It then creates an Event Manager event whose name begins with `sys.unix.binlog`, and adds a variable called `binlog_event`, which contains the `binlogd` event data. Finally, it posts the package to the Event Manager daemon for further distribution. The Event Manager daemon deals with the package as it would any Event Manager event, and has no direct knowledge of the contents of the `binlog_event` variable.

When you request a detailed view of an event, either by running the `evmshow -d` command from the command line or by selecting `Details...` in the event viewer's event summary window, Event Manager runs the detailed display program defined for the event in the `/etc/evmchannel.conf` file. The resulting display always begins with an explanation of the event and a detailed view of its contents. If the event is a `binlogd` event, this display is followed by a translation of the contents of the `binlog_event` variable. This translation is useful for troubleshooting a system problem. Example 13–5 shows a detailed display of a `binlogd` event, including a DECevent translation.

Example 13–5: A binlogd Event Showing the DECevent Translation

```
=================== Binary Error Log event ====================
Event Manager event name: sys.unix.binlog.op.shutdown
 Binary error log events are posted through the binlogd
 daemon, and stored in the binary error log file,
 /var/adm/binary.errlog. This event is posted by the shutdown(8),
 halt(8), and reboot(8) commands when the system is being shut
 down. The message includes details of the user who initiated
 the shutdown.
================================================================
Formatted Message:
    System shutdown msg:  System rebooted by root:
Event Data Items:
    Event Name        : sys.unix.binlog.op.shutdown
    Priority          : 200
    Timestamp         : 26-Jan-2000 20:54:36
    Host IP address   : 16.69.224.11
    Host Name         : kopper
    Format            : System shutdown msg: $message
    Reference         : cat:evmexp.cat:300
Variable Items:
    subid_class = 301
    message = "System rebooted by root:"
    binlog_event = [OPAQUE VALUE: 96 bytes]
============================ Translation =====================
DECevent version: V3.2

Logging OS                       2. operating system
System Architecture              2. Alpha
Event sequence number          752.
Timestamp of occurrence              26-JAN-2000 20:54:36
Host name                            kopper
System type register      x0000000F  AlphaStation 600 or 500
Number of CPUs (mpnum)    x00000001
CPU logging event (mperr) x00000000
Event validity                   1. O/S claims event is valid
Event severity                   5. Low Priority
Entry type                     301. Shutdown ASCII Message Type
SWI Minor class                  9. ASCII Message
SWI Minor sub class              2. Shutdown
ASCII Message                    System rebooted by root:
================================================================
```

Event Manager obtains the `binlogd` event translation by passing the event to either DECevent or Compaq Analyze. If neither of these programs is available, or if the translation attempt fails, the translation area of the display shows a message indicating the failure.

Several factors govern the type of binlogd event translation that is available on any given system:

- DECevent is available for older-generation Alpha processor platforms, including some early EV6 platforms. Compaq Analyze must be used to translate events for newer EV6 platforms.
- If DECevent is to be used for translation, the DECevent event formatter utility, /usr/sbin/dia, must be installed on the local system. If the utility is not installed on your system, you need to install it from the Associated Products CD-ROM. Consult your installation documentation for more information. If your system is supported by Compaq Analyze you do not need to install DECevent.
- Unlike DECevent, Compaq Analyze uses a client/server model and it is not necessary to install it on every system that it uses. If your site has licensed Compaq Analyze to run on only a small number of systems, those systems can still provide translation services for other systems. If you need to use a remote Compaq Analyze server to do translations, you must edit the local channel configuration file, as described below.
- Newer processors produce binlogd events with a new header format that differs from the format produced by earlier platforms. The newer format events are known as Common Event Header (CEH) events. If your system does not produce CEH events you cannot use Compaq Analyze to translate them, and you must install the DECevent formatter utility, /usr/sbin/dia.

If your system uses DECevent or uses a Compaq Analyze server running on the local system for binlogd event translation, you do not need to change the standard configuration. If you plan to use a Compaq Analyze server running on a remote system, you need to edit the /etc/evmchannel.conf file. In a default installation, the fn_details line for the binlog event channel is configured as follows:

```
fn_details    "binlog_details -decevent -ca localhost"
```

This line instructs Event Manager to use DECevent to provide translations if it is available; otherwise Event Manager attempts to connect to a Compaq Analyze server running on the local host. If neither of these options is successful, Event Manager attempts to run Compaq Analyze in standalone mode and, if this fails, no translation is done. It is advisable to leave these options in place as the first two items in the list, but if you have other systems running the Compaq Analyze server you can choose to append further -ca items.

In the following example, Event Manager tries in turn DECevent, Compaq Analyze on the local system, Compaq Analyze on the remote system gandalf, and finally Compaq Analyze on the remote system tigger. (This

line is broken at the backslash (\) to fit the page, and appears as a single line in the file).

```
fn_details  "binlog_details -decevent -ca localhost -ca gandalf \
-ca tigger"
```

After you edit the configuration file, run the `evmreload -c` command to make the Event Manager channel manager aware that the file is updated.

Event Manager does not start the Compaq Analyze server; it must be running on the selected system already for the translation to succeed. The server usually starts automatically when the system is initialized. For more information, see the Compaq Analyze documentation.

See Section 13.4 for procedures that enable you to determine whether either translation utility is available on your system.

13.3 Using Event Manager in System Administration

The ability of Event Manager to monitor multiple event sources and combine them into a single event stream makes it a very useful means of monitoring system activity. By default, the logger is configured to send mail to the root user when events with a priority of 600 (alert) or greater are posted. You should review the full event log on a daily basis by using the event viewer or command line utilities. You can configure the logger to take other actions, such as sending a pager message according to any criteria you choose. You can monitor events at your terminal as they occur by using the `evmwatch` command.

The following sections illustrate the commands you can use to monitor and review event activity. As you become familiar with the Event Manager command set, you build up a set of favorite commands, shell scripts, and filters that help you to keep track of what is happening on your system.

13.3.1 Displaying Events Using evmshow

Because an Event Manager event is a binary data package, it must be converted to text before you can display it on a terminal. The `evmshow` command reads binary Event Manager events from its `stdin` stream or from a named file, and outputs the same events in text form to `stdout`. For example, you may display the contents of a file containing Event Manager events by using the following command:

```
# cat my_events | evmshow | more
```

This command displays the events from the log file in the default manner, that is, it takes the format data item from each event, expands it with the values of any variables it references, and displays it. References to variables are identified by a dollar sign ($). Therefore, if the `my_events` file contains

an event with a format data item of `AdvFS: AdvFS domain` *`$domain`* `is full`, and the event also contains a variable named *`domain`* with a value of `root_domain`, the corresponding line of the output is:

```
AdvFS: AdvFS domain root_domain is full
```

This information tells you what happened, but not when it happened, or the importance of the event. You can modify the output of the `evmshow` command to include any data items in the event, including its timestamp and priority, by using the `-t` option to specify a show-template. A show-template is a text string that indicates which data items you want to be displayed for an event, and how you want them to be displayed.

The following example illustrates the use of a show-template to display an event with a timestamp, a priority, and the formatted event message. In the show-template, the names of the items to be displayed are each preceded by an at sign (@) . Two at signs (@@) indicate that the event's format item should be expanded and displayed. The second line shows the output for the domain full event. In the output, the event priority is surrounded by brackets, and there are two spaces before the message text, exactly as specified in the show-template:

```
# cat my_events | evmshow -t "@timestamp [@priority]  @@" | more
22-Jun-2000 11:22:27 [600] AdvFS: AdvFS domain root_domain is full
```

You can set up your own show-template to display the items that are important to you, in any format you want. See `EvmEvent`(5) for a list of all the data items. After you determine your preferred style you can set a default show-template in the environment variable `EVM_SHOW_TEMPLATE` and use fewer keystrokes at the command line. The following Korn shell (`ksh`) commands are equivalent to those in the previous example:

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority]  @@"
# cat my_events | evmshow | more
```

If you want more information about an event, you can request a detailed display, including an explanation and a full dump of its contents, by using the `evmshow` command with the `-d` option. The following example shows a detailed display of the AdvFS domain full event:

```
# cat my_events | evmshow -d | more
============================ EVM Log event =======================
EVM event name: sys.unix.fs.advfs.fdmn.full

    This event is posted by the AdvFS filesystem to provide
    notification that the specified AdvFS domain is full.  No more
    space is available for writing.      1
=================================================================

Formatted Message:
    AdvFS: AdvFS domain root_domain is full      2

Event Data Items:     3
```

```
    Event Name        : sys.unix.fs.advfs.fdmn.full
    Cluster Event     : True
    Priority          : 600
    PID               : 1177
    PPID              : 724
    Timestamp         : 22-Jun-2000 11:22:27
    Host IP address   : 0.0.0.0
    Host Name         : x.x.example.com
    User Name         : root
    Format            : AdvFS: AdvFS domain $domain is full     4
    Reference         : cat:evmexp.cat:450

Variable Items:      5
    domain (STRING) = "root_domain"

=============================================================================
```

1 The explanation of the event. In some cases, this data field contains a recommended action to rectify a problem.

2 The Formatted Message section.

3 The Event Data Items section, which lists all the standard data items contained in the event. See EvmEvent(5) for a description of each of these items.

The items shown here are typical of many events, but sometimes some of these are missing, and occasionally you may see additional items. For example, most events are not distributed across all nodes of a cluster, and so in most cases the Cluster Event item is not displayed.

4 The Format data item is almost the same as the content of the Formatted Message data item, but it includes a reference to a variable called *domain*, indicated by the $ symbol preceding it.

5 The Variable Items section, which contains the value of the domain variable.

See Section 13.3.12.2 for information on how to select events for detailed display.

You can use the evmshow -x command to display the explanation alone. Alternatively, use the -x and -t options together to provide a summary of the event followed immediately by its explanation. For example:

```
 #cat my_events | evmshow -x -t "@timestamp
[@priority]  @@" | more \
21-Jun-2002 11:22:27 [600] AdvFS: AdvFS domain root_domain is full
 This event is posted by the AdvFS filesystem to provide
 notification that the specified AdvFS domain is full.
 No more space is available for writing.
```

The examples in this section show how to display Event Manager events that are contained in a single log file. You can display events that are stored in the various system log files, or monitor them as they occur by using the

evmget and evmwatch commands, which are introduced in Section 13.3.3 and Section 13.3.6.

Most systems produce a large number of events, many of which report normal operation. Use event filters to limit the display to a set of events that you consider interesting. Section 13.3.2 introduces the Event Manager filtering facilities.

Regardless where the events come from, you use the evmshow command to format them for display. See evmshow(1) for more details of the show-template.

13.3.2 Introducing Event Filters

This section introduces event filters and relates them to the evmshow command examples from the previous section. Filtering is used more extensively in later sections, which describe event retrieval and monitoring techniques. The full filter syntax is defined in EvmFilter(5).

An Event Manager event filter is a text string that tells Event Manager which events you want to retrieve. For example, the filter string [priority >= 600] selects events that have a priority of 600 or higher. A filter can be very simple, but the filter language is powerful, and with some practice you can easily build and store a filter expression that defines precisely the set of events that you want to monitor. Filters are used by several of the Event Manager command line utilities, by the Event Manager logger, and by system daemons and client applications.

The evmshow, evmget and evmwatch commands support the -f option which you use to specify a filter string. You can select the events to be displayed from the my_events file, as shown in the following example:

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority]  @@"
# cat my_events | evmshow -f "[priority >= 600]" | more
```

(The preceding example was introduced in Section 13.3.1.) In this example, the -f option specifies the filter, and selects events that have a priority of 600 or higher. The command reads all events from the file, but returns only those events that match the filter string.

If you know the names of the events you want to retrieve, you can specify them in a filter, as shown in the following example:

```
# cat my_events | evmshow -f "[name sys.unix.fs.advfs.fdmn.full]" | more
```

You can use wildcard characters in place of name components as follows:

- An asterisk (*) character matches zero or more complete components
- A question mark (?) matches exactly one complete component

For example, use the following command to shorten the preceding example command:

```
# cat my_events | evmshow -f '[name *.advfs.fdmn.full]' | more
```

The wildcard asterisk matches the components `sys.unix.fs`. To avoid any possibility that the shell expand the wildcard character with filenames, enclose the filter string in single quotes instead of the double quotes. This is always a wise precaution when special characters are used in shell commands.

When you filter by name, Event Manager assumes that there is a wildcard `.*` at the end of the name string, even if it is not included in the command. Therefore, you may receive events with more name components than you specify. The following two commands are equivalent to each other, but the final wildcard (`.*`) in the first command is unnecessary:

```
# cat my_events | evmshow -f '[name *.advfs.*]'
# cat my_events | evmshow -f '[name *.advfs]'
```

You can find the names of events by specifying `@name` as one of the items in your show-template when you run the `evmshow` command.

Use the filter syntax to combine multiple conditions into a single filter with the `AND`, `OR` and `NOT` keywords, and you can use parentheses to group conditions. The following example command selects all events whose names include the component `advfs`, and that have a priority of 600 or higher:

```
# cat my_events | evmshow -f '[name *.advfs] and [priority >= 600]'
```

The following command also selects events with the name component `binlog`, regardless of their priority. In the following example, the keyword `priority` is abbreviated to `pri`, and `name` is abbreviated to `na`. Most filter keywords can be abbreviated as described in `EvmFilter`(5).

```
# cat my_events | evmshow -f '([na *.advfs] and [pri >= 600]) or [na *.binlog]'
```

The examples in this section illustrate the most commonly used filter keywords. When you are familiar with applying filters to the `evmshow` command and the Event Manager commands described in the following sections, you can use the more advanced filter features to create and save useful filters, and to increase your ability to select the events that are most interesting. Advanced filter techniques are described in Section 13.3.12, and the full syntax is given in `EvmFilter`(5).

13.3.3 Retrieving Stored Events Using evmget

System log files store events in many different formats and with different levels of detail, making it difficult to produce an ordered view of all events by using traditional system utilities. You can use the `evmget` command to produce an ordered view by retrieving events from each of the various log files, converting them to Event Manager events if they are not already in

that form, and returning a single stream of Event Manager events. Using the evmshow command, then you can turn the Event Manager event stream into a display format.

The following command pipeline uses the evmget command to retrieve all system events, and passes them to the evmshow command for display:

```
# evmget | evmshow -t "@timestamp [@priority]  @@" | more
```

The evmget command makes a service connection to the Event Manager daemon, which starts a new copy of the get-server program, /usr/sbin/evm_getsrv. The get-server program reads the channel configuration file, and runs the get function, usually a shell script, for each channel configured in the channel configuration file, /etc/evmchannel.conf. This configuration file is described in Section 13.2.2.2.

The get function does the following:

- Reads the channel's log file
- Converts the events into EVM format
- Feeds events back to the evmget command which writes them to its stdout stream

After all the channel get functions run and all the events are returned, the get-server daemon and the evmget command both terminate.

Note

Even though events may be stored in log files as lines of text, or in a special binary format, the evmget command returns all events in the form of binary Event Manager events, which can be passed to evmshow for display. If you send the output of evmget directly to your terminal, the command displays an error message because the binary output cannot be displayed properly and could affect the settings of your terminal. If you pipe the output into another command, such as more or pg, the evmget command is unable to detect the error, and random characters are displayed.

Like the evmshow command, the evmget command supports a filter option to allow you to limit the events it returns. For example, the following command displays only high-priority events:

```
# evmget -f '[pri >= 600]' | evmshow | more
```

It is more efficient to specify a filter with the evmget command than with the evmshow command. This is because the evmget command passes its filter string to the event channel's get function, which only returns events

that match the filter. Fewer events are passed back through the get-server daemon to the evmget command, and the commands operate faster because they transfer and process fewer events.

If you want to save retrieved events for later analysis, or to copy them to another system, you can redirect the output of the evmget command into a file. For example:

```
# evmget -f '[pri >= 600]' > my_events
```

Saving the binary output of the evmget command provides greater flexibility than saving the text output of the evmshow command. At a later time you can sort and filter the binary file and pass it to the evmshow command to view it in any format you like.

When you experiment with the evmget command , the events appear in batches, probably with all the binary error logger events appearing first. Within each batch, the events are likely to be ordered chronologically. This is because the binlog event channel is specified first in the default channel configuration file, so its get function runs first. Each get function feeds its events back to the evmget command in turn, and the evmget command outputs them in the order in which it receives them. Because you usually want to see events in some order (often, but not always, chronological order) you need to pipe the events through the evmsort command, which is described in Section 13.3.4. Section 13.3.5 introduces using the evmget command with the -A option, which makes it possible to retrieve, sort, and display events without building a pipeline.

Depending on the size and type of your system and the number of events being logged, event retrieval may take a noticeably long time. This is because each retrieval operation requires every channel's get function to read through its log files, convert its events to Event Manager events, and then apply the filter string (if any) to determine whether the event is passed back to the evmget command. The larger the log files, the longer this process takes. Careful log file management helps to speed up the process. If you know that you want to display events that belong to a particular event channel, you can shorten the process by using the evmget -C command to display only the specified channel. For example:

```
# evmget -f '[pri >= 600]' -C binlog | evmshow | more
```

In this example, the get function runs only on the binlog channel, so the command completes its task quickly. A filter string is specified to return events that have a priority greater than 600. You can determine what channels are configured by using the evminfo -lc command, or by examining the channel configuration file. See evminfo(1) for more information.

13.3.4 Sorting Events Using evmsort

The evmsort command takes a stream of Event Manager events as input, sorts them into the requested order, and writes them to its stdout stream. The command is most useful in sorting the output from the evmget command, but it can be used to sort Event Manager events from any source. See evmsort(1) for more information.

Section 13.3.3 explained that the events retrieved by the evmget command are output in batches, corresponding to the event channel configuration. You can use the evmsort command to sort the events into a preferred order, before passing them to the evmshow command for display. The following example shows a typical command sequence:

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority]  @@"
# evmget -f '[pri >= 600]' | evmsort | evmshow | more
```

By default, the evmsort command sorts events into chronological order, so the previous command is suitable for most cases. You can use the -s option to declare a sort specification if you want the events sorted differently. A sort specification is a text string that defines one or more sort keys, which are the data items on which you want to sort the events. The specification is a list of data item names, separated by colons (:). For example:

```
priority:timestamp
```

The preceding specification sorts events by timestamp within priority, so the first group of events that are returned are those with the lowest priority, sorted in their order of occurrence. You may use this specification as follows:

```
# evmget -f '[pri >= 600]' | evmsort -s "priority:timestamp" | evmshow | more
```

The default sort order is ascending, but you can change it to descending for an individual item specifier by appending a minus sign (-). You can explicitly request ascending order by specifying a plus sign (+). For example, the following command displays the highest priority events first (descending order), but within each priority range the events are sorted oldest first (ascending order):

```
# evmget -f '[pri >= 600]' | evmsort -s "priority-:timestamp+" | evmshow | more
```

For consistency with the show-template syntax, the evmsort command allows you to precede each item specifier with an at (@) character, as described in Section 13.3.1. There is no requirement to do this, and it does not affect the operation.

When you establish your sorting preferences, you can create a new default sort sequence by setting the environment variable EVM_SORT_SPEC. The following Korn shell (ksh) commands are equivalent to the previous example:

```
# export EVM_SORT_SPEC="priority-:timestamp+"
# evmget -f '[pri >= 600]' | evmsort | evmshow | more
```

You can override the value of the EVM_SORT_SPEC variable at any time by supplying a different sort specification with the -s option.

13.3.5 Using the -A Option to Simplify the Command String

The Event Manager commands are designed to be building blocks, with each command doing one specific operation. This gives you great flexibility in developing shell scripts to manipulate event information. When you enter commands from the command line you may prefer to simplify the command.

The most common command sequence for event retrieval is the evmget command, piped into the evmsort command, piped into the evmshow command. You can then pipe the text output into the more command to display the output. Consider the following example:

```
# evmget -f '[pri >= 600]' | evmsort -s "priority-:timestamp+" |
evmshow | more
```

You can simplify the preceding command by using the evmget -A command option, which automatically pipes the command output to other Event Manager commands. For example, you can use the -A option to simplify the previous command example as follows:

```
# evmget -A -f '[pri >= 600]' -s "priority-:timestamp+" | more
```

When the evmget -A command starts, it automatically runs the evmsort -A command, and pipes its output into that command. When the evmsort command starts, the -A option causes it to start the evmshow command, piping events into it for display. You can supply a sort specification with the -s option and a show-template with the -t option. These options are passed along to the evmsort command and evmget commands respectively.

The evmwatch command supports the -A described in Section 13.3.6.

13.3.6 Monitoring Events Using evmwatch

You can use the evmwatch command to monitor event activity through a terminal window. This command is an Event Manager subscribing client. It makes a connection to the Event Manager daemon, sends it a subscription request, and waits to receive events. As events arrive, the evmwatch command writes them to the standard out stream (stdout) as binary Event Manager events.

You cannot display the output of the evmwatch command because it is a stream of binary events. You must use the evmshow command to format the events. The following example monitors all events, and displays them on your terminal as they occur:

```
evmwatch | evmshow -t "@timestamp [@priority]  @@"
```

Depending on your system type, and the level of event activity, this command may run for a while before any events are displayed. The command continues to run until you terminate it to regain control of your terminal, usually by pressing Ctrl/c.

When a system is operating correctly, many of the events posted are low-priority informational events. You may want to filter these events out, particularly if your system has a high level of event activity. You can do this by supplying a filter to the evmwatch command:

```
# evmwatch -f "[priority >= 400]" |
evmshow -t "@timestamp [@priority]  @@"
```

This example watches for events with a priority of error or higher. You can change the filter string to exclude any set of events that occur regularly and are uninteresting. Alternatively, you may need to watch for a particular set of events.

The preceding examples do not show the output of evmshow piped into more for display, because evmwatch is a realtime monitor. The evmwatch command displays events as they occur, rather than displaying them from a file. A command like pg or more may wait for the operator to intervene before reading more data from its input pipe; over time, this could lead to congestion in the pipeline. The Event Manager daemon cannot wait for its client (the evmwatch command) to clear its backlog; this results in the evmwatch command missing events. You should display the output from the evmwatch command directly on a terminal window, instead of using of piping commands to more or pg; also use the scrollbar to review the event list.

Avoid piping the output of the evmwatch command into the evmsort command because the evmsort command cannot sort events until it reads to the end of its input. As a monitoring program, the evmwatch command usually waits for input until it is killed explicitly. As a result, if you pipe the output of the evmwatch command directly into the evmsort command, there is no output from the evmsort command.

The -A option simplifies the command string by running the evmsort command and the evmshow command automatically. The evmwatch command also supports the -A option and automatically runs the evmshow command when you use it. You can specify a show-template as an option to the evmwatch command as follows:

```
# evmwatch -A -f "[priority >= 400]" -t \"@timestamp \
[@priority]  @@"
```

As with the evmget command, you can capture a set of interesting events in a file, to review later. It is more useful to store events in binary form than in text form, so you should send the output of the evmwatch command

directly to a file, as shown in the following example, rather than piping it into the evmshow command first.

```
# evmwatch -f "[priority >= 400]" > my_events
```

The evmwatch command supports additional options that are useful for monitoring events from within a shell script. See evmwatch(1) for more information.

13.3.7 Posting Quick Message Events Using evmpost

Although most events are likely to be posted by system and application software, there may be times when you want to post an event from the command line or from a shell script. For example, you may want to post a message event in the system log to note that a task is complete, or that you noticed something interesting. Making an entry in the system log makes it easy to establish when other events occurred relative to your entry.

You can post an event by using the evmpost command. The simplest form of this command is the quick message form, which you can specify by using the -a (administrator) or -u (user) option. To post a message, you supply the message on the command line as a quoted string:

```
# evmpost -a "Fire drill started - evacuating computer room"
```

Administrative quick messages are posted with the name sys.unix.evm.msg.admin, so you can search for them with a name filter:

```
# evmget -f '[name *.msg.admin]' |
evmshow -t 'timestamp [@priority]  @@'
27-Jun-2000 15:40:49 [200]  EVM admin msg: Fire drill
started - evacuating computer room
```

By default, the message is posted as a notice event, with a priority of 200. You can change the priority with the -p option. For example, setting the priority to 400 categorizes the message as an error event:

```
# evmpost -p 400 -a  \
"Users reporting possible network problems"
```

By default, only the root user or members of the adm group can post events with the -a option, although you can make it available to other privileged users by editing the authorization file, /etc/evm.auth, as described in Section 13.2.3.2. Any user can specify the -u option to post messages in the same way. If necessary you can restrict this privilege to trusted users by editing the authorization file.

13.3.8 Listing Registered Events

You register events by adding template file entries as described in Section 13.2.5, and running the `evmreload` command with the `-d` option to make them known to the Event Manager daemon, or restarting the system.

You can use the `evmwatch -i` command to retrieve a list of registered events. Pipe the output from the `evmwatch -i` command to the `evmshow` command to display the event templates in any desired format. For example:

```
# evmwatch -i | evmshow -t "@name [@priority] @format" -x
```

Templates are returned as binary Event Manager events which you can either redirect into a file or pipe to the `evmshow` command for display. In the preceding example, the show-template (`-t` option) displays the name of the event, the priority, and the message format. The `-x` option causes each summary line to be followed by an explanation of the event.

Because you are displaying templates (not real system events) you specify a command sequence that requests only the event's message format, not an expanded message. In the output, the summary lines display the messages with names of variables rather than their values. For example you may see the following summary line and explanatory text:

```
sys.unix.fs.advfs.fdmn.bal.error [400] AdvFS: Balance error on AdvFS domain $domain
  This event is posted by the balance(8) command to indicate that an
  error has occurred while balancing the domain.

  Action: Please see balance(8) for further information.
```

In this example, the *$domain* variable is replaced by the domain name when you use the `evmget` command to retrieve a posted instance of the event.

If you do not want to see all registered events, use a filter to limit the output of the `evmwatch` command to the events in which you are interested:

```
# evmwatch -i -f '[name *.evm]' | evmshow -t "@name \
[@priority] @format" -x
```

13.3.9 Posting Events from a Shell Script

Use the `evmpost` command to post a newly registered event, by passing event information to the command in source (text) format. A full description of the event syntax is provided in `evmpost`(1). Source-level posting is most useful in a shell script that performs a routine operation, where the event may indicate success or failure of the operation. This section describes a procedure to create and post a new event that informs you when a backup is finished. The basic steps are:

1. Create a template file and verify its syntax.

2. Install the template file and make it known to the Event Manager daemon.
3. Update the authorization file to allow the events to be posted.
4. Write shell script commands to post the event.

The *Programmer's Guide* gives event design guidelines. You should be familiar with the concepts described in that book before you begin designing a new event. In this example, the backup script posts one of two events, `local.admin.backup.ok` with a priority of 200 (notice) and `local.admin.backup.failed`, with a priority of 400 (error). The failure event includes a variable item named *result_code*, to hold the exit code returned by the backup program. The variable is an 8-bit unsigned integer, and in the template it has a dummy value of zero. This dummy value is replaced with an actual value when the event is posted. The template file syntax is described in the `evmtemplate`(4) .

The following procedure describes how to create and post a new event:

1. Create the `/var/evm/adm/templates/local` directory if it does not exist.
2. Use a text editor, such as `vi`, to create the following text file:

```
# This file contains EVM event templates for local
# backup notification events.
event {
  name local.admin.backup.ok
  format "BACKUP: Backup completed OK"
  priority 200
}

event {
  name local.admin.backup.failed
  format "BACKUP: Backup failed - code $result_code"
  var {name result_code type UINT8 value 0}
  priority 400
}
```

3. Save the file in the `/var/evm/adm/templates/local` directory with the name `backup.evt`.

   You can install new template files in any directory under `/var/evm/adm/templates`, but name subdirectories and template files according to the names of your events for ease of identification. Keeping a small number of closely-related event templates in a single template file simplifies maintenance.
4. Verify the template syntax. The syntax of a template file is identical to the syntax used to post an event, so you can use the `evmpost -r` command to verify the syntax. The `-r` option instructs the `evmpost`

command not to post the event, but to validate the syntax, convert the input into binary Event Manager events, and then write the Event Manager events to its standard output (stdout) stream. Use the evmpost -M command option to prevent the merging of template items into the event, or to add any environmental items such as a timestamp or host name.

As with any stream of binary Event Manager events, you can use the evmshow command to verify the output of the evmpost command. To do this, enter the following command:

```
# cat /var/evm/adm/templates/local/backup.evt |
 evmpost -r -M | evmshow -t "@priority @@"
```

If you created the file correctly, the following output is displayed:

```
200 BACKUP: Backup completed OK
400 BACKUP: Backup failed - code 0
```

5. Verify that the file is owned by root or bin, and that its permissions are set to 0400, 0600, 0440 or 0640. Correct the permissions by using the chown command and the chmod command if necessary.

6. Run the following command to instruct the Event Manager daemon to reload its configuration:

   ```
   # evmreload -d
   ```

   If the command displays an error message, correct the problem and reenter the command. The most likely problem is that the ownership or permissions of the file are incorrect.

7. Verify template registration by using the evmwatch -i command option, which retrieves templates from the Event Manager daemon's database. The evmwatch command outputs the templates in the form of binary Event Manager events; you can use the evmshow command to display them. You need to show only the names of the events to be sure that they are registered correctly, as shown in the following example:

   ```
   # evmwatch -i -f "[name local.admin.backup]" |
    evmshow -t "@name"
   local.admin.backup.ok
   local.admin.backup.failed
   ```

8. Update the authorization file, /etc/evm.auth, to allow the events to be posted. Add the following lines to ensure that only the root user can post the events and any user can see the events:

   ```
   # Local backup events:
   event_rights {
      class        local.admin.backup
      post         root
      access       +
   }
   ```

Only the first three components of the name are specified. These components are common to the two new events, and when either of the events is posted its name matches this entry,

9. Run the `evmreload -d` command option, so that the daemon recognizes the new authorizations.

10. Verify that the events were logged correctly by using the following commands:

```
# echo 'event {name local.admin.backup.ok}' | evmpost
# echo 'event {name local.admin.backup.failed}' | evmpost
# evmget -f '[name local.admin.backup]' |
 evmshow -t '@timestamp [@priority]  @@'

28-Jun-2002 15:21:39 [200]  BACKUP: Backup completed OK
28-Jun-2002 15:21:40 [400]  BACKUP: Backup failed - code 0
```

In the preceding example, the `evmpost` command reads the source input from its standard input (`stdin`) stream, converts it to an Event Manager event, and posts it. The output from the final command shows the posted events. It includes the priorities specified in the template file because the Event Manager daemon merges the template information into each event as it is posted. The value of the code in the second event is zero, because that is the dummy value supplied in the template file, and that value was not overridden in the posted event. In the backup script the value is set to something other than zero.

11. Add the posting commands to your backup script, as shown in the following example:

```
#! /bin/sh
# This shell script runs the backup operation
# and posts an event to indicate success
#or failure.

do_backups  # Performs the backup operation
if [ $? -eq 0 ]
then
 # success
 echo 'event {name local.admin.backup.ok}'| evmpost
else
 # failure
 RES=$?
 evmpost << END
 event {
   name local.admin.backup.failed
   var { name result_code type UINT8 value $RES }
    }
END
fi
```

In the preceding example, the input to the evmpost command for the success event is simple, so it is supplied on the same line by using the echo command. For the failure event, the value of the result_code variable must be supplied also. To supply this value, the shell's << syntax provides a more structured multiline form of input. Both forms of input supply source code input to the evmget command through its standard input (stdin) stream.

See evmpost(1) for more information about posting events from the command line, or from within a shell script.

13.3.10 Understanding the Event Manager Mark Event

When you review or monitor event activity, you observe the following event that occurs every 15 minutes:

```
26-Jun-2000 08:57:45 [200] EVM: Mark event
```

The evmlog event channel posts this event to ensure that there is periodic event activity. If your system has a problem and you need to determine when it was last operational, you can look for mark commands in the system log by using the following command:

```
# evmget -f "[name *.evm.mark]" | evmshow -t "@timestamp @last_timestamp  @@"
26-Jun-2000 00:57:35 26-Jun-2000 04:42:40  [16 times] EVM: Mark event
26-Jun-2000 04:57:41 -  EVM: Mark event
26-Jun-2000 05:12:41 -  EVM: Mark event
26-Jun-2000 05:27:41 -  EVM: Mark event
26-Jun-2000 05:42:41 26-Jun-2000 09:12:45  [15 times] EVM: Mark event
```

If the default logger configuration file is in use, you usually see three individual mark events, followed by a single event preceded by [n times], where n is a number up to 16. This is the result of the logger's suppression facility, which minimizes wasted space by combining multiple events over a period of up to four hours. The normal timestamp value shows the first occurrence of a combined event, and the last_timestamp data item shows the time of the last occurrence. The example includes the last_timestamp data item in the show-template, which displays the last mark event, posted at 09:12:45. This mark event tells you that the system was operational at that time.

To disable mark event posting, edit the channel configuration file to make either of the following changes:

- Comment out the evmlog channel's fn_monitor entry to disable it completely
- Change the mon_period value for the channel to change the frequency with which the event is posted

See Section 13.2.2.2 and `evmchannel.conf`(4) for details of the channel configuration file. See Section 13.2.2.3 and `evmlogger.conf`(4) for more information about event suppression.

13.3.11 Viewing Events Using the SysMan Event Viewer

The SysMan graphical event viewer provides a simple and convenient interface to the system event logs. The event viewer is an integral part of the SysMan system management suite; you can use it in a variety of graphical domains, including an X Windows display or a character cell terminal, as a PC application, or from a Web browser. You can launch the viewer from the SysMan Station also. See Chapter 1 for information about using SysMan.

To launch the event viewer from the command line, enter the `sysman` command, then open the Monitoring and Tuning menu branch. Select the View Events option to start the event viewer. To launch the event viewer directly from CDE, open the tool drawer on the CDE front panel and select in turn System_Admin, DailyAdmin, and Event Viewer.

When you run the event viewer for the first time a warning message may indicate that events are filtered to show only high priority events. If your system is operating normally it is likely that no events are displayed in the event summary window. To choose the events you want to see, select `Filter...` at the bottom of the window, and change the filter criteria in the Filter window. If you want to see all stored events, make sure that all the check boxes at the left side of the window are in the unchecked state, and select `OK`. If your system produces a high level of event activity you can reduce the number of events shown, and the time taken to display them, by checking the Priority box and adjusting the priority range. Setting the range to 400-700 displays all events with a priority of `error` and higher. Setting the low end of the range to 300 includes warning events in the display.

You can check any of the buttons at the left of the Filter window to include additional criteria in the display filter. Each time you make a change you must select `Apply` to apply the change to the event list, or select `OK` to apply the change and return to the main viewer window.

The Filter dialog window offers an intuitive and convenient way for you to build an event filter string without having to type it. If you are familiar with the filter syntax and you want to make better use of its power, you can enter a filter string through the Advanced Filter dialog box, which you access by selecting `Options...` at the bottom of the main event window. You can also save a filter string and reuse it later. For more information about the filter syntax, see `EvmFilter`(5).

One of the most important features of the viewer is the ease with which you can display a detailed view of any event. Simply select the event in

the summary window and select `Details...` to see all the information available, including explanation text and, in the case of a `binlog` event, the translation from DECevent or Compaq Analyze. From the Event Details window you can browse through the event list without returning to the main window.

You can change the viewer display, including the source of events, by selecting `Customize...` and `Options...`. To change the order in which events are displayed, select `Sort...`. Select `Help...` from any window for detailed information about the viewer and its facilities.

Note

The event viewer does not monitor event activity in real time. To display an updated view of the event list, select `Refresh` from the main window.

See `sysman`(8) and `evmviewer`(8) for more information on using these applications. See the online help associated with the event viewer for information on using the viewer options.

13.3.12 Advanced Selection and Filtering Techniques

The following section describes some additional filtering techniques that you can use to further improve event selection, so that you receive only the events in which you are interested.

- How to filter events according to their time of posting (Section 13.3.12.1)
- How to filter using the `event-id` identifier (Section 13.3.12.2)
- How to filter using reserved component names (Section 13.3.12.3)
- How to use filter files (Section 13.3.12.4)

13.3.12.1 Filtering By Time

You can filter for events according to the time at which they were posted by using the `timestamp`, `before`, `since`, and `age` keywords. You may find that the `age` keyword is the easiest of these keywords to use, and the most useful for everyday operation.

When you use the *`timestamp`* keyword, you must supply a string that defines a time range in the following way:

`year``:`*`month-of-year`*`:`*`day-of-month`*`:`*`day-of-week`*`:hours:`*`minutes`*`:`*`seconds`*

You can use an asterisk (`*`) as a wildcard character for any of the components, so to select events that occurred on July 6, 2002 you may use the following commands:

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority] @@"
# evmget -A -f '[timestamp 2002:7:6:*:*:*:*]' | more
```

The asterisks (*) in the final four components indicate that you are interested in all events that occurred on that day, no matter what time they occurred. Also, you can specify one or more ranges in any position, as shown in the following command:

```
# evmget -A -f '[timestamp 2002:*:*:1-3,5:*:*:*]' | more
```

The fourth component specifies the day of the week. Searching for events with posting times in the range 1-3 or 5 yields all events that were posted on a Monday, Tuesday, Wednesday or Friday in the year 2002.

The `before` and `since` keywords use similar specifier strings, but you cannot use wildcard characters and there is no day of the week indicator. For example, the following command finds events that were posted after 3:00p.m. on July 6, 2002:

```
# evmget -A -f '[since 2002:7:6:15:0:0]' | more
```

The `age` keyword provides a more convenient and intuitive way to select events according to their timestamps. As a system administrator you may be most interested in recent events that indicate a system problem. You can combine the event filter's `priority` and `age` keywords to find such events. For example, the following command sequence shows all events with a priority of error (400) or higher, that occurred either yesterday or today (the age of the event is less than 2 days):

```
# evmget -A -f '[pri >= 400] and [age < 2d]' | more
```

In the preceding example, `2d` specifies events that are less than 2 days old. You can specify an age in seconds (`s`), minutes (`m`), hours (`h`), days (`d`), or weeks (`w`). See `EvmFilter`(5) for information about how each specifier is used in calculating an event's age.

You can use a more complex filter to return events that occurred within a more specific period. The following example finds error events that occurred more than 3 days ago, but less than 6 days:

```
# evmget -A -f '[pri >= 400] and ([age < 6d] and
[age > 3d])' | more
```

See `EvmFilter`(5) for detailed information on selecting events according to their timestamps, and the full filter syntax.

13.3.12.2 Using the Event-Id to Select Events for Detailed Display

Using the `evmshow -d` command option to display events can result in a large amount of output and you may want to limit the number of displayed events. Events that are posted through Event Manager contain a sequential

identifier known as the event-id. You can use the event-id to select a specific event or a range of events for detailed display.

The event-id is not guaranteed to be unique within any particular set of events because the daemon's counter is set to zero each time it is restarted. To ensure that an event is unique, you must also use the timestamp when selecting events as shown in the following example:

```
# evmget -A -f '[age < 1d]' -t "@timestamp @event_id @@" | more

15-Apr-1999 14:19:06 0  EVM daemon: Configuration completed
15-Apr-1999 14:19:06 1  EVM daemon: Initialization completed
15-Apr-1999 14:19:06 2  EVM logger: Logger started
15-Apr-1999 14:19:06 3  EVM: Mark event - initial
15-Apr-1999 14:19:06 5  EVM logger: Started eventlog /var/evm/evmlog/evmlog.19990415
[1]              [2]
.
.
.
```

[1] The age filter keyword selects all events that have occurred today, as indicated by the timestamp in the first column of data.

[2] The @event_id specifier in the show template instructs the evmshow command to display the event-id for each retrieved event, which is shown in the second column of data.

When the event-ids are displayed, you can select the interesting events. For example, use the following command to display details of the initial mark event, which has an event-id of 3 in the preceding example output:

```
# evmget -f '[age < 1d] and [event_id = 3]' | evmshow -d | more
```

You can select a range of events by using a more complex filter as shown in the following example:

```
# evmget -f '[age < 1d] and [event_id >= 1] and [event_id <= 3]'|
 evmshow -d | more
```

Choose the time range carefully to select the right set of events. If you recently rebooted your system, specify a filter of [age < 2h] to select events occurring within the preceding 2 hours.

The most convenient way to select events for detailed display is to use the event viewer described in Section 13.3.11.

13.3.12.3 Searching for Reserved Component Names

Some event names include reserved component names as name extensions. These components begin with an underscore character (_), and usually are followed by a component that identifies the item for which the event is being posted. For example, the names of many hardware-related events include the component _hwid, followed by the numeric hardware identifier

of the item. Reserved component names are appended automatically as an extension to the event name. The name is appended, followed by the value for the named variable. This is done for every reserved component name. For example, an event with the name `@SYS_VP@.temperature_high` and the variable `_degrees` with the value 212 would be observed as an event with the name `@SYS_VP@.temperature_high._degrees.212`.

You can search for all such events by using the following command:

```
# evmget -A -f '[name *._hwid]' | more
```

If you know the hardware identifier of a specific device, you can narrow the search for events related to that device by using a command similar to the following:

```
# evmget -A -f '[name *._hwid.4]' | more
```

13.3.12.4 Using Filter Files

You can save a useful filter in a file and recall it by using the Event Manager's indirect filter facility. Filter files have names with the suffix `.evf`, and can contain any number of named filters. For example, the following filter file entry selects all `binlog` events that refer to SCSI devices:

```
filter {
    name "scsi"
    value "[name @SYS_VP@.binlog.hw.scsi]"
    title "Binlog SCSI events"
}
```

In this example, the `@SYS_VP@` is a standard Event Manager macro that is replaced by `sys.unix` when the filter is used.

To use indirect filtering, specify the at sign (@), followed by the name of the file containing the filter instead of a filter string, as shown in the following example:

```
# evmget -A -f @binlog
```

You do not need to include the `.evf` suffix when you specify a filter file name in such commands.

The previous example uses the first filter in the file, but you can choose a different filter by specifying its name as follows:

```
# evmget -A -f @binlog:scsi
```

You can include as many filters as you like in a single file, or you can keep each filter in its own file. The preceding example specifies the `binlog` filter, which is included in Event Manager. Other filters are provided in the `/usr/share/evm/filters` directory. Use these files as examples for establishing your own filter library.

The evmshow -F command option provides an easy way for you to see the contents of a stored filter. The -F option causes the evmshow command to display the filter string and then exit without reading any events. In the following example, the evmshow command displays the contents of the filter named scsi, stored in the binlog.evf file:

```
# evmshow -f @binlog:scsi -F
( [name sys.unix.binlog.hw.scsi] )
```

See evmfilterfile(4) for complete information about the syntax of filter files, and where to locate your files.

Note

Do not edit the filter files provided in the /usr/share/evm/filters directory. Your changes may be overwritten without warning by a future installation update.

13.3.13 Logging and Forwarding Events

The response to an event is any action determined by your site-specific needs and conditions. This response can range from activating alarms or paging responsible personnel, to making a log entry or ignoring an expected occurrence of a regular activity.

You can configure the event processing sequence to perform a series of dependent tasks, by using an event output by one task as the trigger to activate the next process. Event Manager provides an interface to the response activity through the logging facility. The available options are event storage and event forwarding.

The Event Manager logger, evmlogger, started automatically by the Event Manager daemon, is responsible for the following:

- Displaying selected events on the system console or other device

  If a terminal device is indicated as the logfile in the configuration file, all events meeting the filter specifications of an eventlog statement are formatted for display on the terminal. (See Section 13.2.2.3 for a discussion of the configuration file.)

- Storing selected events in one or more log files
- Forwarding selected events to interested parties in some other form

By default, the logger handles events posted through its local daemon, but you can also configure it to handle events posted on remote systems; see Section 13.2.3.3 for more information.

The logger is an ordinary Event Manager client that is controlled through a configuration file. The default is the `/etc/evmlogger.conf` file, described in Section 13.2.2.3. See `evmlogger.conf`(4) for more information on this file and `evmlogger`(8) for more information on the command.

13.3.13.1 Logging Events

All events meeting the specifications of an `eventlog` group in the configuration file are written to the event log. See Section 13.1.3.3 for the default location of this file and the naming conventions.

As shown in Example 13–3, you can include a `suppress` group specification in an `eventlog` statement in the configuration file. When you include such a statement, events meeting the suppression criteria are not entered in the log. One instance of the event is stored, with additional data indicating the number of events and the time of the first and last occurrence of the event. See `evmlogger.conf`(4) for the explanation of this criterion.

13.3.13.2 Using Forwarding to Handle Events Automatically

If you want to automate the handling of selected events, you can configure the Event Manager logger to forward the event by executing a command. For example, you can mail the event information to a paging service, or invoke an event-handling application program.

By default, the logger is configured to mail high priority events to the root user. You can use that default forwarding command as an example for developing your own actions. See Section 13.2.2.3 and `evmlogger.conf`(4) for more information.

All events meeting the filter specifications of a `forward` statement in the configuration file are written to the standard input (`stdin`) of the command specified in the statement. The command is the name of a shell script, a single UNIX command, a series of UNIX commands (pipeline), or any other executable statement. The following operations are typically specified as a forwarding action:

- Specifying the `mail` command or `mailx` command, or another command line mail processor, to send a mail message to a responsible person or paging service
- Invoking additional software that causes emergency shutdown procedures to commence
- Invoking a dependent process that is waiting for the event to occur

When configuring the logger to forward an event, note the following:

- The event selected for forwarding is piped into the configured forwarding command. If your commands need to deal with text information, the

`evmshow` command must be the first command in the pipeline so that the event is converted to text form.

- The logger executes the forwarding command asynchronously, meaning that it starts the command and then continues with its normal operation without waiting for the command to finish. The following behaviors are normal:
  - If multiple forwarders are specified in the logger's configuration file, and the same event is to be handled by more than one forwarder, the logger starts each forwarding command without waiting for the others to finish, so the commands may execute simultaneously.
  - If the logger receives another event to be processed by a forwarding command, and the command is still processing the previous event, the logger queues the new event. When the command finishes, the logger restarts it, passing it the new event. By default, the logger queues up to 100 events for each forwarding command. You can increase this limit by specifying a `MAXQUEUE` keyword in the forwarder's configuration.

  See `evmlogger.conf`(4) for more information.
- Event text may include characters such as quotes, which have special meaning to the shell. Be sure to post test versions of the event to verify that your command executes correctly under realistic conditions.
- You must take care that the forwarding command does not itself result in the posting of events which would cause an event loop. For example, if you use mail to forward events, the forwarder's filter must exclude mail events.

Use the logger's secondary configuration file facility for adding forwarders or other configuration items as described in Section 13.2.2.4.

13.3.13.3 Logging Events from Remote Systems

By default, the logger only subscribes for events that are posted through its local daemon. You can configure the logger to log or forward events posted on other systems by adding one or more remote_hosts sections to your configuration. The best way to do this is to specify remote connections in a secondary configuration file. See Section 13.2.2.4 for information about secondary configuration files.

The logger reports the status of remote connections by posting an event each time it establishes, loses or reestablishes a connection.

Example 13–6 shows a sample remote_hosts section. See `evmlogger.conf`(4) for full syntax details.

Example 13–6: Sample Logger Configuration File Entries for Remote Logging

```
remote_hosts {
        name            appsys_hosts                    1
        hostnames       appsys1,appsys2
        hostnames       appsys3                         2
        targets         appsys_log                      3
        filter          "[priority >= 400]"             4
        retry           10                              5
    }

    eventlog {                                          6
        name            appsys_log
        logfile         /eventlogs/applog.dated
        explicit_target yes                             7
        filter          all                             8
    }
```

1 The `name` keyword identifies this group.

2 The `hostnames` lines specify the list of remote hosts to which the logger subscribes. This list can be split across multiple lines, and each line can include any number of hosts.

3 The `targets` line indicates that events received from the remote nodes listed in this group are logged in the event log defined by the eventlog group named `appsys_log`. Targets can be either event logs or forwarders.

4 The `filter` line specifies that all events with a priority of 400 or greater that are posted on any of the remote systems are logged.

5 The `retry` line specifies that the logger should attempt to reconnect once every ten seconds if a connection to any of the remote hosts cannot be established or is subsequently lost.

6 This is the `eventlog` group that is targeted by the sample remote_hosts section.

7 Setting the `explicit_target` keyword to `yes` (or `true`) ensures that this eventlog group only logs events that are received through `remote_hosts` groups that specify its name in a target line. Events received from the local daemon are not handled by this group.

8 Setting the filter for the log to `all` ensures that it logs all events that are received from the remote hosts. The filter strings specified by the `remote_hosts` group restricts the set of events that are received.

If you change the configuration, remember to run the `evmreload -l` command to make the logger recognize the changes.

If you specify a `remote_hosts` section in a TruCluster environment, the individual loggers running on the various cluster nodes each establish the same set of remote connections, and the events are logged separately on each node of the cluster. If you want to change this behavior, use the `mkcdsl` command to establish member-specific versions of the secondary configuration file. See the `mkcdsl`(8) for information about context sensitive symbolic links (CDSLs).

13.4 Troubleshooting Event Manager

If you suspect that Event Manager is not operating correctly, the first step is to examine the message files in the `/var/evm/adm/logfiles` directory. Messages in these files are displayed also through the Event Manager viewer and `evmget`, as part of the `misclog` event channel.

The following list describes some common problems and the initial steps to take in trying to resolve such problems:

- `Kernel events are not being posted`

  Verify the Event Manager daemon log file for errors by using the following command:

  ```
  # more /var/evm/adm/logfiles/evmdaemon.log
  ```

  Examine for the presence of the kernel interface pseudodevice by using the following command:

  ```
  # ls -l /dev/kevm
  ```

  If this pseudodevice is not present, create it by using the following command:

  ```
  # dsfmgr -vF
  ```

- `A subscribing application fails to receive expected events`

  Verify that the poster is authorized to post these events by examining the authorization file with the following command:

  ```
  # more /etc/evm.auth
  ```

  Verify that the event is registered by using the following command:

  ```
  # evmwatch -i -f '[name event_name]' |
    evmshow -t "@name"
  ```

  If the events are still not shown, run `evmreload` and examine it again. If they are still not visible, verify that the template files are correctly installed.

  Verify that the subscriber is authorized to access these events, by using the following command:

  ```
  #  more /etc/evm.auth
  ```

  Verify that the expected events are actually being posted by using the following command:

  ```
  # evmwatch | evmshow -t "@name @@"
  ```

  Run the program that posts the event, and verify that the preceding `evmwatch` command displays them correctly.

- `A posting program is unable to post events`

Verify that the Event Manager daemon is running by using the following command:

```
# ps -aef | grep evmd
```

Verify that the poster is authorized to post these events by examining the authorization file by using the following command:

```
# more /etc/evm.auth
```

Verify that the event is registered by using the following command:

```
# evmwatch -i -f '[name event_name]' |
  evmshow -t "@name"
```

If the events are still not shown, run the `evmreload` command and examine it again. If they are still not visible, verify that the template files are correctly installed.

- `Expected syslog or binlog events are not visible through Event Manager`

  You must either be logged in as root or belong to the `adm` group in order to access `syslog` and `binlog` events.

  By default, Event Manager only retrieves `binlog` events that were posted within the last 8 days. If you want to see older `binlog` events, edit the channel configuration file, `/etc/evmchannel.conf`. In the `binlog` channel group, the default `fn_get` line includes the option `-r 8d`, meaning that events for only the past 8 days are retrieved. You can either remove this option completely to see all `binlog` events or change the 8 to some other value.

  Use the `ps` command to verify that the `binlogd` and `syslogd` daemons are running.

  Be sure that the `/etc/syslog_evm.conf` file is configured to forward the events you expect to see.

  Use the following commands to test communication with `syslog` and `binlog`:

  ```
  # evmwatch | evmshow &# logger "test syslog message"
  # logger -b "test binlog message"
  ```

- `Event retrieval through evmget or the event viewer is slow`

  Examine the sizes of all log files, particularly the `evmlog` files (`/var/evm/evmlog`), the binary error log (`/var/adm/binary.errlog`), and the SysMan Station daemon log files (`/var/adm/sysman/sysman_station/logs`).

  Use the `ls -L` command when listing file sizes to ensure that you see the file itself and not a symbolic link or a context-dependent symbolic link (CDSL).

See binlogd(8) for details of binary log size management, but note that Event Manager retrieves events from the archive log file, so starting a new log may not immediately reduce the number of events available to the Event Manager. You can use the cron utility to perform a regular archiving task. You can reduce the sizes of the evmlog files by changing configuration values in the /etc/evmlogger.conf file and the /etc/evmchannel.conf file.

- Expected events are not being logged

  Examine the event priority. Only events with a priority of 200 or higher are logged by the Event Manager logger.

- Cannot post or subscribe to events through a remote daemon

  Ensure that remote access is configured and if it is not:

  1. Set remote_connection to True in the remote daemon's configuration file.
  2. Verify that the remote host is specified by the /etc/evm.auth file. You can use the following command:

     ```
     # grep remote-host /etc/evm.auth
     ```

  3. If the specified authentication type is evm_callback, try changing it to evm_open temporarily. If this is successful, the problem may be with the specification of the remote hosts or user mapping (that is, the local user may not exist or have the desired permissions). For further information, see Section 13.2.3.3. Be sure to reset the authentication type to evm_callback.
  4. Run the following command:

     ```
     # evmreload -d
     ```

  Be sure to consider the security implications of enabling a remote connection. See Section 13.2.3 for more information about security.

- Invalid filter message from remote connections

  This may happen when an attempt to connect to a remote system to retrieve or monitor events results in an invalid filter message, even though the same filter works correctly when used on the local system.

  The filter syntax changes with new releases of the operating system, and newer keywords or abbreviations may not be recognized by older versions. Log in to the remote system and review EvmFilter(5) to determine whether the syntax used in your filter is supported by that version of the operating system.

- Binlog events are not being translated

Use the following procedures to troubleshoot the absence of a translation utility:

1. Run the following command:

   # **usr/sbin/dia**

   If DECevent is installed, this command displays the translated contents of the current binary error log file, `/var/adm/binary.errlog`.

2. If the `dia` command is not found, use the following command to test the status of the DECevent software subset (the distribution kit):

   # **setld -i | grep OSFDIA**

   This command returns the string `OSFDIABASE***` when the DECevent Base Kit (Translation/Analysis) is installed. If it is not installed, mount the installation media and use the `setld` command to install the subset. See `setld`(8) for more information.

Verify the presence of Compaq Analyze as follows:

1. Use the following command to see if the Compaq Analyze director service is running on the local host:

   # **ps agx | grep desta**

2. If the `desta` daemon is not running, the Compaq Analyze utility may be installed but not running or properly configured. To verify that Compaq Analyze is installed, look for the binaries by using the following command:

   # **ls /usr/opt/compaq/svctools/bin/desta***

3. If you do not find the binaries, install Compaq Analyze from the distribution media by using the `setld` command. Contact your Sales and Support organization or your local vendor for information on Compaq Analyze.

- `evmlogger: Missed receipt of` *`number`* `events`

  This error occurs when events overflow the receive buffer, whose size is set to the default system socket buffer maximum. You can alter this value as follows:

  1. Use the `sysconfig` command to determine the current value of the `sb_max` system parameter.

     # **sysconfig -q socket sb_max**

  2. Estimate the new buffer size.

  3. Change the runtime value for this parameter with the following command:

```
# sysconfig -r socket sb_max=newvalue
```

This change remains in effect until the next reboot.

4. When you decide to make this change permanent, use either `sysconfigdb` or `dxkerneltuner` to change the value of `sb_max`.

14

Administering Crash Dumps

This chapter describes how you configure and generate system crash dumps and how you save and store crash dumps and their associated data using either the Graphical User Interface or manually. Crash dumps are a snapshot of the running kernel, taken automatically when the system shuts down unexpectedly. Crash dumps are referenced most often when you contact your technical support representatives to analyze and correct problems that result in a system crash. However, if you are an experienced system administrator or developer you may be familiar with techniques of crash dump analysis and you may want to take and analyze your own dump files.

The following topics are discussed in this chapter:

- An overview of crash dumps (Section 14.1)
- A discussion of two new graphical user interfaces for crash dump configuration and creating a crash dump “snap shot” (Section 14.2)
- A discussion on how to create a crash dump (Section 14.3)
- Information on choosing the content and method of a crash dump (Section 14.4)
- Instructions on how to take a crash dump manually (Section 14.5)
- Information on how to store and archive crash dumps (Section 14.6)

14.1 Overview of Crash Dumps

When a system shuts down unexpectedly, it writes all or part of the data in physical memory either a) to swap space on disk (the virtual memory space) or b) to memory. Such shutdown events are referred to as system crashes or panics. The stored data and status information is called a crash dump. Crash dumps differ from the core dumps produced by an application, after which the system usually keeps running. After a crash dump, the system is shut down to the console prompt (>>>) and may or may not need to be rebooted, depending on the `auto_action` Boot Halt Restart option.

During the reboot process, the system moves the crash dump into a file and copies the kernel executable image to another file. Together, these files are the crash dump files and are often required for analysis when a system crashes or during the development of custom kernels (debugging). You may

need to supply a crash dump file to your technical support organization to analyze system problems.

To administer dumps, you must understand how crash dump files are created. Also, you must reserve space on disks for the crash dump and crash dump files. The amount of space you reserve depends on your system configuration and the type of crash dump you want the system to perform.

14.1.1 Related Documentation and Utilities

Crash dumps make use of the virtual memory swap space provided on disk. Administering the swap space is described in Chapter 3. System event management is described in Chapter 12, which describes the `binlogd` and `syslogd` event management channels.

Additional information on crash dumps and related topics is available in manuals and reference pages.

14.1.1.1 Manuals

The following lists manuals that provide useful information for crash dumps and related topics.

- The *Kernel Debugging* manual provides information on analyzing crash dumps. You may need to install software development subsets and appropriate licenses to use this feature.
- The *Installation Guide* manual provides information on the initial swap space and dump settings configured during installation.

14.1.1.2 Reference Pages

The reference pages listed here provide further information regarding associated utilities.

`savecore`(8)	The program that copies dump data from swap partitions or from memory to a file.
`expand_dump`(8)	Decompresses a kernel crash dump file.
`dumpsys`(8)	Copies a snapshot of memory to a dump file without halting the system. This is known as a continuable dump and is useful for estimating crash dump size during dump configuration planning.
`sysconfig`(8) and `sysconfigdb`(8)	Maintains the kernel subsystem configuration and is used to set kernel crash dump attributes that control crash behavior. You can use

	the Kernel Tuner graphical user interface (`/usr/bin/X11/dxkerneltuner`) to modify kernel attributes. See `dxkerneltuner`(8) for information. Online help is also available for this interface. The Kernel Tuner can be launched from CDE and is located in the Application Manager: System Admin folder.
`swapon`(8)	Specifies additional files for paging and swapping. Use this command if you need to add additional temporary or permanent swap space to produce full dumps.
`dbx`(1)	The source level debugger.

14.1.1.3 SysMan Menu Applications

Applications for configuring and creating crash dumps are available from the SysMan Menu:

Configure System Dump	Use this application to configure the generic system configuration variables associated with the `savecore` command.
Create Dump Snapshot	Use this application to configure the `dumpsys` command, which dumps a snapshot of memory manually.

See Section 14.2 for more information.

14.1.2 Files Used During Crash Dumps

By default, the `savecore` command copies a crash dump file into the `/var/adm/crash` directory, although you can redirect crash dumps to any file system that you designate and also to a remote host. In common with many other system directories, the `/var/adm/crash` directory is a context-dependent symbolic link (CDSL), which facilitates joining systems into clusters. The CDSL for this directory is `/var/cluster/members/member0/adm/crash`. Within this directory, the following files are created or used:

`/var/adm/crash/bounds`	A text file specifying the incremental number of the next dump (the `n` in `vmzcore.n`)

`/var/adm/crash/minfree`	A file that specifies the minimum number of kilobytes to be left after crash dump files are written
`/var/adm/crash/vmzcore.n`	The crash dump file, named `vmcore.n` if the file is not compressed (no `z`)
`/var/adm/crash/vmunix.n`	A copy of the kernel that was running at the time of the crash, typically of `/vmunix`
`/etc/syslog.conf`, `/etc/binlog.conf`, and `/etc/evmdaemon.conf`	The logging configuration files

14.2 Crash Dump Applications

There are two applications that simplify the processes of configuring crash dumps and creating crash dump files manually. These applications are available from the `Support and Services` branch of the SysMan Menu.

The first application is Configure System Dump. Its purpose is to configure the parameters of the system dump so that you have the appropriate information for your needs should a crash dump occur in the future.

The second application is Create Dump Snapshot. It allows you to set various options and to take a snapshot of memory, which is stored in a file for examination when you cannot halt the system to generate a crash dump.

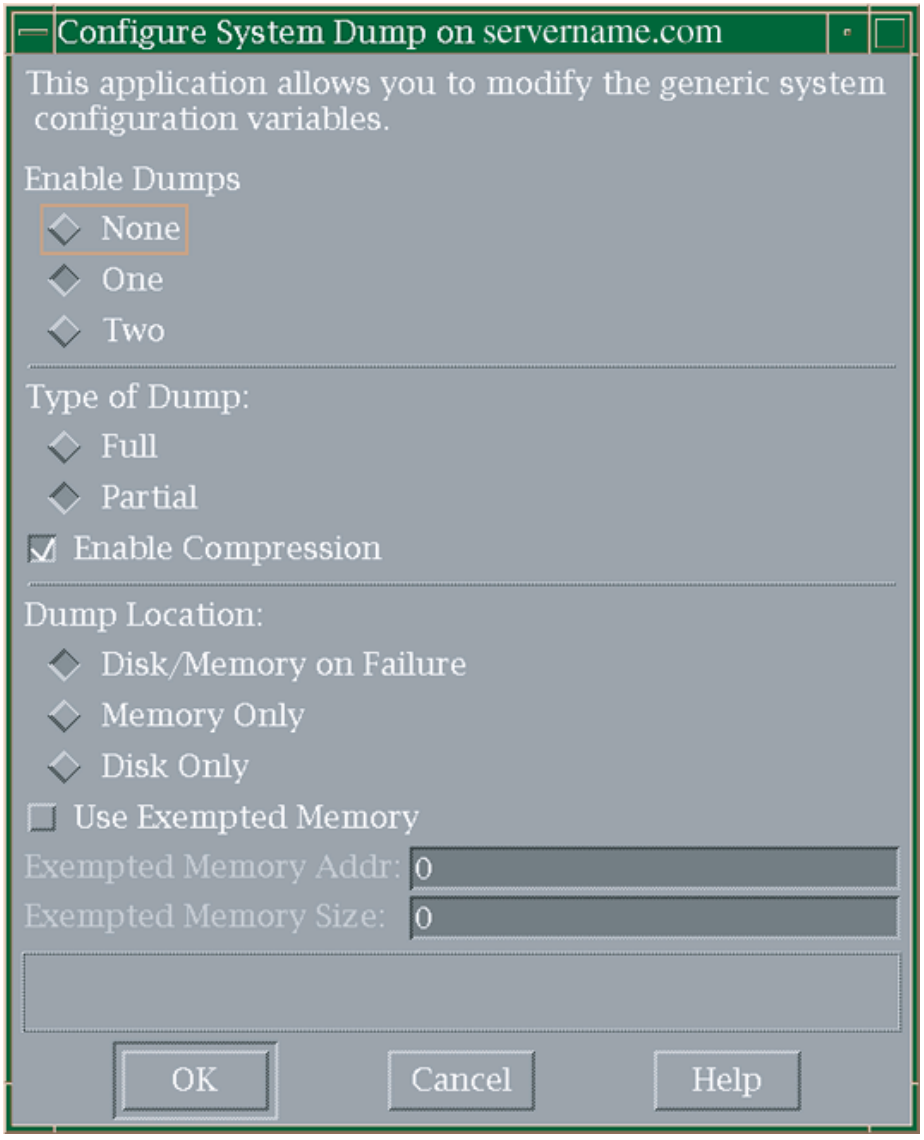
14.2.1 Using the Configure System Dump Application

The Configure System Dump application lets you tailor the crash dump data according to your needs. This application allows you to set various options that influence the crash dump file should a crash dump occur in the future.

You can access this application from the SysMan Menu by selecting `Support and Services` then selecting `Configure Dump`.

Figure 14–1 shows the main window of this application.

Figure 14–1: Configure System Dump application



After you invoke this application from the SysMan Menu, you can provide the following information:

1. The first selection, `Enable Dumps`, requires that you choose one of the following:

None	Disables the mechanism to generate a crash dump.

One	Enables the mechanism so that one set of crash dump files (the crash dump file and a copy of the kernel) is written, should a crash dump occur.
Two	Also enables the mechanism so that a set of crash dump files is written, should a crash dump occur. This option also provides for a subsequent set of crash dump files if an additional system fault occurs while the crash dump files are written.

2. In the second selection, you can choose a Full or Partial dump.

   A full dump saves the crash dump header information and all physical memory.

   A partial dump saves the crash dump header and copy of the part of the physical memory believed to contain significant information at the time of the system crash selected portion of physical memory.

3. You may choose to compress the crash dump file with the `Enable Compression` check box. You should always enable compression unless some reason dictates otherwise.

4. The next selection, `Dump Location`, specifies how the crash dump data is stored:

Disk/Memory on Failure	Saves the crash dump file to disk. If this fails, a partial compressed memory dump is attempted.
Memory Only	Saves the crash dump file to the memory space.
Disk Only	Saves the crash dump file to disk; no attempt of a partial compressed memory file is attempted on failure.

5. In the final selections, you can specify whether or not the crash dump file should be dumped to exempted memory. If so, select the `Use Exempted Memory` check box to enable the following two fields:

Exempted Memory Address	Specify the starting memory address where the dump should be saved.
Exempted Memory Size	Specify the size of the memory region.

Note

These fields accept decimal and hexadecimal entries. Be sure to precede all hexadecimal entries wth `0x`.

The Configure System Dump application offers online help, which provides more information.

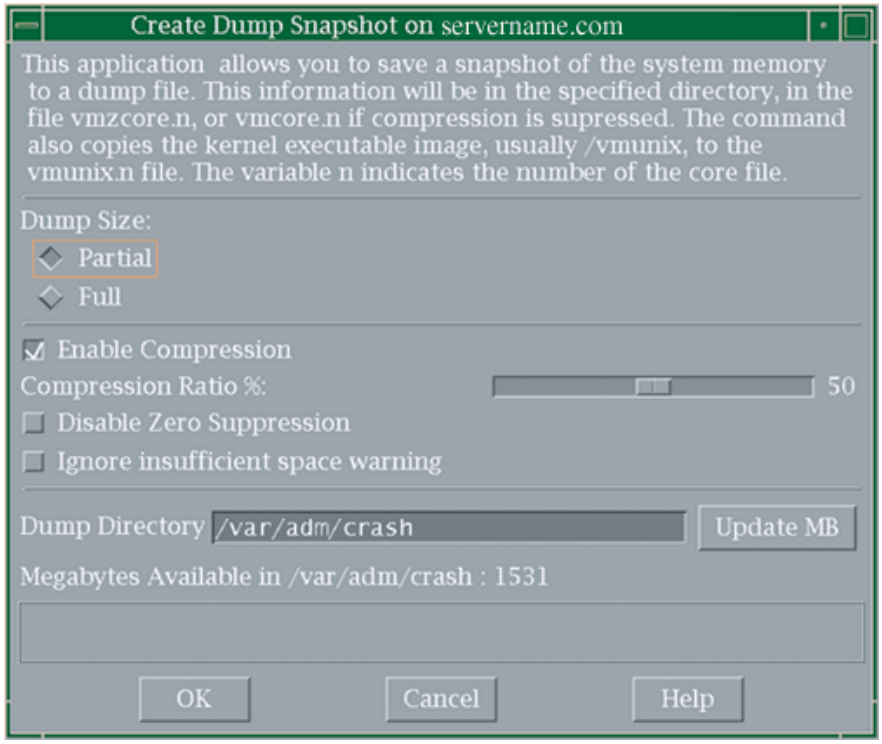
14.2.2 Using the Create Dump Snapshot Application

The Create Dump Snapshot application, illustrated in Figure 14–2 , allows you to save a snapshot of system memory to a dump file.

You can access this application from the SysMan Menu by selecting `Support and Services` then selecting `Create Dump Snapshot`.

Figure 14–2 shows the main window of this application.

Figure 14–2: Create Dump Snapshot application



After you invoke this application from the SysMan Menu, you can provide the following information:

1. Designate a full or partial dump.
2. Specify whether or not you want the data compressed. If so, use the `Compression Ratio %` slide bar to specify the compression ratio; a lower value increases the compression, if possible.
3. Indicate whether the utility should suppress contiguous zeroes with the `Disable Zero Suppression` check box. This suppression is not recommended.
4. Select the `Ignore insufficient space warning` check box unless you want the application to warn you if there was not enough space to save the crash dump data.
5. Enter the full pathname for the directory, where you would like the crash dump file to be written, in the `Dump Directory` field. The number of megabytes available in that directory is displayed in the `Megabytes Available in` field. Select `Update MB` to update that display field.

The Create Dump Snapshot application offers online help which provides more information.

14.3 Crash Dump Creation

After a system crash, you normally reboot your system by issuing the `boot` command at the console prompt. During a system reboot, the `savecore` command moves crash dump information from the swap partitions or memory into a file and copies the kernel that was running at the time of the crash into another file. You can analyze these files to help you determine the cause of a crash. The `savecore` command also logs the crash in system log files.

You can invoke the `savecore` command from the command line. See `savecore`(8) for information.

14.3.1 Setting Dump Kernel Attributes in the Generic Subsystem

You can control the way that a crash dump is taken by setting kernel attributes defined in the `generic` subsystem, as follows:

`dump_savecnt`	Limits the number of successful crash dumps that are generated for a single crash and reboot sequence or disables dumping. See Section 14.3.2.

`dump_to_memory`	Specifies whether primary system core dumps are written to memory or to disk. See Section 14.3.2.
`dump_sp_threshold`	Controls the partitions to which the crash dump is written. The default value causes the primary swap partition to be used exclusively for crash dumps that are small enough to fit the partition. See Section 14.3.4.
`dump_user_pte_pages`	Specifies whether or not you want to include user page tables in partial crash dumps. This attribute is off by default. See Section 14.4.2.
`expected_dump_compression`	Specifies the level of compression that you typically expect the system to achieve. The setting is 500 by default, but can be an integer from 0 to 1000. See Section 14.4.4.
`partial_dump`	Specifies whether a partial crash dump or a full crash dump is preserved. This attribute is on by default. See Section 14.4.3.
`compressed_dump`	Specifies whether a dump is compressed to save space. This attribute is on by default. Even if set to off, the value of other dump attributes may cause it to be automatically set to on. See Section 14.4.5 and also Section 14.4.6.
`dump_kernel_text`	Enables or disables the inclusion of kernel text pages in the dump creating a larger dump file. This attribute only applies when partial dumps are enabled. See Section 14.4.3.
`live_dump_dir_name`	Specifies the full path to the directory where continuable dumps are written. See Section 14.5.1.

`live_dump_zero_suppress`	Enables or disables zero compression of continuable dumps. Dump files take slightly longer to create but occupy less space. See Section 14.5.1.

If available, dumping to exempt memory is controlled by the following attributes:

`dump_exmem_addr`	Identifies the starting address (virtual or physical) for a region of exempt memory used for writing primary dumps.
`dump_exmem_size`	Specifies the size (in bytes) of the exempt memory region to which dumps are written.
`dump_exmem_include`	Specifies whether or not exempt memory pages are included in the dump.

See Section 14.4.6 for a description of this feature.

The following command displays typical dump attribute settings:

```
# sysconfig -q generic | grep dump
compressed_dump = 1
dump_exmem_addr = 0
dump_exmem_size = 0
dump_exmem_include = 0
dump_kernel_text = 0
dump_savecnt = 1
dump_sp_threshold = 4096
dump_to_memory = 0
dump_user_pte_pages = 0
expected_dump_compression = 500
live_dump_zero_suppress = 1
live_dump_dir_name = /var/adm/crash
partial_dump = 1
```

See `sys_attrs_generic`(5) for a description of the dump attributes and settings. See `sysconfig`(8) and `sysconfigdb`(8) for information on setting attribute values.

14.3.2 Crash Dump File Creation

When the `savecore` command begins running during the reboot process, it determines whether a crash dump occurred and whether the file system contains enough space to save it. (The system saves no crash dump if you

shut it down and reboot it; that is, the system saves a crash dump only when it crashes.)

The value of the dump_savecnt attribute controls the number of dumps. Possible values are:

0 (zero)	Never generate a crash dump.
1	Generate a primary crash dump (the default).
2	Generate a secondary crash dump.

The value of the dump_to_memory attribute controls the location of dumps and interacts with the value of the dump_savecnt attribute as follows:

-1	Writing dumps to memory is disabled. This value also disables writing a secondary dump when the value of the dump_savecnt attribute is 2.
0 (zero)	Dumps are written to disk except in the event of disk failure, in which case they are written to memory. This is the default behavior.
1	Dumps are written only to memory when sufficient memory is available. A special case is if secondary dumps are enabled (dump_savecnt=2). See sys_attrs_generic(5) for more information.

Under certain circumstances, dumps in memory may be overwritten. To prevent an overwrite from happening, you can write dumps to a protected region of memory called exempt memory. See Section 14.4.6 for more information.

If a crash dump exists and the file system contains enough space to save the crash dump files, the savecore command moves the crash dump and a copy of the kernel into files in the default crash directory, /var/adm/crash. (You can modify the location of the crash directory.)

You can choose to:

- Write all crash files to a remote host using a network connection as described in Section 14.4.7.
- Write continuable dump files to an alternate directory as described in Section 14.5.1.

The savecore command stores the kernel image in the vmunix.*n* file, and by default it stores the (compressed) contents of physical memory in the vmzcore.*n* file.

The *n* variable specifies the number of the crash, which is recorded in the bounds file in the crash directory. After the first crash, the savecore command creates the bounds file and stores the number 1 in it. The command increments that value for each succeeding crash.

The savecore command runs early in the reboot process so that little or no system swapping occurs before the command runs. This practice helps ensure that crash dumps are not corrupted by swapping.

14.3.3 Crash Dump Logging

After the savecore command writes the crash dump files, it performs the following steps to log the crash in system log files:

1. Writes a reboot message to the /var/adm/syslog/auth.log file.

   If the system crashed because of a panic condition, the panic string is included in the log entry.

   You can cause the savecore command to write the reboot message to another file by modifying the auth facility entry in the syslog.conf file. If you remove the auth entry from the syslog.conf file, the savecore command does not save the reboot message.

2. Attempts to save the kernel message buffer from the crash dump.

   The kernel message buffer contains messages created by the kernel that crashed. These messages may help you determine the cause of the crash.

   The savecore command saves the kernel message buffer in the /var/adm/crash/msgbuf.savecore file, by default. You can change the location to which savecore writes the kernel message buffer by modifying the msgbuf.err entry in the /etc/syslog.conf file. If you remove the msgbuf.err entry from the /etc/syslog.conf file, savecore does not save the kernel message buffer.

   Later in the reboot process, the syslogd daemon starts up, reads the contents of the msgbuf.err file, and moves those contents into the /var/adm/syslog/kern.log file, as specified in the /etc/syslog.conf file. The syslogd daemon then deletes the msgbuf.err file. See syslogd(8) for more information about how system logging is performed.

3. Attempts to save the binary event buffer from the crash dump.

   The binary event buffer contains messages that can help you identify the problem that caused the crash, particularly if the crash resulted from a hardware error.

The savecore command saves the binary event buffer in the /usr/adm/crash/binlogdumpfile file by default. You can change the location to which savecore writes the binary event buffer by modifying the dumpfile entry in the /etc/binlog.conf file. If you remove the dumpfile entry from the /etc/binlog.conf file, savecore does not save the binary event buffer.

Later in the reboot process, the binlogd daemon starts up, reads the contents of the /usr/adm/crash/binlogdumpfile file, and moves those contents into the /usr/adm/binary.errlog file, as specified in the /etc/binlog.conf file. The binlogd daemon then deletes the binlogdumpfile file. See binlogd(8) for more information about how binary error logging is performed.

4. The system may crash before all kernel events are handled and posted. In such cases, the savecore program recovers such events and stores them for later processing. This recovery happens only if any such events are available and if the savecore program is able to extract and save the events successfully. By default, the events are stored in the /var/adm/crash/evm.buf file. See savecore(8) and EVM(5) for more information.

14.3.4 Swap Space

When the system creates a crash dump to disk, it writes the dump to the swap partitions. The system uses the swap partitions because the information stored in those partitions has meaning only for a running system. After the system crashes, the information is useless and can be overwritten safely.

Before the system writes a crash dump, it determines how the dump fits into the swap partitions, which are defined in the /etc/sysconfigtab file. For example, the following fragment of the /etc/sysconfigtab file entry shows three swap partitions available:

```
vm:
    swapdevice=/dev/disk/dsk0b, /dev/disk/dsk3h, /dev/disk/dsk13g
    vm-swap-eager=1
```

The following list describes how the system determines where to write the crash dump:

- If the crash dump fits in the primary swap partition it is dumped to the first partition listed under swapdevice in the /etc/sysconfigtab file. The system writes the dump as far toward the end of the partition as possible, leaving the beginning of the partition available for boot-time swapping.

- If the crash dump is too large for the primary swap partition, but fits the secondary or tertiary swap space, the system writes the crash dump to the other swap partitions, `/dev/disk/dsk3h` and `/dev/disk/dsk13g`.
- If the crash dump is too large for any of the available swap partitions, the system writes the crash dump spanning the secondary and tertiary swap partitions until those partitions are full. If it requires more space, it then writes the remaining crash dump information starting from the end of the primary swap partition (possibly filling the primary swap partition also).
- If the aggregate size of all the swap partitions is too small to contain the crash dump, the system creates no crash dump.

Each crash dump contains a header, which the system always writes to the end of the primary swap partition. The header contains information about the size of the dump and where the dump is stored. This information allows `savecore` to find and save the dump at system reboot time.

In most cases, compressed dumps fit on the primary swap partition. The next section describes `dump_sp_threshold`, which is relevant in understanding how a crash dump is created. The use of the remaining kernel attributes controls the content of the dump. These attributes are described in Section 14.4.

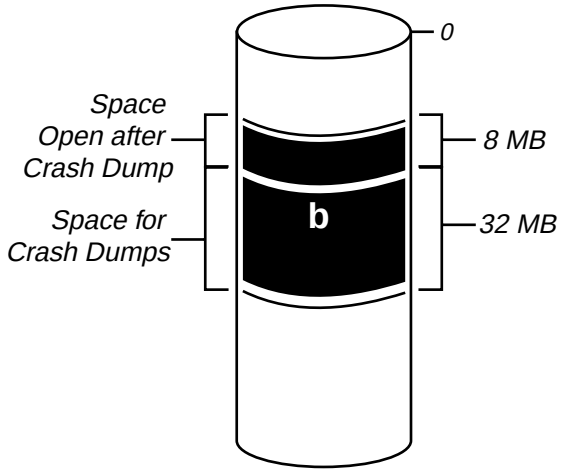
Controlling the Use of Swap Partitions

You can configure the system so that it fills the secondary swap partitions with dump information before writing any information (except the dump header) to the primary swap partition. The attribute that you use to configure where crash dumps are written first is the `dump_sp_threshold` attribute.

The value in the `dump_sp_threshold` attribute indicates the amount of space you normally want available for swapping as the system reboots. By default, this attribute is set to 16,384 blocks, meaning that the system attempts to leave 8 MB of disk space open in the primary swap partition after the dump is written.

Figure 14–3 shows the default setting of the `dump_sp_threshold` attribute for a 40 MB swap partition. (40 MB is not typical of a swap partition size on most systems, the example uses small numbers for the sake of simplicity.)

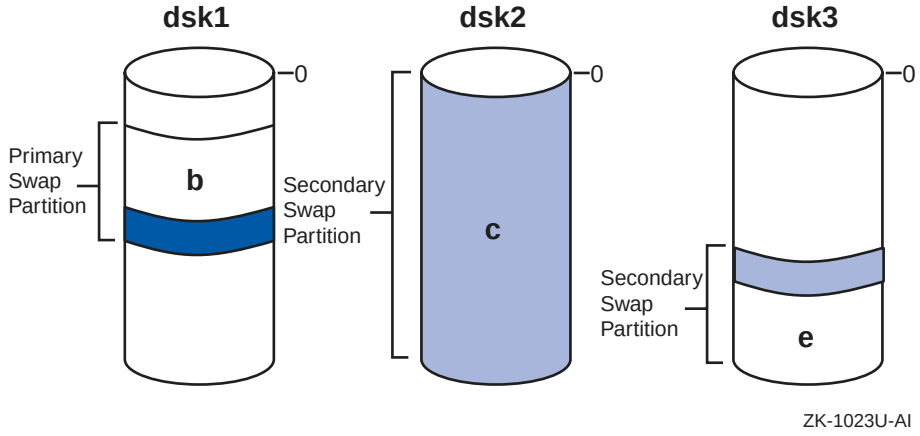
Figure 14–3: Default dump_sp_threshold Attribute Setting



The system can write 32 MB of dump information to the primary swap partition shown in Figure 14–3. Therefore, a 30 MB dump fits on the primary swap partition and is written to that partition. However, a 40 MB dump is too large; the system writes the crash dump header to the end of the primary swap partition and writes the rest of the crash dump to secondary swap partitions, if available.

Setting the `dump_sp_threshold` attribute to a high value causes the system to fill the secondary swap partitions before it writes dump information to the primary swap partition. For example, if you set the `dump_sp_threshold` attribute to a value that is equal to the size of the primary swap partition, the system fills the secondary swap partitions first. (Setting the `dump_sp_threshold` attribute is described in Section 14.4.1.) Figure 14–4 shows how a crash dump is written to secondary swap partitions on multiple devices.

Figure 14–4: Crash Dump Written to Multiple Devices



If a noncompressed crash dump fills partition `e` in Figure 14–4, the system writes the remaining crash dump information to the end of the primary swap partition. The system fills as much of the primary swap partition as is necessary to store the entire dump. The dump is written to the end of the primary swap partition to attempt to protect it from system swapping. However, the dump can fill the entire primary swap partition and may be corrupted by swapping that occurs as the system reboots.

Estimating Crash Dump Size Using dumpsys

To estimate the size of crash dumps, you can use the `dumpsys` command, which produces a run time or continuable dump. See Section 14.5.1 for information on using the `dumpsys` command. You may need to temporarily create file system space to hold the experimental dumps. You can produce both full and partial dumps using this method. Crash dumps are compressed by default unless you specify the `dumpsys -u` command option. You use the `expand_dump` command to produce a noncompressed dump from the compressed output of the `dumpsys` command.

Because the crash dumps written to swap are about the same size as their resulting saved crash dump files, you can easily determine how large a crash dump was by examining the size of the resulting crash dump file. For example, to determine the size of the first crash dump file created by your system, enter the following command:

```
# ls -s /var/adm/crash/vmzcore.0
20480 vmzcore.0
```

This command displays the number of 512-byte blocks occupied by the crash dump file. In this case, the file occupies 20,480 blocks, so you know that a crash dump written to the swap partitions also occupies about 20,480 blocks.

In some cases, a system contains so much active memory that it cannot store a crash dump on a single disk. For example, suppose your system contains 2 GB of memory but only has several 4 GB disks, most of which are dedicated to storing data. Crash dumps for this system may be too large to fit on a single swap partition on a single device. To cause crash dumps to spread across multiple disks, create a second (and perhaps tertiary) swap partitions on several disks. The system automatically writes dumps that are too large to fit in the specified portion of the primary swap partition to other available swap partitions.

14.3.5 Planning Crash Dump Space

Because crash dumps are written to the swap partitions on your system, you allow space for crash dumps by adjusting the size of your swap partitions, thereby creating temporary or permanent swap space. See `swapon`(8) for information about modifying the size of swap partitions.

Note

Be sure to list all permanent swap partitions in the `/etc/sysconfigtab` file. The `savecore` command, which copies the crash dump from swap partitions to a file, uses the information in the `/etc/sysconfigtab` file to find the swap partitions. If you omit a swap partition from the `/etc/sysconfigtab` file, the `savecore` command may be unable to find the omitted partition.

Space requirements can vary from system to system. During the installation procedure the following algorithm to calculate required space in the `/var` file system is used:

```
3 * memsize / 24MB + 3 * 15MB
```

Where `memsize` is the amount of physical memory in megabytes and 15MB is the approximate size of a custom kernel. This algorithm allows for the preservation of three dumps. The following sections give you guidelines for estimating the amount of space required for partial and full crash dumps on your system. In addition, setting the `dump_sp_threshold` attribute is described.

14.3.6 Planning and Allocating File System Space for Crash Dump Files

Using the information on typical crash dump sizes for your system, you can plan and allocate the file system space that you need for the `/var/adm/crash` directory.

For example, suppose you save partial crash dumps. Your system has 96 MB of memory and you have reserved 85 MB of disk space for crash dumps and swapping. In this case, you should reserve 20 MB of space in the file system for storing crash dump files. You need to reserve considerably more space if you want to save files from more than one crash dump. If you want to save files from multiple crash dumps, consider archiving older crash dump files. See Section 14.6 for information about archiving crash dump files.

By default, the `savecore` command writes crash dump files to the `/var/adm/crash` directory. To reserve space for crash dump files in the default directory, you must mount the `/var/adm/crash` directory on a file system that has a sufficient amount of disk space. (For information about mounting file systems, see Chapter 6 and `mount`(8).) If you expect your crash dump files to be large, you may need to use a Logical Storage Manager (LSM) volume to store crash dump files. For information about creating LSM volumes, see the *Logical Storage Manager* manual.

If your system cannot save crash dump files because of insufficient disk space, the system returns to single-user mode. This return to single-user mode prevents system swapping from corrupting the crash dump. When in single-user mode, you can make space available in the crash directory or change the crash directory. One possibility in this situation is to issue the `savecore` command at the single-user mode prompt. On the command line, specify the name of a directory that contains a sufficient amount of file space to save the crash dump files. For example, the following `savecore` command writes crash dump files to the `/usr/adm/crash2` directory:

```
# savecore /usr/adm/crash2
```

After the `savecore` command has saved the crash dump files, you can bring up your system in multiuser mode, or bring up the network to dump remotely to another host using the `ftp` command.

Specifying a directory on the `savecore` command line changes the crash directory only for the duration of that command. If the system crashes later and the system startup script invokes the `savecore` script, the `savecore` command copies the crash dump to files in the default `/var/adm/crash` directory.

You can control the default location of the crash directory by setting the SAVECORE_DIR variable with the `rcmgr` command. For example, to save crash dump files in the `/usr/adm/crash2` directory by default (at each system startup), issue the following command:

```
# /usr/sbin/rcmgr set SAVECORE_DIR /usr/adm/crash2
```

If you want the system to return to multiuser mode, regardless of whether it saved a crash dump, issue the following command:

```
# /usr/sbin/rcmgr set SAVECORE_FLAGS M
```

14.4 Choosing the Content and Method of Crash Dumps

Crash dumps are compressed and partial by default, but can be full, noncompressed, or both. Normally, partial crash dumps provide the information that you need to determine the cause of a crash. However, you may want the system to generate full crash dumps if you have a recurring crash problem and partial crash dumps are not helpful in finding the cause of the crash.

A partial crash dump contains the following:

- The crash dump header
- A copy of part of physical memory

  The system writes the part of physical memory believed to contain significant information at the time of the system crash, basically kernel node code and data. By default, the system omits user page table entries.

A full crash dump contains the following:

- The crash dump header
- A copy of the entire contents of physical memory at the time of the crash

You can modify how crash dumps are taken:

- By adjusting the crash dump threshold
- By overriding the default so that the system writes user page table entries to partial crash dumps
- By selecting partial or full crash dumps
- By revising the expected dump compression
- By selecting compressed or noncompressed crash dumps

These options are explained in the following sections.

14.4.1 Adjusting the Primary Swap Partition's Crash Dump Threshold

To configure your system so that it writes even small crash dumps to secondary swap partitions before the primary swap partition, use a large value for the `dump_sp_threshold` attribute. The value you assign to this attribute indicates the amount of space that you normally want available for system swapping after a system crash, as described in Section 14.3.

To adjust the `dump_sp_threshold` attribute, issue the `sysconfig` command. For example, suppose your primary swap partition is 40 MB. To raise the value so that the system writes crash dumps to secondary partitions, issue the following command:

```
# sysconfig -r generic dump_sp_threshold=81920
```

In the preceding example, the dump_sp_threshold attribute, which is in the generic subsystem, is set to 81,920 512-byte blocks (40 MB). In this example, the system attempts to leave the entire primary swap partition open for system swapping. The system automatically writes the crash dump to secondary swap partitions and the crash dump header to the end of the primary swap partition.

The sysconfig command changes the value of system attributes for the currently running kernel. To store the new value of the dump_sp_threshold attribute in the sysconfigtab database, modify that database by using the sysconfigdb command. For information about the sysconfigtab database and the sysconfigdb command, see sysconfigdb(8).

Note

After the savecore program has copied the crash dump to a file, all swap devices are immediately available for mounting and swapping. The sharing of swap space only occurs for a short time during boot, and usually on systems with a small amount of physical memory.

14.4.2 Including User Page Tables in Partial Crash Dumps

By default, the system omits user page tables from partial crash dumps. These tables do not normally help you determine the cause of a crash and omitting them reduces the size of crash dumps and crash dump files. However, your technical support person may instruct you to include user page tables for crash dump analysis.

To include user page tables in partial crash dumps, set the value of the dump_user_pte_pages attribute to 1. The dump_user_pte_pages attribute is in the generic subsystem. The following example shows the command you issue to set this attribute:

```
# sysconfig -r generic dump_user_pte_pages = 1
```

The sysconfig command changes the value of system attributes for the currently running kernel. To store the new value of the dump_user_pte_pages attribute in the sysconfigtab database, modify that database by using the sysconfigdb command or use the Kernel Tuner GUI (dxkerneltuner).

To return to the system default of not writing user page tables to partial crash dumps, set the value of the dump_user_pte_pages attribute to 0 (zero).

14.4.3 Selecting Partial or Full Crash Dumps

By default, the system generates partial crash dumps. If you want the system to generate full crash dumps, you can modify the default behavior by setting the kernel's `partial_dump` variable to 0 (zero) as follows:

```
# sysconfig -r generic partial_dump=0 partial_dump: reconfigured
# sysconfig -q generic partial_dump generic: partial_dump = 0
```

You can use the Kernel Tuner GUI or the `sysconfigdb` command to modify kernel entries and preserve the modifications across reboots. To return to partial crash dumps, reset the `partial_dump` variable to 1.

When partial dumps are enabled, you can enable the `dump_kernel_text` attribute to include kernel text pages.

14.4.4 Expected Dump Compression

The `expected_dump_compression` variable is used to signal how much compression you typically expect to achieve in a dump . By default, the value of `expected_dump_compression` is set to 500, the median for a minimum allowed value of 0 (zero) and a maximum value of 1000. The following steps describe how you calculate the appropriate `expected_dump_compression` variable for your system:

1. Create a compressed dump, using the `dumpsys` command, as described in Section 14.5.1. Using the `ls -s` command, record the size of this dump as value `a`.
2. Use the `expand_dump` command to produce a noncompressed version of the dump. Using the `ls -s -s` command, record the size of this dump as value `b`.
3. Divide `a` by `b` to produce the approximate compression ratio.
4. Repeat the previous steps several times and choose the largest value of the compression ratio. Multiply the compression ratio by 1000 to produce an expected dump value.
5. Add 10 percent of the expected dump value to create a value for the `expected_dump_compression` variable.

Set the kernel's `expected_dump_compression` variable to the required value using the `sysconfig` command as follows:

```
# sysconfig -r generic expected_dump_compression=750
expected_dump_compression: reconfigured
# sysconfig -q generic partial_dump
generic:
expected_dump_compression=750
```

You can also use the Kernel Tuner GUI or the sysconfigdb command to modify kernel entries and preserve the modifications across reboots.

14.4.5 Selecting and Using Noncompressed Crash Dumps

By default, crash dumps are compressed to save disk space, allowing you to dump a larger crash dump file to a smaller partition. This can offer significant advantages on systems with a large amount of physical memory, particularly if you want to tune the system to discourage swapping for realtime operations. On reboot after a crash, the savecore command runs automatically and detects that the dump is compressed, using information in the crash dump header in the swap partition. It then copies the crash dump file from the swap partition to the /var/adm/crash directory. The compressed crash dump files are identified by the letter z in the file name, to distinguish them from noncompressed crash dump files. For example: vmzcore.1.

You can use this type of compressed crash dump file with some debugging tools such as dbx, which is not true of the type of compression produced by tools such as compress or gzip. If you need to use a tool that does not support compressed crash dump files, you can convert it to a conventional noncompressed format with the expand_dump utility. The following example shows how you use the expand_dump utility:

```
# expand_dump vmzcore.2 vmcore.2
```

You may want to disable compressed dumps if you always use tools or scripts that do not work with the compressed format, and it is not convenient to use the expand_dump command. To disable compressed dumps, use the following sysconfig command:

```
# sysconfig -r generic compressed_dump=0
```

The preceding command temporarily changes the mode of dumping to noncompressed and the mode reverts to compressed dumps on the next reboot. To make the change persistent, use the sysconfigdb command to update the value of the compressed_dump attribute in the /etc/sysconfigtab file or use the Kernel Tuner GUI to modify the value in the generic subsystem.

Note

Memory dumps are compressed. If the compressed_dump system attribute is not set, the system automatically enables compression before attempting to write a memory dump.

See savecore(8), expand_dump(8), and sysconfig(8) for more information on crash dump compression and how to produce a noncompressed crash dump file.

14.4.6 Dumping to Exempt Memory

Exempt memory is a region of physical memory that is set aside for a specific purpose. You can create an exempt region of memory by specifying it in the /etc/sysconfigtab file. This causes the exempt region to be created when the system boots. For example:

```
cma_dd:
 CMA_Option = Size-0x3000000, Alignment - 0, /
 Addrlimit - 0x4000000, Type - 0x96, Flag-0
```

The preceding /etc/sysconfigtab file entry reserves a region of exempt memory that is 48MB in size. Its Type is specified as M_EXEMPT by the value 0x96, the value of Addrlimit sets the starting position of the exempt region, which at 0x4000000 is 64MB into physical memory. Each time the machine boots, it attempts to reserve this same area of physical memory, making it unavailable for any other use.

Another way of creating exempt regions of memory is by using the contig_malloc() function call with the type M_EXEMPT in a pseudodevice driver. See the malloc.h file for information on the M_EXEMPT type. See contig_malloc(9r) for information on using the function call.

You can use the vmstat command with the -M option to examine exempt memory regions.

To dump to exempt memory, the dump_to_memory attribute must be enabled as described in Section 14.3.2. You also configure the following attributes as required:

dump_exmem_size	Specifies the size (in bytes) of the exempt memory region to which dumps are written. By default, the value is 0 (zero), which disables writing a dump to an exempt memory region.
dump_exmem_addr	Identifies the starting address (virtual or physical) for a region of exempt memory used for writing primary dumps.
dump_exmem_include	Specifies whether or not exempt memory pages are included in the dump. By default, the value is 0 (zero) and exempt memory pages are excluded.

The setting of the dump_exmem_addr attribute has no effect unless you also configure the dump_exmem_size attribute. Ensure that you keep a record of any run-time settings for the attributes so that you are able to find the crash dump after recovery from a system failure.

The following example shows how you reconfigure these attributes:

```
# sysconfig -q generic dump_to_memory
generic:
dump_to_memory = 0
# sysconfig -r generic dump_to_memory=1
dump_to_memory: reconfigured
# sysconfig -q generic dump_to_memory
generic:
dump_to_memory = 1
```

Memory dumps are compressed by default. The compressed_dump system attribute automatically is enabled if it is not set to on. The savecore command uses the vmzcore character special device file to recover the compressed dumps. See savecore(8) and vmzcore(7) for more information.

14.4.7 Dumping to a Remote Host

Use the savecore command with the -r option to write crash dump files from a client host to a remote host using an ftp connection. You can specify either of the following definitions for a remote destination:

- The name of the remote host and a valid account and password
- The path to a configuration file containing the ftp connection and login information

For example, the following command specifies a login to the remote host in verbose mode, which enables you to debug the ftp connection.

```
# savecore -v -r soserv:jeffdump:Cr$hDeBuG
```

When it connects to the target host, the savecore utility directs the remote ftpd server daemon to create a directory named after the client host name. The crash dump files (bounds, msgbuf.savecore, evm.buf, vmunix.N, and vmcore.N or vmzcore.N) are written to the directory. You must ensure that you have adequate space for the crash dump on the remote device.

See savecore(8) and ftpd(8) for more information and for restrictions when using this feature.

14.5 Generating a Crash Dump Manually

The following sections describe how you can create a crash dump file manually under two conditions:

Continuable dump	Use the `dumpsys` command to copy a snapshot of the running memory to a dump file without halting the system. (That is, the system continues to run.)
Forced dump	Use the `crash` console command to cause a crash dump file to be created on a system that is not responding (that is, hung).

It is assumed that you have planned adequate space for the crash dump file and set any kernel parameters as described in the preceding sections.

14.5.1 Continuable Dumps on a Running System

When you cannot halt the system and take a normal crash dump, use the `dumpsys` command to dump a snapshot of memory. Because the system is running while the `dumpsys` command takes a snapshot, memory may change as its content is copied. Analysis of the resulting dump can demonstrate incomplete linked lists and partially zeroed pages, which are not problems, but reflect the transitory state of memory. For this reason, some system problems cannot be detected by using the `dumpsys` command and you may need to halt the system and force a crash dump as described in Section 14.5.2. By default, the `dumpsys` command writes the crash dump in the `/var/adm/crash` directory.

The `/var/adm/crash/minfree` text file specifies the minimum number of kilobytes that must be left on the file system after the `dumpsys` command copies the dump. By default, this file does not exist, indicating that no minimum is set. To specify a minimum, create the file and store the number of kilobytes you want reserved in it. You can override the setting in the `minfree` file by using the `-i` option. The `-s` option displays the approximate number of disk blocks that full and partial dumps require. The exact size can not be determined ahead of time for the following reasons:

- For noncompressed dumps only, the actual dump optimizes disk space by default, suppressing the writing of contiguous zeroes.
- System use of kernel dynamic memory (`malloc`/`free`) changes on the running system.
- The number of indirect disk blocks required to store the dump is unknown.

The following examples show a dump from a system with 512 KB of physical memory. The examples show a noncompressed crash dump. Dumps are usually compressed by default:

```
# dumpsys -s
Approximate full dump size = 1048544 disk blocks,
 if compressed, expect about 524272 disk blocks.
```

```
Approximate partial dump size = 94592 disk blocks,
 if compressed, expect about 47296 disk blocks.

# dumpsys -i /userfiles
Saving 536797184 bytes of image in /userfiles/vmzcore.0
# ls /userfiles
bounds vmzcore.0 vmunix.0
```

Two attributes in the `generic` kernel subsystem enable you to control continuable dumps:

`live_dump_dir_name`	Specifies a path to the directory where the continuable dump files are written. The default value is the `/var/adm/crash` directory.
`live_dump_zero_suppress`	Enables or disables zero compression of continuable dumps. Using this option produces files that take longer to create but occupy less space.

See `dumpsys`(8) and `sys_attrs_generic`(5) for more information. See the *Kernel Debugging* manual for information on analyzing the continuable crash dump.

14.5.2 Forcing Crash Dumps on a Hung System

You can force the system to create a crash dump when the system hangs. On most hardware platforms, you force a crash dump by following these steps:

1. If your system has a switch for enabling and disabling the Halt button, set that switch to the Enable position.
2. Press the Halt button.
3. At the console prompt, enter the `crash` command.

Some systems have no Halt button. In this case, follow these steps to force a crash dump on a hung system:

1. Type Ctrl/p at the console prompt.
2. At the console prompt, enter the `crash` command.

If your system hangs and you force a crash dump, the panic string recorded in the crash dump is the following:

```
hardware restart
```

This panic string is always the one recorded when system operation is interrupted by pressing the Halt button or by typing Ctrl/p.

14.6 Storing and Archiving Crash Dump Files

If you are working entirely with compressed (vmzcore.*n*) crash dump files, they should be compressed for efficient archiving. The following sections discuss certain special cases.

Section 14.6.1 describes how to compress files for storage or transmission if:

- You are working with uncompressed (vmcore.*n*) crash dump files.
- You need the maximum amount of compression possible — for example, if you need to transmit a crash dump file over a slow transmission line.

Section 14.6.2 describes how to uncompress partial crash dump files that are compressed from vmcore.*n* files.

14.6.1 Compressing a Crash Dump File

To compress a vmcore.*n* crash dump file, use a utility such as gzip, compress, or dxarchiver. For example, the following command creates a compressed file named vmcore.3.gz:

```
# gzip vmcore.3
```

A vmzcore.*n* crash dump file uses a special compression method that makes it readable by debuggers and crash analysis tools without requiring decompression. A vmzcore.*n* file is compressed substantially compared to the equivalent vmcore.*n* file, but not as much as if the vmcore.*n* file is compressed using a standard UNIX compression utility, such as gzip. Standard compression applied to a vmzcore.*n* file makes the resulting file about 40 percent smaller than the equivalent vmzcore.*n* file.

If you need to apply the maximum compression possible to a vmzcore.*n* file, follow these steps:

1. Uncompress the vmzcore.*n* file by using the expand_dump command; see expand_dump(8) for more information. The following example creates an uncompressed file named vmcore.3 from the vmzcore.3 file:

   ```
   # expand_dump vmzcore.3
   ```

2. Compress the resulting vmcore.*n* file using a standard UNIX utility. The following example uses the gzip command to create a compressed file named vmcore.3.gz:

   ```
   # gzip vmcore.3
   ```

You can uncompress a vmzcore.*n* file only with the expand_dump command. (Do not use gunzip, uncompress, or any other utility). After you uncompress a vmzcore.*n* file into a vmcore.*n* file by using the expand_dump command, you cannot compress it back into a vmzcore.*n* file.

14.6.2 Uncompressing a Partial Crash Dump File

This section applies only if you are uncompressing a partial crash dump file that was previously compressed from a vmcore.*n* file.

If you compress a vmcore.*n* dump file from a partial crash dump, you must use care when you uncompress it. Using the gunzip or uncompress command with no options results in a vmcore.*n* file that requires space equal to the size of memory. In other words, the uncompressed file requires the same amount of disk space as a vmcore.*n* file from a full crash dump.

This situation occurs because the original vmcore.*n* file contains UNIX File System (UFS) file holes. (UFS files can contain regions, called holes, which have no associated data blocks.) When a process, such as the gunzip or uncompress command, reads from a hole in a file, the file system returns zero-valued data. Thus, memory omitted from the partial dump is added back into the uncompressed vmcore.*n* file as disk blocks containing all zeros.

To ensure that the uncompressed core file remains at its partial dump size, you must pipe the output from the gunzip or uncompress command with the -c option to the dd command with the conv=sparse option. For example, to uncompress a file named vmcore.0.Z, issue the following command:

```
# uncompress -c vmcore.0.Z | dd of=vmcore.0 conv=sparse

262144+0 records in

262144+0 records out
```

A

Administration Utilities

This appendix identifies and defines the administrative utilities and commands.

A.1 X11 Graphical User Interfaces (CDE Application Manager)

The X11–based graphical user utilities (GUIs) are available under the CDE Application Manager or from the command line. In some cases, the GUIs have analogous SysMan utilities or are superseded by a SysMan Menu task. Invoke the CDE applications as described in Chapter 1.

Not all administrative tasks are available as SysMan Menu options. You need to use a combination of GUIs, SysMan Menu tasks, and commands. With each release, more SysMan Menu options are added and older administrative methods become obsolete. To help you understand your administrative options, Table A–1 to Table A–6 list the utilities. Each table provides the following information:

- The first column identifies the task, which can be a subsystem that you want to configure or an administrative application that you want to run on a system component, such as a disk or a file system. There are three application formats that may appear in this column:
  - X11–compliant Graphical User Interfaces (GUIs), such as the Kernel Tuner (`dxkerneltuner`).
  - SysMan Menu utilities that you can run in different user environments.
  - Command line scripts. You must install the `OSFRETIRED`*XXX* subsets to access some of these scripts, as described in the *Installation Guide*.
- The second column lists the SysMan Menu task. Text in brackets, such as `[dns]`, are the command options that you can use with the `sysman` command to invoke a utility directly from the command prompt. For example:

```
# sysman dns
# sysman dns_client
```

The first command example invokes the submenu of all DNS tasks. The second command invokes the specific utility that you can use to configure the local system as a DNS client.

- The third column lists the commands (command line options) that perform the equivalent task.

Many SysMan Menu tasks do not have an analogous CDE GUI. The command line provides most functions, but it has limitations. For example, you cannot select and apply changes to multiple user accounts by using the `usermod` command, but you can do it by using the Account Manager GUI (`dxaccounts`).

The following tables are organized by CDE application. The Application Manager provides the following folders containing system administration utilities:

Configuration	Table A–1 lists the configuration utilities, which are used for initial system configuration and regular system maintenance.
Daily Admin	Table A–2 lists the daily administration utilities, which are used for routine system administration tasks.
MonitoringTuning	Table A–3 lists the utilities that are used for monitoring system operation and performance tuning.
Software_Management	Table A–4 lists the utilities that are used for installation software management.
Storage_Management	Table A–5 lists the utilities that are used for administering file systems and storage.
Tools	Table A–6 lists the utilities that provide system statistics.

Table A–1: System Administration Configuration Applications

Subsystem to Configure	SysMan Menu Option	Command Line Interface
ATM (atmsetup(8))	Set up Asynchronous Transfer Mode (ATM) [atm]	atmconfig(8)
Audit Configuration	Audit Configuration [auditconfig]	none
CDE Setup (dtsetup(8))	none	none
DHCP Server (xjoin(8), DHCP(7))	Network Setup Wizard [net_wizard] Set up System as DHCP Server [joind]	none
Disk (diskconfig(8))	Some file system tasks performed by diskconfig can be found under the Storage options.	disklabel(8), newfs(8)
DNS (BIND) (bindconfig(8))	Domain Name Service (DNS (BIND)) [dns, dns_client, dns_server, dns_deconfigure]	Retired in Version 5.1
DOP (dop(8))	Configure Division of Privileges (DOP) [dopconfig]	dop(8)
Dump	Configure Dump [dumpconfig]	none
IPsec	Configure Internet Protocol Security [IPsec]	none
latsetup	Configure Local Area Transport (LAT) [lat]	latsetup(8), lat_manual_setup(7)
LDAP	Set up LDAP Configuration [ldap]	ldapadd(8), ldapdelete(8), ldapmodify(8)
Mail (mailconfig(8))	Configure mail [mailsetup]	mailsetup(8)
NFS (nfsconfig(8))	Network File System (NFS) [nfs]	Retired in Version 5.1
NIS (nissetup(8))	Configure Network Information Service (NIS) [nis]	none
NTP	Network Time Protocol (NTP) [ntp, ntp_config, ntp_status, ntp_start, ntpstop]	Retired in Version 5.1
Network	Network Setup Wizard [net_wizard]	none
PPP (pppd(8))	Point-to-Point Protocol (PPP) [ppp]	none

Table A–1: System Administration Configuration Applications (cont.)

Subsystem to Configure	SysMan Menu Option	Command Line Interface
Print (printconfig(8))	Configure Line Printers [lprsetup]	lprsetup(8)
Security	Security Configuration [secconfig]	none
SLIP	Serial Line Networking [serial_line]	startslip(8)

Table A–2: System Administration Daily Admin Applications

CDE Administrative Task	SysMan Menu Option	Command Line Interface
Account Manager (dxaccounts(8)). See also features offered under Advanced Server for UNIX (ASU)	Accounts [accounts, users, groups, nis_users, nis_groups, ldap_users, ldap_groups].	useradd(8), usermod(8), userdel(8), groupadd(8), groupmod(8), groupdel(8)
Archiver (dxarchiver(8))	none	tar(1), pax(1), cpio(1)
Audit Manager (dxaudit(8))	none	none
Display Window (dxdw(8))	none	iostat(1), netstat(1), vmstat(1), who(1)
Event Viewer	View Events [event_viewer]	evmget(1), evmshow(1), evmpost(1), and other associated commands, See EVM(5)
File Sharing (dxfileshare(8))	Share Local Directory (etc/exports) [export]	mount(8), automount(8), exports(4)
Get/Set ACL (dxsetacl(8))	none	none
Host Manager (dxhosts(8))	none	none
License Manager (dxlicenses(8))	Register License Data [lmfsetup]	lmf(8), lmfsetup(8)
Mail User Admin (mailusradm(8))	none	none

Table A–2: System Administration Daily Admin Applications (cont.)

CDE Administrative Task	SysMan Menu Option	Command Line Interface
Power Management (dxpower(8))	none	sysconfig(8)
SysManShutdown	Shut Down the System [shutdown]	shutdown(8)
System Information (dxsysinfo(8))	none	du(1), df(1), swapon(8)

Table A–3: System Administration Monitoring and Tuning Applications

CDE Administrative Task	SysMan Menu Option	Command Line Interface
Class Scheduler	Class Scheduling [class_sched, class_setup, class_start, class_stop]	class_admin(8)
HP Insight Manager	Set Up HP Insight Manager [imconfig]	none
Configuration Report	Create Configuration Report [config_report]	sys_check(8)
Escalation Report	Create Escalation Report [escalation]	sys_check(8)
Kernel Tuner (dxkerneltuner(8))	none	sysconfig(8), sysconfigdb(8)
Process Tuner (dxproctuner(8))	none	nice(1), renice(8), ps(1), kill(1)

Table A–4: System Administration Software Management Applications

CDE Administrative Task	SysMan Menu Option	Command Line Interface
Software Management	Installation, [install, setldload, setldlist, setldd].	setld(8)
Update Installation Cleanup	Cleanup After an OS Update (updadmin), [updadmin]	updadmin(8)

Table A–5: System Administration Storage Management Applications

CDE Administrative Task	SysMan Menu Option	Command Line Interface
Advanced File System (dtadvfs(8))	Advanced File System (AdvFS) Utilities [advfs]	See advfs(4)
Bootable Tape	Create a Bootable Tape [boot_tape]	btcreate(8), btextract(8)
File System Mgmt	File Systems Management Utilities [filesystems]	mount(8), newfs(8), fstab(4)
Logical Storage Manager (lsmsa(8), dxlsm(8X))	Logical Storage Manager (LSM) Utilities [lsm, volsetup].	volsetup(8)
Prestoserve I/O Accelerator (dxpresto(8X))	Configure the Prestoserve software [presto]	prestosetup(8))

Table A–6: System Administration Tools

CDE Administrative Task	SysMan Menu Option	Command Line Interface
I/O Statistics (dxdw(8))	View Input/Output (I/O) statistics [iostat]	iostat(1)
Network Statistics (dxdw(8))	none	netstat(1)
System Messages (dxdw(8))	See the Event Viewer	syslogd(8)
Virtual Memory Statistics (dxdw(8))	View Virtual Memory (VM) Statistics [vmstat]	vmstat(1)

A.2 SysMan Menu Tasks and Associated Utilities

The tables in Section A.1 identify the GUIs and command line utilities that perform functions similar to the various SysMan Menu tasks. The following SysMan Menu utilities are available. You can use the accelerator keyword, such as [accounts], with the sysman command to launch a utility from the command line. The following is a list of the initial categories of the SysMan Menu utilities; each is discussed in its own section.

- Accounts [accounts]
- Hardware [hardware]
- Mail [mail]
- Monitoring and Tuning [monitoring]

- Networking [network]
- Printing [printers]
- Security [security]
- Software [software]
- Storage [storage]
- Support and Services [support]
- General Tasks [general_tasks]

A.2.1 Accounts

Accounts [accounts] enables you to maintain user accounts and manage system resources. See Chapter 7 for information on administering user accounts. The Accounts tasks provided are as follows:

Manage Local Users [users]

Administer the `/etc/passwd` file, which records user accounts data

Manage Local Groups [groups]

Administer the `/etc/group` file, which records user resource access data

Manage NIS Users [nis_users]

Administer NIS user accounts

Manage NIS Groups [nis_groups]

Administer NIS user groups

Manage LDAP Users [ldap_users]

Administer LDAP user accounts

Manage LDAP Groups [ldap_groups]

Administer LDAP user groups

A.2.2 Hardware

Hardware [hardware] enables you to display information about system hardware and peripheral devices. See the *Hardware Management* manual for information on administering user accounts. The hardware tasks provided are as follows:

View Hardware Hierarchy [hw_hierarchy]

Displays all the system components as a hierarchy. For example, the CPU and all devices attached to the buses. See `hwmgr`(8) for more information.

View Cluster [hw_cluhierarchy]

Displays the hierarchy of all members of a cluster.

View Device Information [hw_devices]

Displays a list of all devices (such as disks) attached to the system. See `hwmgr`(8) for more information.

View central processing unit (CPU) information [hw_cpus]

Displays the type of processors on the system and their status, such as time on line.

Manage CPUs [hw_manage_cpus]

Manage CPUs on multiprocessor systems.

Online Addition/Replacement (OLAR) policy information [hw_olar_policy_info]

Administer the policy information for removal and addition of components. See `olar_config`(4) for more information.

A.2.3 Mail

Mail [mail] enables you to configure e-mail and manage mail accounts. The mail tasks provided are as follows:

Configure Mail [mailsetup]

Enables you to configure the electronic mail services on the system.

Manage Users' Mail Accounts [mailusradm]

Enables you to set up e-mail for system account holders.

A.2.4 Monitoring and Tuning

Monitoring and Tuning [monitoring] enables you to configure and use system event-reporting and tuning utilities. This task provides the following utilities:

View Events [event_viewer]

Enables you to invoke the Event Manager viewer. See EVM(5) and Chapter 13 for more information.

Set Up HP Insight Manager [imconfig]

Enables you to configure HP Insight Manager. See Chapter 1 for more information.

Class Scheduling [class_sched]

Enables you to allocate CPU time resources to groups of processes. See class_admin(8) and Chapter 3 for more information. This task provides the following utilities:

Configure Class Scheduler [class_setup]

Enables you to create scheduling databases that govern the use of system resources (such as CPU time) by processes. You can set a current schedule also.

[Re]Start Class Scheduler [class_start]

Starts the scheduling daemon to implement the currently-selected schedule.

Stop Class Scheduler [class_stop]

Stops the scheduling daemon and turns off resource sharing.

View Virtual Memory (VM) Statistics [vmstat]

Enables you to monitor virtual memory statistics. See vmstat(1) for more information.

View Input/Output (I/O) Statistics [iostat]

Enables you to monitor I/O (input/output) statistics. See iostat(1) for more information.

View Uptime Statistics [uptime]

Enables you to monitor how long the system has been up, and determine the average workload since the last boot. See uptime(1) for more information.

A.2.5 Networking

Networking [network] enables you to set up and administer network resources. The utilities provided are as follows:

Network Setup Wizard [net_wizard]

A utility that guides you through the steps of setting up the network environment. See Chapter 1 for a brief overview. See the *Network Administration: Services* manual for more information on networking configuration options.

Basic Networking Services [networkbasic]

A set of tasks you can perform to configure the most commonly used individual networking features. This task provides the following utilities:

Set up Asynchronous Transfer Mode (ATM) [atm]

Set up ATM services and configure ATM adapters. See the *Network Administration: Connections* manual for more information.

Set up Network Interface Cards [interface]

Enables you to configure network devices, providing information such as the TCP/IP address and network mask.

Set up static routes (`/etc/routes`) [route]

Enables you to set up the network to use static routes, and defines a router node. Static routes are the most common form of communication with local and remote networks. See the *Network Administration: Services* manual for more information.

Set up routing services (`gated`, `routed`, IP router) [routing]

Enables you to configure the network to use a particular method of routing. Your options are Gateway Routing Daemon (`gated`), Routing Daemon (`routed`), or an Internet Protocol (IP) Router. See the *Network Administration: Services* manual for more information.

Set up hosts file (`/etc/hosts`) [host]

Enables you to add remote host systems to the `/etc/hosts` file. This makes the hosts known to the local system so that network communication can be established.

Set up hosts equivalency file (`/etc/hosts.equiv`) [hosteq]

Enables you to add remote host systems and users to the `/etc/hosts.equiv` file. This enables users on remote hosts to use resources on the local system. (See the *Security*

Administration manual for information on security risks associated with host equivalency).

Set up remote who services (`rwhod`) [rwhod]

Enables you to obtain information on users of the local network.

Set up the networks file (`/etc/networks`) [networks]

Enables you to specify networks known to the local system.

Additional Network Services [networkadditional]

A set of utilities you can use to configure other networking features. This task provides the following utilities:

Domain Name Service (DNS (BIND)) [dns]

Configure domain name services on the local system. The following utilities are provided:

- Configure system as DNS server [dns_server]
- Configure system as DNS client [dns_client]
- Deconfigure DNS on this system [dns_deconfigure]

Serial Line Networking [serial_line]

Enables you to configure the following serial-line networking options:

Point-to-Point Protocol (PPP) [ppp]

Enables you to configure PPP, including the following tasks:

- Create option files [ppp_options]
- Modify pap-secrets file [pap]
- Modify chap-secrets file [chap]

See the *Network Administration: Connections* manual for more information.

Configure system for UNIX to UNIX copy (`uucp`) connections [uucp]

Enables you to configure UUCP over a modem, TCP/IP, or hardwired connection. See the *Network Administration: Services* manual for more information.

Network Time Protocol (NTP) [ntp]

Enables the automatic regulation on the system's internal clock by comparing time values with a server, or to act as a time server to client

systems. See the *Network Administration: Services* manual for more information. The following utilities are available:

- Configure system as an NTP client [ntp_config]
- View status of NTP daemon [ntp_status]
- [Re]start NTP daemon [ntp_start]
- Stop NTP daemon [ntp_stop]

Configure Internet Protocol Security(IPsec) [ipsec]

Enables you t configure and manage Internet Protocol Security on a system.

Network File System (NFS) [nfs]

Enables you to configure Network File System, and share file systems between hosts. See the *Network Administration: Services* manual for more information. This task provides the following utilities:

- View NFS configuration status [nfs_config_status]
- Configure system as an NFS client [nfs_client]
- Deconfigure system as an NFS client [nfs_deconfig_client]
- Configure system as an NFS server [nfs_server]
- Deconfigure system as an NFS server [nfs_deconfig_server]
- View NFS daemon status [nfs_daemon_status]
- Start/Restart NFS daemons [nfs_start]
- Stop NFS daemons [nfs_stop]

Configure Network Information Service NIS [nis]

Enables you to allow users to use the resources of networked systems, such as logging in to different hosts. User names and passwords are distributed between hosts. See the *Network Administration: Services* manual for more information.

Configure Local Area Transport (LAT) [lat]

Enables you to set up LAT. See the *Network Administration: Connections* manual for more information.

Set up the system as a DHCP Server (joind) [joind]

Enables you to set up DHCP. See the *Network Administration: Connections* manual for more information.

View network daemon status [dmnstatus]

Enables you to verify the status of the various network daemons such as `gated` or `rwhod`.

Start or Restart network services [inet_start]

Enables you to start or restart any stopped networking daemons such as `gated` or `rwhod`.

Stop network services [inet_stop]

Enables you to stop all network services.

A.2.6 Printing

Printing [printers] enables you to configure system print facilities. This task invokes the following utility:

Configure line printers [lprsetup]

Enables you to add local and remote (networked) print devices to the list of available devices, and make these resources available to users.

A.2.7 Security

Security [security] enables you to administer system security, system auditing, and privileged user access to administrative utilities. See the *Security Administration* manual for more information. The following utilities are provided:

Configure Division of Privileges (DOP) [dopconfig]

Enables you to give any user full access to privileged programs such as SysMan Menu tasks. See `dop`(8) for more information.

Manage DOP Actions [dopaction]

Enables you to create, modify, or delete DOP actions and their associated privileges.

Security Configuration [secconfig]

Enables you to configure base or enhanced security.

Audit Configuration [auditconfig]

Enables you to set up and start security auditing.

A.2.8 Software

Software [software] enables you to manage operating system and layered software installations and updates. This task provides the following utilities:

Installation [install]

Enables installation of the operating system and components. This task provides the following utilities:

Install software [setldload]

Enables you to add software to the system from a RIS server or from the distribution media (CD–ROM).

List installed software [setldlist]

Enables you to list the software that is currently installed on the system.

Remove installed software [setldd]

Enables you to permanently remove software from the system.

Clean up after an OS update [updadmin]

Enables you to remove unnecessary files from the system to save space or archive files to tape after running an installation update (`updateinstall`).

Register license data [lmfsetup]

Enables you to register software product authorization keys (PAKs).

A.2.9 Storage

Storage [storage] enables you to administer file systems and data storage. The following utilities are provided:

File Systems Management Utilities [filesystems]

Enables basic administration of disk storage. See Chapter 6 for more information. The following utilities are provided:

General File System Utilities [generalfs]

Provides utilities that you can use with either UFS or AdvFS. This task provides the following utilities:

- Dismount a File System [dismount]
- Display Currently Mounted File Systems [df]
- Mount File Systems [mount]

- Share Local Directory (`/etc/exports`) [export]
- Mount Network Directory (`/etc/fstab`) [net_mount]

Advanced File System Utilities [advfs]

Enables you to perform basic administration tasks on AdvFS domains. See `advfs`(4) and the *AdvFS Administration* manual for more information. The following utilities are provided:

- Manage an AdvFS Domain [domain_manager]
- Manage an AdvFS File [file_manager]
- Defragment an AdvFS Domain [defrag]
- Create a New AdvFS Domain [mkfdmn]
- Create a New AdvFS Fileset [mkfset]
- Recover Files from an AdvFS Domain [salvage]
- Repair an AdvFS Domain [verify]

UNIX File System (UFS) Utilities [ufs]

Enables you to perform basic administration tasks on UFS. See Chapter 6 for more information on administering UFS. The following utility is provided:

Create a New UFS File System [newfs]

Enables you to write a new file system to a disk partition.

Logical Storage Manager (LSM) Administration [lsm]

Enables you to perform basic administration of Logical Storage Manager (LSM) volumes. See the *Logical Storage Manager* manual for more information. The following utilities are provided:

- Initialize the Logical Storage Manager (LSM) [volsetup]
- Logical Storage Manager (LSM) Administrator [lsmmgr]

Create a Bootable Tape [boot_tape]

Enables you to create a standalone kernel on a bootable tape, which can assist in disaster recovery. See the `btcreate`(8), `btextract`(8) and Chapter 9 for more information.

Identify SAN Appliances Wizard [idsanappl]

Enables you to identify storage area network (SAN) management appliances and add them to the SysMan Station.

A.2.10 Support and Services

Support and Services [support] enables you to run preconfigured `sys_check` system census tasks as part of troubleshooting and error recovery, or in case you need to escalate a problem to your technical support organization. See Chapter 3 and `sys_check`(8) for information. The following utilities are available:

Configure Dump [dumpconfig]

Enables you to to configure the properties of a crash dump.

Create Dump Snapshot [onlinedump]

Enables you to generate a crash dump file manually

Create Escalation Report [escalation]

Enables you to prepare a system census report for delivery to your technical support organization.

Create Configuration Report [config_report]

Enables you to prepare a system census report for baseline, troubleshooting, or tuning purposes.

A.2.11 General Tasks

General Tasks [general_tasks] provides you with a set of miscellaneous administrative utilities as follows:

Shut down the System [shutdown]

Enables you to perform managed shutdowns of the system. See `shutdown`(8) and Chapter 2 for more information.

Quick Setup [quicksetup]

Runs the basic system setup wizard, which guides you through typical basic system configuration. See Chapter 1 for a description of the features.

Configure Prestoserve software [presto]

Enables you to configure Prestoserve. See the *Guide to Prestoserve* manual for more information.

Configure X Display Manager [xsetup]

Enables you to select CDE or XDM as the default windowing environment.

Cloning Setup Information [cloneinfo]

Displays information on using the `sysman -clone` command to clone your system's configuration and apply it to other systems. See the *Installation Guide — Advanced Topics* manual for information on cloning systems.

Command Line Interface Information [sysmancli]

Displays information on using the `sysman -cli`, a command line interface for running SysMan tasks from the system prompt, or for shell programming.

Set up LDAP Configuration [ldap]

Enables you to configure Lightweight Directory Access Protocol.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y