Tru64 UNIX

System Administration

Part Number: AA-RH9FB-TE

April 2000

Product Version: Tru64 UNIXVersion 5.0A or higher

This guide describes the tasks you must perform to maintain a Compaq Tru64™ UNIX® (formerly known as Digital 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 guide.

Compaq Computer Corporation
Houston, Texas

Contents

About This Guide

1 System Administration Methods and Utilities

2 Starting Up and Shutting Down the System

3 Customizing the System Environment

4 Configuring the Kernel

5 Hardware Management

6 Administering File Systems

8 Administering the Print Services

9 Administering the Archiving Services

10 Administering the System Accounting Services

11 Monitoring and Testing the System

12 Administering the Basic System Event Channels

13 Event Management Using EVM

14 Administering Crash Dumps

Index

Examples

Figures

Tables

About This Guide

This manual describes the tasks you perform to administer the operating system running on a workstation or server.

Audience

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

New and Changed Features

This revision of the manual documents the following new features, changed features, and retiring interfaces.

SysMan Menu and SysMan Station

The SysMan Menu portable interface has been enhanced, and is the preferred method of performing administrative tasks where possible. The SysMan Station is the preferred method of monitoring systems and of launching administrative applications. The following legacy administrative tool environments still exist:

- The System Admin tools folder in the CDE Application Manager. This folder contains some X11-compliant graphical user interfaces (GUIs). However, some icons that formerly launched GUIs now default to SysMan Menu tasks. In future releases, the X11-compliant graphical applications will be moved to optional subsets.
- The System Setup configuration checklist (formerly the SysMan Checklist), which is displayed during the first login to a newly installed system. This interface enables you to quickly set up the system, including networking.
- Administrative commands, utilities and scripts that you run from a terminal command prompt. However, some scripts have been moved to obsolete software subsets and you may need to load the subsets.

The SysMan Menu provides a framework for organizing various system management tasks. The tasks are arranged hierarchically. Each task represents a utility that you launch directly from the menu. This interface allows you to run the tasks from any of the following user environments:

- Any X11–compliant windowing environment such as CDE.
- Personal computers running the Microsoft Windows, Windows NT, and Windows 2000 user environments.
- Any local or remote character cell terminal, including terminal windows on PCs.
- A Web browser such as Internet Explorer or Netscape.

The SysMan Menu also provides a command-line interface to its applications, enabling you to use it within shell scripts or perform operations directly from the command line. The SysMan Station provides a graphical topological representation (map) of the system and all its components. It allows you to visually select system components and then launch administrative applications that apply to that component. See *Chapter 1* for general information about these features and their associated commands and utilities.

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 resources are available:

- You can review all documentation for all releases at the following web site: http://www.unix.digital.com/faqs/publications/pub_page/pubs_page.html
- You can view the Technical Updates for any additional information not included in the documentation provided with your media. Access the Technical Updates from the following web site:http://www.unix.digital.com/faqs/publications/pub_page/update_list.html

New Information

This manual includes the following new information:

- *Chapter 7* has been updated to include information on Windows 2000 Single Sign On.
- *Chapter 11* is a new chapter, created with existing information about monitoring tools and test tools which was relocated from other chapters. Information on the `collect` utility has been added, and information on the `sys_check` utility has been enhanced. New versions of several utilities will ship in this release.

- *Chapter 13* has been updated to include information on using a new utility to translate binary events.
- *Chapter 14* is a new chapter documenting crash dump administration. It contains unchanged information relocated from other chapters.

Changed or Removed Information

The following chapters have been revised to document new features and to correct documentation errors:

- *Chapter 3* has been amended to relocate monitoring information to a new chapter. The information on the Performance Monitor has been removed.
- *Chapter 12* has been amended to relocate crash dump and testing information to new chapters.

  Appendix D has been amended to relocate some obsolete device information to reference pages.

Unchanged Information

With the exception of minor corrections, the information in the remaining chapters and appendixes has not changed since the last revision.

Organization

This guide consists of the following chapters and appendixes:

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. Additionally, 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 dynamically and statically configure an operating system kernel.
Chapter 5	Describes hardware and device administration and device naming, how you administer storage devices (principally disk devices), and tasks such as adding swap space.
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.
Appendix B	Lists the device mnemonics.
Appendix C	Contains information about the CI bus and the Hierarchical Storage Controller (HSC) configuration.
Appendix D	Contains information on specific hardware devices.

Related Documents

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

- The Owner's Manual for your system and for any peripheral device connected to the system.
- The *Installation Guide* and *Installation Guide — Advanced Topics* describes how to install your operating system. Several important administrative tasks are described in detail in theses guides, such as installing software and installation cloning.
- The *Network Administration* manual describes how to set up, configure, and troubleshoot your network.
- The *Concepts and Planning Guide* and *Installation and Administration Guide* provide information on administering Windows domain accounts and sharing printers with PC users.
- The *Security* guide provides information on security that affects account management and file system sharing.
- The *AdvFS Administration* and *Logical Storage Manager* guides provide information on advanced file systems and storage management.

- The *System Configuration and Tuning* guide provides information on system performance tuning and advanced kernel configuration.

Icons on Tru64 UNIX Printed Books

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

G	Books for general users
S	Books for system and network administrators
P	Books for programmers
D	Books for device driver writers
R	Books for reference page users

Some books in the documentation help meet the needs of several audiences. For example, the information in some system books 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 books in the Tru64 UNIX documentation set.

Reader's Comments

Compaq 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`
- Mail:

  Compaq Computer Corporation
  UBPG Publications Manager
  ZKO3-3/Y32
  110 Spit Brook Road
  Nashua, NH 03062-2698

  A Reader's Comment form is located in the back of each printed manual. The form is postage paid if you mail it in the United States.

Please include the following information along with your comments:

- The full title of the book and the order number. (The order number is printed on the title page of this book and on its back cover.)
- 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 Compaq technical support office. Information provided with the software media explains how to send problem reports to Compaq.

Conventions

This guide uses the following conventions:

MB1, MB2, MB3	MB*N* refers to the mouse button that you must press when selecting an item or initiating 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. The following sections are included in this chapter:

- Section 1.1 provides an overview of administrative methods and utilities.
- Section 1.2 provides pointers to other documentation available for the administrative utilities, such as online and Web-based help.
- Section 1.3 explains the system setup utilities that are automatically displayed during the first root login to a system (after a full installation).
- Section 1.4 introduces the different administrative methods and utilities.
- Section 1.5 describes the administrative utilities that you launch from the Common Desktop. Environment (CDE)
- Section 1.6 explains how to use the SysMan Menu.
- Section 1.7 explains how to use the SysMan Menu command-line interface.
- Section 1.8 explains how to use the SysMan Station.
- Section 1.9 explains how to set up Insight Manager, to view system status and launch the SysMan Menu and the SysMan Station from a Web browser.
- Section 1.10 explains how to configure the SysMan Menu and the SysMan Station clients so that you can launch them directly from Windows on a PC.
- Section 1.11 explains how to set up a serial line console to access a remote system using a modem line.

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 an IBM-compatible Personal Computer (PC), such as Windows 98 and Windows NT
- Web-based management, using a Web browser such as Internet Explorer or Insight Manager.
- A terminal, or terminal window running under any of the above 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`), used to administer user accounts and groups.
- Kernel Tuner (`dxkerneltuner`), used to customize your UNIX kernel.
- File Sharing (`dxfileshare`), used 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 will 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 may also 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 will display the shutdown delay selection as a slider bar. You use the mouse button to click on this bar and drag it to select 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 will 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 can also connect to any system using 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. This section provides 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 – Each utility has its own reference page that describes how to invoke the utility, and its available options. For example, the `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.

- Online help – Each utility provides an online help volume that describes how you use it and gives a detailed description of the available option 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. Refer to the `curses`(3) reference page for more information on `curses`

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

- Web Browser-based help – When you configure and invoke the Netscape viewer as described in the *Installation Guide* the home page defaults to `file:/user/doc/netscape/Digital_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/SYS-MAN/index.html`, The Insight Manager Web-based Management environment. 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.

Note

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 Compaq 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 Insight daemon to display online help.

– Links to product information on the World Wide Web.

When Insight Manager is configured, you can also connect to the Insight Manager Web agents of any system in the local network domain that is running the 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 Insight Manager.

Choose Insight Manager Agents and then click on “Summary?” to access the Insight Manager Web-based user guide.

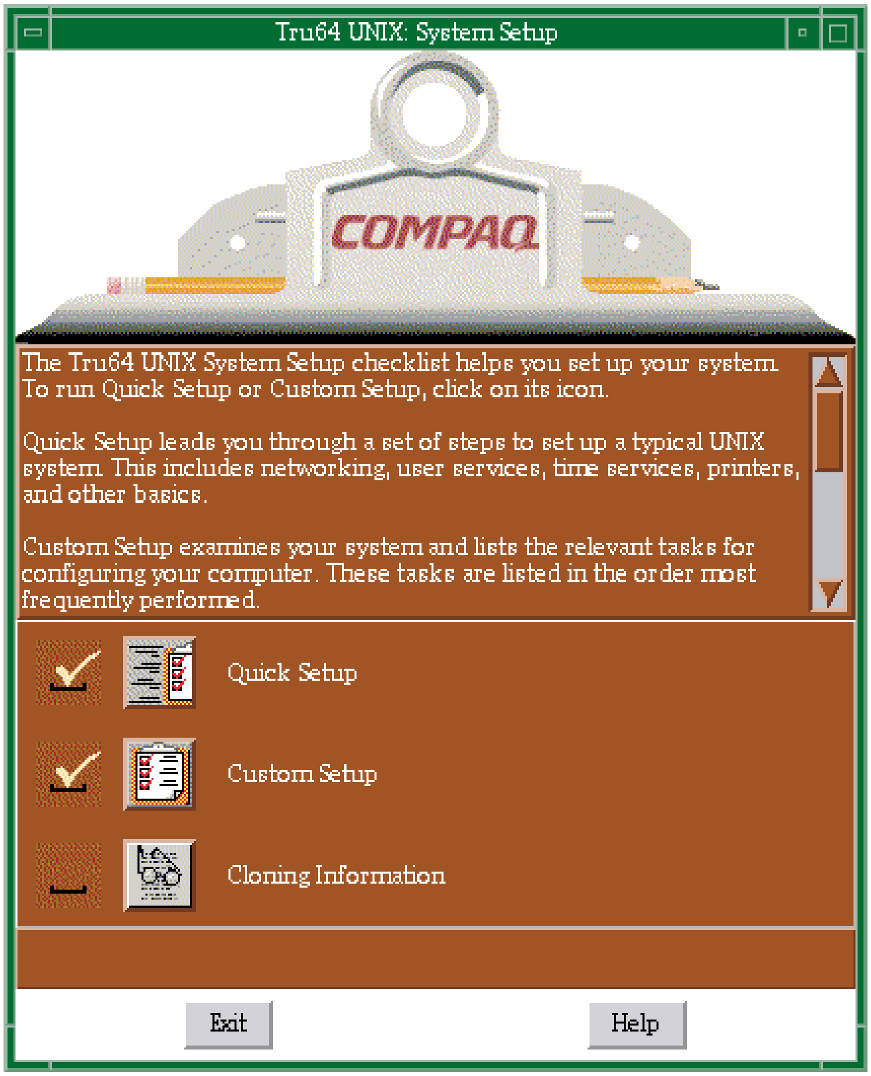
Note that there are restrictions on using Insight Manager, depending on what user environment you are using. See Section 1.9 for information.

1.3 Setting Up Your System

The initial configuration of your system (setup) is usually performed as a post-installation task and System Setup is invoked automatically at first root (superuser) login after an installation. During installation, you may have already 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 is presented as a graphical user interface (the clipboard) 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.

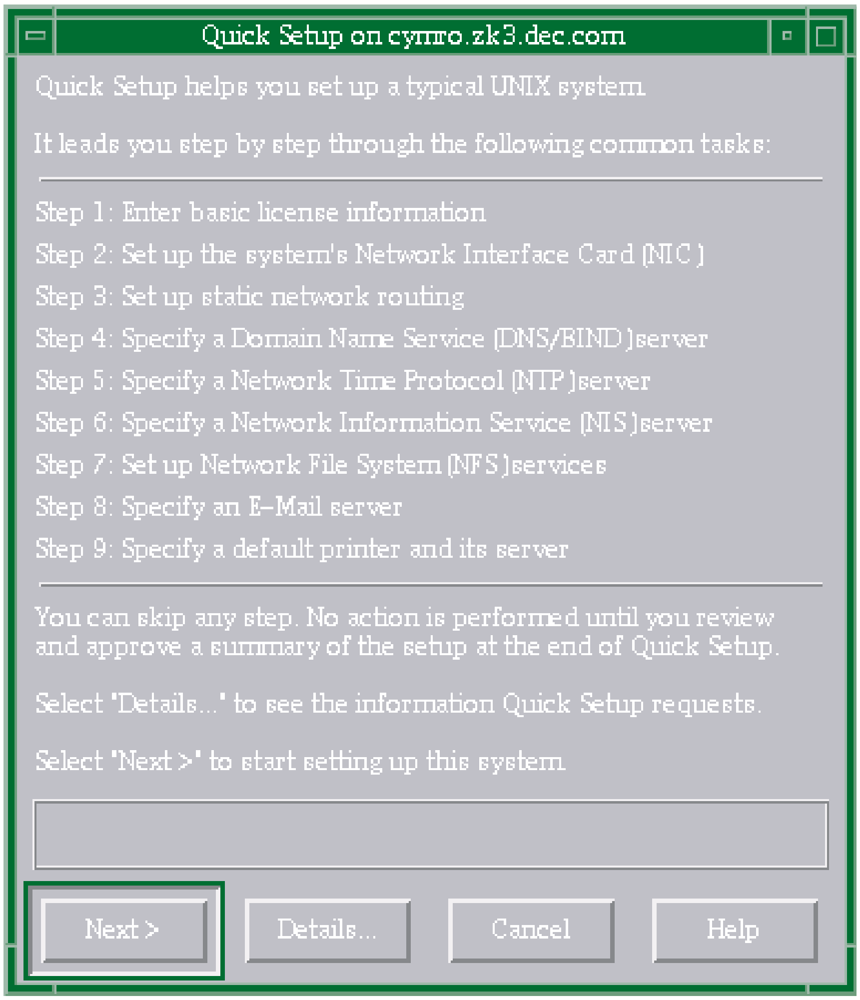
Figure 1–1: System Setup Graphical Interface



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 options 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* and the `sysman_clone`(8) reference page for more information.

Refer to Section 1.5.2 for more information and refer to the `setup`(8) reference page for a list of options.

1.4 Administrative Methods

Most of the tasks described in this book 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 as follows:

– From any local or remote character-cell terminal

– From any X11-compliant windowing environment, such as CDE

– From Microsoft Windows on a personal computer (PC)

– From the Web browser Insight Manager page

More information is provided in Section 1.6.

• The SysMan Station

The SysMan Station is a graphical representation of a system (or cluster) that enables you monitor system status from the CPU down to the level of individual system components such as disks. You can also view and monitor logical groups such as file systems or AdvFS domains and create customized views. When viewing any system component, you can obtained 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.

More information is provided in Section 1.8.

• 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`.

More information is provided in Section 1.5.

• 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 - DailyAdmin in the CDE environment, or by invoking dxaccounts from a terminal window. (The interface will run 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, such as netsetup to the appropriate SysMan Menu method. These command-line utilities have been moved to optional OSFRETIREDxxx subsets. Refer to 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 via a modem. See Section 1.11.

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

  Traditionally, experienced UNIX administrators have used a combination of individual shell commands, scripts. and utilities, or simply edited the system files. Most sections of this book 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. It is strongly recommended that you use the appropriate utilities to update system files so that the structure of these files is preserved.

  Important considerations are:

  – CDSLs – context dependent symbolic links.

    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 the hier(5) reference page.

  – Binary databases, configuration definitions.

    Many system components write data both to text and binary files, and their administrative utilities often re-create the binaries. Other system information is often 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 recently been modified 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. Refer to the *Installation Guide — Advanced Topics* for more information.

1.5 Administrative Utilities Under CDE

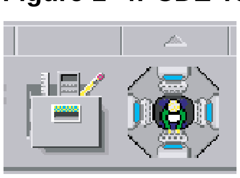
The Common Desktop Environment (CDE) is the default X11 windowing user environment, although the utilities described in this section will run on other X11–compliant user environments. When you complete the full installation, the System Setup graphical interface will be displayed to guide you through 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 will 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 can only 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. Note that 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
```

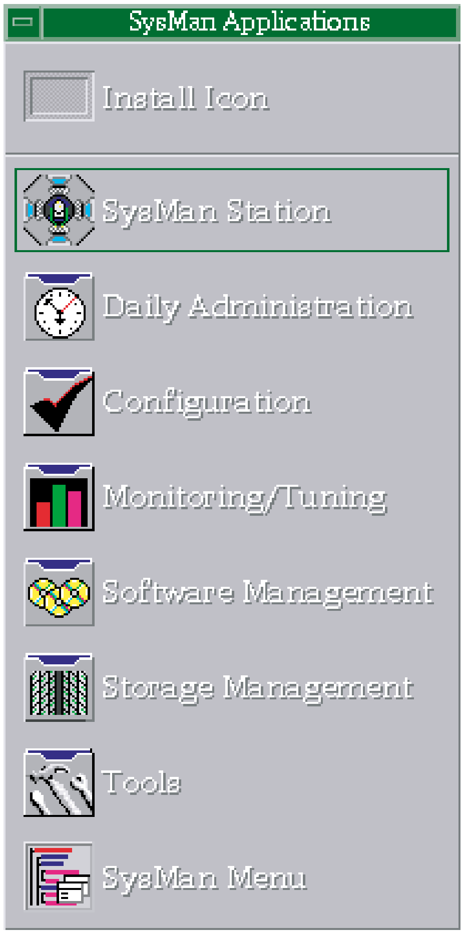
```
# /usr/sbin/X11/dxarchiver
```

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 by clicking on the arrow for the SysMan Applications panel. You can see this arrow above the icon for the SysMan Station, shown in figure Figure 1–4. When you click on 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. Note that in a root login to CDE, this icon appears on the front panel as shown in Figure 1–4.
- Click on a folder icon, such as Configuration to open the sapplications folders, which contain utilities described in Appendix A.
- Launch the SysMan Menu.

2. From the CDE Front Panel by clicking on its tool-drawer icon, shown in Figure 1–4. When the top-level folder is displayed, 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.

Online help is available for the SysMan Menu utilities without actually running any utility. Click on the `Help Manager` icon on the CDE front panel to display the online help browser. The browser includes help families for CDE, the CDE Desktop, and System Management. You can also 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* guide for more information.

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

Additional information is available as follows:

- `sysman`(8) – describes the SysMan Menu and explains how to invoke it for various environments. See also Section 1.6.
- `sysman_station`(8) – describes the SysMan Station and explains how to invoke it. See also Section 1.8.
- `sysman_cli`(8) – describes the command-line option for SysMan Menu, and defines the command options. See also 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 displayed automatically, prompting you to complete system configuration tasks. The initial window contains two options:

- Quick Setup – This option provides a step-by-step guide (or wizard) that navigates you through a typical system configuration. Use the 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.

- 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. Note that 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 `dxlicenses` GUI, 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. Refer to the `dxlicenses`(8) and `lmf`(8) reference pages, and the *Software License Management* guide.
- Disk Configuration – Invokes the `diskconfig` GUI, which enables you to configure and administer disk devices on the system. Refer to the `diskconfig`(8) and `disklabel`(8) reference pages, and Chapter 5.
- Network Configuration Step By Step – Invokes the SysMan Menu Network Setup Wizard option, which is a guide that leads you through the process of configuring and administering networking components on the system. Refer to the `sysman`(8) and `network_manual_setup`(7) reference pages, and the *Network Administration* guide. 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

  Note that in addition to the options offered in the Network Setup Wizard, you may also need to set up other options, such as NTP, depending on your site-specific networking requirements. Refer to the *Network Administration* guide for more information.

- DNS (BIND) Configuration – Invokes the SysMan Menu utility `Configure system as a DNS client`, which enables you to configure the domain name server (DNS). Refer to the `bindconfig`(8) and `network_manual_setup`(7) reference pages, and the *Network Administration* guide.
- NIS Configuration – Invokes the `nissetup` script, which enables you to configure NIS, the network information service. This is also known as `ypsetup`. Refer to the `nissetup`(8) and `network_manual_setup`(7) reference pages, and the *Network Administration* guide.
- 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. Refer to the `sysman`(8), `nfssetup`(8), and `nfs_intro`(4) reference pages, and the *Network Administration* guide. See also Chapter 6 for more information.
- File Sharing – Invokes the `dxfileshare` option, which enables you to access and share file systems. Refer to the `dxfileshare`(8) reference page and the *Network Administration* guide. See also 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. Refer to the `sysman`(8), `ntp`(1), and `ntp_intro`(7) reference pages, and the *Network Administration* guide.
- 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). Refer to the `sysman`(8), `ppp_manual_setup`(7), and `pppd`(8) reference pages, and the *Network Administration* guide.
- SLIP Configuration – See the entry for PPP and the `slconfig`(8) reference page.
- 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. Refer to the `dxaccounts`(8) and `adduser`(8) reference pages, and Chapter 7.
- Mail Configuration – Invokes the Mail Configuration option, which enables you to configure the system to send and receive electronic mail. Refer to the `sysman`(8), `mail_intro`(7), and `mailconfig`(8) reference pages, and the *Network Administration* guide.

- LAT Configuration – Invokes the `latsetup` script, which enables you to configure the Local Area Transport service. Refer to the `latsetup`(8) and `lat_intro`(7) reference pages, and the *Network Administration* guide.
- UUCP Configuration – Invokes the `uucpsetup` Connections Configuration script, which enables you to configure UNIX-to-UNIX connections and modems. Refer to the `uucpsetup`(8) and `uucp_intro`(7) reference pages, and the *Network Administration* guide.
- Printer Configuration – Invokes the SysMan Menu Configure line printers option, which enables you to configure local and remote printers. Refer to the `sysman`(8), `printconfig`(8), and `lprsetup`(8) reference pages, and Chapter 8.
- Security Configuration – Invokes the SysMan Menu `Security` utilities, which enable you to configure base or enhanced security. Refer to the `secconfig`(8) reference page and the *Security* guide.
- Audit Configuration – Invokes the SysMan Menu `Security` utilities, which enable you to configure the audit subsystem. Refer to the `auditconfig`(8) reference page, and the *Security* guide.
- 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 are normally only run by the root user. Refer to the `dop`(8) and `sysman`(8) reference pages and the *Security* guide.
- Prestoserve I/O Acceleration Configuration – Invokes the `prestosetup` script, which enables you to configure Prestoserve. Refer to the `presto`(8) and `presto_setup`(8) reference pages, and the *Guide to Prestoserve*.
- GUI Selection – Invokes a script that enables you to configure the display manager to CDE or xdm.
- ATM – Invokes a script that enables you to configure Asynchronous Transfer Mode (ATM).
- Insight Manager – Invokes a utility that enables you to enable and configure the 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 will have 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 a 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.

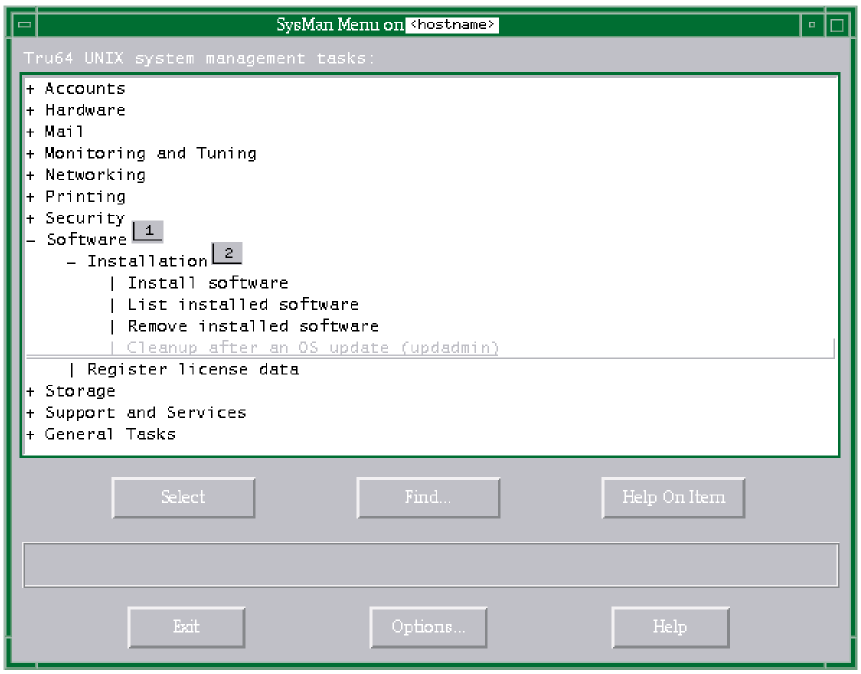
```
# /usr/sbin/sysman -list
```

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 tasks. You can collapse or expand each option if suboptions are available, as indicated by a character preceding each menu item. The plus sign (+) indicate s that further menu items are available while 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 can only select a task to invoke an administrative utility. Figure 1–6 Shows the SysMan

Menu invoked in the CDE user environment. Note that 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 fully expanded, showing `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 will expand 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. Refer to the online help for detailed instructions on navigating through the utilities. The following option buttons appear on the SysMan Menu:

- Select – chooses the highlighted item. Selecting a branch will expand or contract it. Selecting a task will invoke 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 is also displayed in the pane located between the two rows of buttons. This 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 will invoke 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.

Additional information is available in the `sysman`(8) reference page and in the online help. Refer to 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 `sysman -cli` options, although unprivileged users can use it to view system setup data. Refer to the *Security* guide for information on using the division of privileges (DoP) utilities to enable nonroot users to become privileged users of SysMan tasks.

This section provides only a brief introduction to the many features of the `sysman -cli` utility. Refer to the `sysman_cli`(8) reference page for a complete list of options and flags. A set of shell script examples are 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 gives you the ability to 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
```

Note that 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 will be 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 can also 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 can also 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 can also 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 can only run in a windowing user environment such as CDE or Microsoft Windows.

Note

You can only connect between compatible server and client versions of the SysMan Station. If you attempt a connection to an incompatible server, you will see 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 as described in Section 1.9.2.

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

To start the SysMan Station fom CDE:

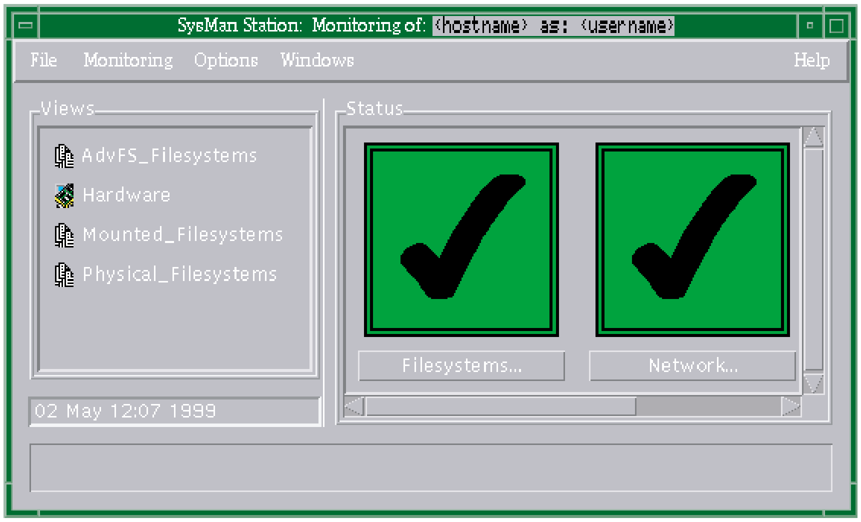
1. Log in as root and use the SysMan Station icon from the CDE Front Panel or from the SysMan Applications sub-panel. (This assumes the default CDE configuration, where the SysMan Station icon appears on the Front Panel under the SysMan Applications sub-panel.)
2. Choose the SysMan Station icon from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

To start SysMan Station from the command prompt, enter:

```
#sysman -station
```

After invoking SysMan Station, you will be 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..., and Storage... options that can be monitored. These options are referred to 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 pressing MB1. A window displaying a list of events will then be 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 also 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 quickly launch your custom views and check on device status.

The main window of SysMan Station provides the following features:

- The Status pane, in which you monitor attention groups. Status options are described in Section 1.8.1
- The Views pane, from which 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 options for changing views or selecting 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 checkmark icon if the status of the attention group is normal. If the status degrades, the icon will change 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:

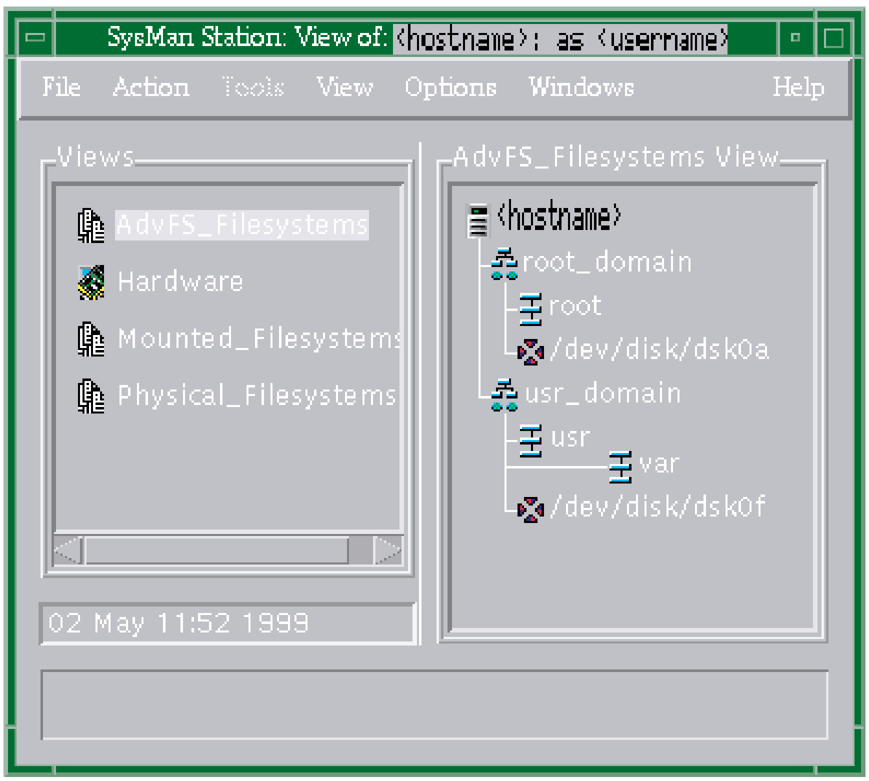
- File systems – Any UFS file systems or AdvFS domains.
- Network – The network and devices connected to the local host, such as `tu0`.
- Storage – Storage devices connected to buses and device interfaces, such as `floppy`, the floppy drive unit that is connected to an `fdi` interface such as `fdi0`.

1.8.2 Using SysMan Station Views

In the Views pane, a list of attention groups is displayed. You can select any item in the list to display a window showing the hierarchical structure of the group. The options are:

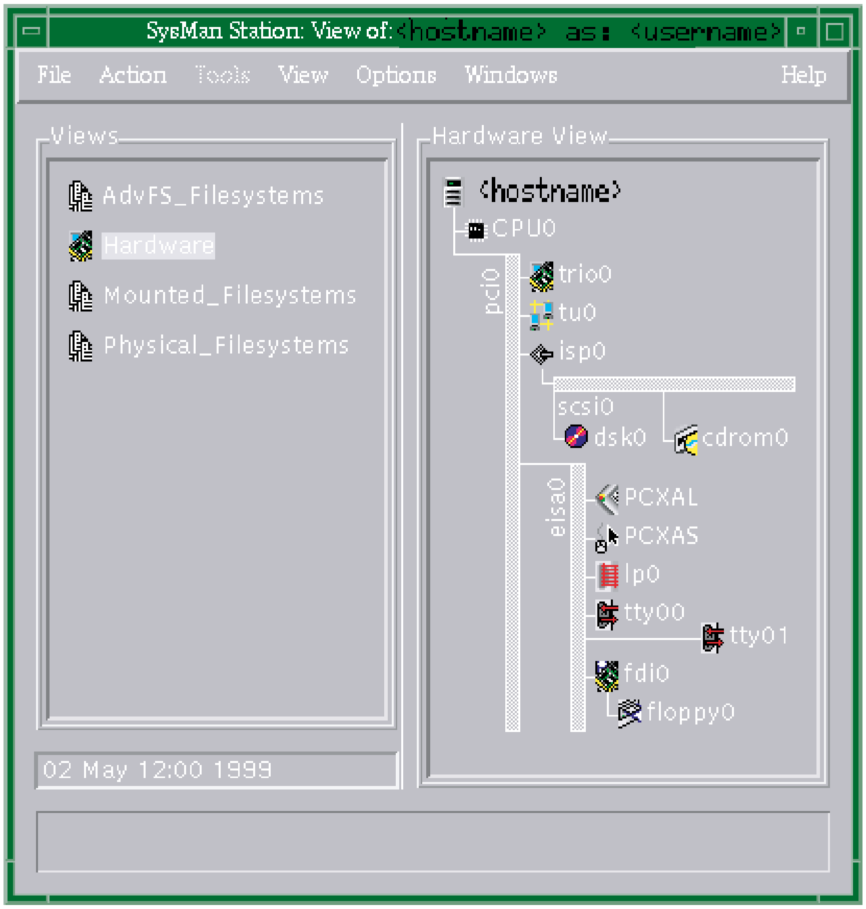
- AdvFS_Filesystems – A view of all AdvFS domains.Figure 1–8 shows a typical AdvFS domains view on a small single-disk system.

Figure 1–8: AdvFS_Filesystems View



- Hardware – 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.

Figure 1–9: Hardware View



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

- Mounted_Filesystems – A view of file systems that are currently mounted, in a similar format to the AdvFS_Filesystems view.
- Physical_Filesystems – 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. Note that 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 – These options display or remove the subcomponents under a component. For example, click on the Expand option when selecting a SCSI bus, and all the attached devices will be displayed. Click on 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, click on the Hide option when selecting a PCI bus such as `pci0`. All the devices attached will be hidden. This means that you cannot display the devices by double-clicking MB1 on the bus or by selecting the Expand menu option.

    Click on Unhide Children to enable display of the PCI bus devices.
- Available SysMan Menu utilities – This option 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 grayed out when not available.

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

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

- File – Options to print the current screen, create a new connection, close the current window and to exit from SysMan Station.
- Action – Options to change the grouping of components and the default appearance of displays, such as the Expand and Hide options.
- 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 will be 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 will be prompted to save you custom view before you exit SysMan Station. You can then assign a name and an icon to the custom view. When you next invoke SysMan Station, your custom view will be added to the Views pane.

For information on installing the SysMan Station under Microsoft Windows, refer to Section 1.10.

1.9 Configuring and Using Insight Manager

Insight Manager is a World Wide Web-based management utility that enables you to look across a heterogeneous computing services environment and access information about any entity connected to the network. Entities are referred to as devices and 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 administrative tasks on components. Using Insight Manager can facilitate other administrative tasks such as asset management, asset security, workload management, and event management. Note that Insight Manager is also used by applications to display online help when the applications are running in java mode.

1.9.1 Web Browser Requirements

Your system must have the minimum requirements noted in this section. Some software components can be downloaded and installed as documented in lated sections:

- The browser must support the following:

- Tables and frames in the display
- JavaScript and the Java Development Kit (JDK) Version 1.1
- TCP/IP as the network transport

• Browsers that have been verified to work with this release are as follows:

Operating System of Client	Browser and Version
Tru64 UNIX version 4.0F or higher.	Netscape Communicator Version 4.5 or higher
Tru64 UNIX version 3.2C through 4.0E.	Netscape Communicator Version 4.06 or higher
Windows NT Version 3.51 or Version 4.0, Windows 95, and Windows 98	Microsoft Internet Explorer version 4.72.2106.8 or higher. Netscape Navigator 4.04 or higher.

• The following browser features must be enabled:
  - Enable Java and JavaScript
  - Accept all cookies or accept cookies originating from the same server as the page being viewed.

1.9.2 Setting Up Insight Manager from the SysMan Menu

The Insight Manager agent (daemon) is configured by default when you install the operating system, you do not need to start it. The agent is initialized during system startup transition to run level 3, and the initialization script is located in `/sbin/rc3.d/S50insightd`. This script runs the `/usr/share/sysman/bin/insightd` command and prints a boot-time message at the console when the agent is successfully started. (See Chapter 3 for an explanation of run levels.)

The SNMPD agents, `/usr/sbin/os_mibs` and `/usr/sbin/cpq_mibs`, are also invoked during transition to run level 3 and are invoked by the `/sbin/rc.3d/S49snmpd` script. To test that the system is properly configured, enter the following commands

```
#  ps agx | grep insight
   487 ??       S        0:35.63 /usr/share/sysman/bin/insightd
# ps agx | grep cpq
   466 ??       S        0:00.36 /usr/sbin/cpq_mibs
# ps agx | grep os_mibs
   468 ??       S        0:00.36 /usr/sbin/os_mibs
```

If you do not wish to have the Insight Manager Web Agent enabled by default, you can configure its default setting through the SysMan Menu as follows.

Invoke the SysMan Menu as described in Section 1.6. and expand the `Monitoring and Tuning` branch to display the `Set Up Insight Manager` task and launch the set up utility. To set up Insight Manager, complete the setup window options as follows:

1. Enable Insight Manager – Check this box to enable access to the Insight Manager agents. This option enables or denies access by all users, including root.
2. Administrator Password and Password verify – Supply a password for the administrator, who will be the root user. When Insight Manager is set up, it allows password access to be granted to other users. You can use this option to update the administrator password at any time.
3. Enable anonymous logins to WebAgent Applications – Allows nonprivileged users to invoke Insight Manager and view details of any connected devices in the local area network although users cannot perform any operations unless authorized. There are security implications to consider when giving nonprivileged users access to Insight Manager.

You can also run the setup utility from the command line as follows:

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

1.9.3 Insight Manager Concepts

As a Web-based utility, Insight Manager is independent of operating environments. This method of administering a computing environment is known as Web-based management (WBEM). At the current level of implementation, some Insight Manager features are fully implemented in some operating environments, but are not yet implemented in others. This means that you can use all features of Insight Manager on Windows NT systems, but you cannot use certain features on UNIX.

In its present implementation, 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 both view details of devices and invoke administrative tasks. On a UNIX system, you can use Insight Manager to view details of devices, but you must invoke the UNIX SysMan Menu or SysMan Station to perform administrative tasks.

Note that other operating environments and platforms are supported. Only Windows NT on PC platforms is considered here, to give you an idea of the restrictions as they apply in an environment where PC clients are served by UNIX servers. In future, all administrative tasks will be fully integrated and provide seamless WBEM administration regardless of hardware platform or operating environment. At present, Insight Manager protocols are able to communicate between the following operating environments in addition to UNIX:

- Windows NT
- IBM OS2
- Netware
- OpenVMS

The main server component of Insight Manager is Insight Manager XE, which provides full administrative services for Windows NT. XE can communicate with any other device in the local area network or domain that is running the Insight Manager agent. In the context of Insight Manager, 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 Insight Manager agents is a device that can communicate with the XE server, although some devices may require additional hardware.

To be manageable through WBEM, a device must have some form of operating environment that is recognized by Insight Manager. The operating environment must be capable of communicating device information to the WBEM network, and also of receiving and executing instructions sent from other (authorized) devices in the WBEM network. The operating environments must be able to run Insight Manager agents which communicate with each other using a standard protocol. Devices, and their operating environments, are able to provide information about hardware and software status using a data model, such as the Common Information Model (CIM) or 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. There are industry standard MIBs and proprietary MIBs. 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 or Windows NT servers, you can use Insight Manager as your common interface to administrative tasks. For example, as an authorized (root) user working at your PC, you can invoke Insight Manager to view the general system status of an AlphaServer running UNIX, then invoke SysMan Station to check the specific status of a peripheral such as the status of file systems on a disk. You can also launch a SysMan Menu task to perform operations on that file system.

From the same browser page, you can then respond to a status warning from a Windows NT server and (if the XE agent is running at your site) perform troubleshooting operations. For example, in a heterogeneous computing environment, devices can run many different operating systems with vastly different disk and file management systems, such as AdvFS on UNIX and the FAT or NTFS file system on Windows NT. What an administrator may need to know is how much space is available on disks, so file systems can be serviced before they run out of space. Insight Manager enables you to monitor all such data at a single point, and can (in some cases) be used to initiate corrective actions.

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

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

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

Two devices have special status on a WBEM network:

- The Windows NT server that is running the XE agent. This system must be configured with the Insight Manager XE software, and will enable many of the remote administrative services on other Windows NT systems. The XE server has a port address of `:280`, and you connect to this server from your browser by specifying the URL as follows:

  ```
  HTTP://chubb.cu.da.com:280
  ```

  Note that an Insight Manager XE server is not necessary for peer-to-peer administration of UNIX systems. You can view system status and perform administrative tasks on UNIX systems using SysMan Menu and the SysMan Station in such an environment.

- The master device, which polls all other available devices for their status and data, stores it in a master record and makes the master record available to any other device. This device is always the one with the lowest local TCP/IP network address (in the format 110.222.333.110). An important consideration here is that you must assign this number to a system that is capable of handling the WBEM traffic. If the device with the lowest TCP/IP address becomes unavailable for any reason, the device with the next lowest TCP/IP address becomes the master. All other devices are slaves.

A feature of Insight Manager is auto-discovery. Any Insight Manager agent can discover any other Insight Manager agent in the local network domain.

You can configure a system so that it does not participate as a master or participate in auto-discovery, as described in Section 1.9.4.

A summary of the requirements for UNIX systems in a WBEM environment is as follows:

- The Insight Manager agent must be installed and configured on the UNIX system. From the UNIX system, you must use the Netscape browser.

  Note

  Launching the SysMan Menu or the SysMan Station from the Insight Manager Web page is currently not supported. You must launch the SysMan Menu and the SysMan Station as described in Section 1.6 and Section 1.8

- A local network of systems is optional. You can run Insight Manager on a single UNIX system, and connect via the Internet from your home PC to perform administrative tasks. The Insight Manager agent does not have to be installed on your connecting PC.
- A Windows NT server running the XE agent is optional and only required if you plan to remote-manage PC clients that are running Windows NT, or other systems that support an Insight Manager agent.
- Client systems, such as PCs running Windows NT, must have the Insight Manager agent installed if you want to remotely manage such systems. Connect to `http://www.compaq.com/products/servers/management/` to download agents for supported systems. This URL can also be reached from the default home page on your UNIX system.
- PCs must be running the Internet Explorer Web browser. Netscape will not work.
- A designated master system that has the lowest TCP/IP address in the local network domain.

1.9.4 Using Insight Manager

When Insight Manager is configured, you can also connect to the Insight Manager Web agents of any system in the local network domain that is running the 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`, unless you are connecting to the XE Agent, in which case it is `280`.

If the connection succeeds, you will see the Insight Manager Web-based Management device home page. This page provides links to other pages both in text format, and as icons. Move the pointer to an item and press MB1 to invoke a link. Note that the UNIX device homepage, and the links from it will differ in content from the home page on other systems, such as a PC running Windows NT. You can always navigate back to one of the following three pages from any Insight Manager page:

- The Device Home page – The topmost page in the hierarchy, and the starting point for all other links.
- The Configuration Options page – Enables you to configure the Insight Manager on the local host.
- The Devices page – Displays all devices found in the local WBEM network.

The following links are available from the Device Home page:

- The Insight Manager page banner (icon link) – Provides links to related World Wide Web pages.
- Login Account (text link) – Enables you to invoke the Account Login administrative page, which can only be used by an authorized user. During Insight Manager setup, you assigned an administrator password as described in Section 1.9.2. You use that password here to:
  - Log in as administrator (if you are not already). When you do this, you are returned to the Device Home page, and your login status will change from anonymous to administrator. Note that you must use administrator and not root on the initial login.
  - If you are already logged in as administrator, you can elect the “changed” link to initialize the password of other Insight Manager accounts: user, operator, or administrator.
- Refresh (text link) – Reloads the current page.
- Options (text link) – Displays the Configuration Options page, which contains items that can be set by the administrator to control access to Insight Manager. The options are:
  - Anonymous Access – Enabling this option allows nonprivileged users to view devices and device data. This option is turned on in the default configuration.
  - Local Access – Enabling this option allows any user with access to the local console to perform administrative tasks. This option should

be selected with care due to the security implications. This option is turned off in the default configuration.

  - Auto Delete Users – When enabled, this option will automatically delete user directories that have not been accessed for a specified number of days. Enter a time in days after which local user directories will be deleted. This option is turned on and set to 30 days in the default configuration.
  - HTTP Auto-discovery – When enabled, the local server will participate in auto-discovery of any local hosts running the Insight Manager agent. Enter a time in minutes to control the frequency of auto-discovery updates. Use this option to prevent auto-discovery of inappropriate systems. This option is turned on and set to one minute in the default configuration.
  - HTTP Auto-discovery master – When enabled, the local server will participate in auto-discovery as a potential master node. Use this option to prevent inappropriate systems from becoming the master device in a WBEM network. This option is turned on in the default configuration.
  - Save Configuration (button) – Select this button to save and implement a new configuration.
  - Default Configuration (button) – Select this button to invoke the default configuration.

- Devices (text link) – Displays a page with the following two lists:
  - Insight Manager XE Servers – If the local WBEM network contains an XE Agent server, information and links will be listed here and you can connect to the server device if authorized.
  - HTTP Auto-discovery Device List – Any devices that can be discovered automatically are listed here by device (host) name and TCP/IP network address. Each name and address is a link to the named system. You can connect directly to another device simply by moving the pointer to the name and pressing MB1. If you opt to connect to another device, the Device Home page for that system will be displayed.
- Insight Management Agents – Displays the page that summarizes system data for the local host. The page is described in Section 1.9.5.
- Configuration Report (icon link) – On UNIX systems, the configuration report is created by the `runsyscheck` utility, which provides a system census report for the local host. If a report has not been created, you can opt to launch the utility and create a first report. Note that the report can take up to an hour to generate, and may impact system performance.

If an existing report is found, the Configuration Report page is displayed and you can browse its content. The report is useful for detecting system configuration and tuning problems, log file and event file settings, and general configuration information. This report is also of use when troubleshooting system problems and can be used in conjunction with the Insight Manager Agents page to resolve such system problems.

Refer to Chapter 3 for information on `sys_check`, including information on how to set up a `cron` task to create regularly updated system census reports. The `sys_check`(8) reference page describes how to invoke the `sys_check` utility from the command line, and documents all the command options.

- SysMan (icon link) – When accessing Insight Manager from a PC using Internet Explorer and connected to the Device Home page of a UNIX system, you can opt to launch the SysMan Menu or the SysMan Station. You are prompted for the name and password of an authorized account. Because the SysMan suite is run as a Java applet, you cannot close or otherwise use the associated Internet Explorer window. If you attempt to do this, the Java applet will also be closed.

  With this option, you can run any SysMan task from any networked PC without the need to install the SysMan Menu and SysMan Station client PC software as described in Section 1.9.5, although the performance of the applications will be slower than when the client software is installed.

For more detailed information on any Insight Manager Web page, consult the online help by selecting the question mark (?) icon.

1.9.5 Using the Insight Management Agents Page

The Insight Management Agents page provides detailed information on the status of a device, its operating system, and any components that can be seen by the Insight Manager agent. If the device is a computer, a component can be a mass storage peripheral, such as a hard disk. This Web page is divided into three frames as follows:

- The Insight Management Agents frame – This frame contains the icons and labels that describe the status of a device and its components (if any). It also contains several options that can be selected to display more information in the Summary (main) frame. The options provided in this frame are as follows:
  - Agent Help (text option) – Invokes the Web-based help page.
  - Summary – Returns to the Summary (main) frame. Use this option to reset to the default display.
  - Device Home – A link to the Device Home page

– Options – Enables you to configure the presentation of device data, and set values such as the refresh interval. Refer to the online help for more information on the configuration options.

Also shown in this frame are the icons and legends (labels) that describe the status of a device and its components. The status can be as follows:

– Unknown (blue button) – Insight Manager was not able to determine the status.

– OK (check mark) – The device is functioning normally.

– Degraded (yellow triangle) – The device is not functioning normally, and you should examine it more closely. For example, you could use the Configuration Report option on the Device Home page to obtain system census data for a UNIX system. Alternatively, use the SysMan applications to invoke the Event Viewer and check for system errors.

– Failed (x mark) – The device or component is not functioning.

- The Navigation frame – This frame lists all the components for which data can be obtained. It also lists any associated data items. It provides status data on hardware, such as network interface (NIC) cards, and also data on general system status, such as CPU utilization. The content of this frame is dependent on what device data can be provided by the device to Insight Manager. The typical contents for a device that is a computer system are as follows:

  – Configuration – Provides information on the system type, the main CPU board and the operating system software. You can select any of these links to display further data in the Summary frame.

  – Mass Storage – Provides information on the disk devices, and current allocations of disk space in use by file systems. Also included is information on device controllers, such as SCSI controllers. If you select a SCSI controller, you can display the buses and any disk devices connected to a bus. In turn, you can select an individual disk and display properties of that disk.

  – NIC (Network Card) – Provides information on the network interfaces, and any other communications interfaces that are configured, such as a serial line interface. You can obtain information about the device, such as its device type, physical address, and settings.

  – Recovery – Provides information on environmental conditions, such as the status of the environmental monitoring subsystem (if configured on the host). You can also display the log of start up messages that was saved when the system was last booted. (Note that on UNIX systems, actions such as reboot are not currently available.)

Data items in the Status frame may appear as links that can be selected to display more information in the Summary frame.

- The Data (main) frame – Displays links selected in the preceding frames. When you first access the Insight Management Agents page, this frame contains information that enables you to identify the device type and model, its physical and network locations, and its general status. Use the Summary link in the Insight Management Agents frame to display this default information.

  The question mark (?) option provides a link to the Insight Manager Web-based documentation, which provides more information on all the Insight Manager pages.

1.10 Using SysMan on a Personal Computer

From Insight Manager, you can install the SysMan Station client and create a launch point for the SysMan Menu in the user environment running on a personal computer. This feature supports Microsoft Windows, MacOS, and Linux. Full information on this feature, together with a download address for the required software, is provided in a Web page available from the UNIX system. This page is located at http://<host>:2301/sysman/index.html, where <host> is the host name and address or the TCP/IP address. The process is as follows:

1. On the UNIX system, use the Netscape Web browser to launch the .../sysman/index.html page. A link to this page is provided on the default UNIX home page, under the heading System Management. Scroll down to the section titled “Managing UNIX from a PC”.

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

2. If it is not already installed on your client system, select the Web page options to download the software kits for the Java run-time environment and the SysMan Station client.

3. 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.

4. When the installation process is complete, click on the Start button and select Programs. The SysMan Station and SysMan Menu are listed as Java applications on the Programs menu. Select either icon to launch the required application.

5. When you launch either application, a dialog box is displayed, 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 might 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 press OK, the application window opens (the time to start up depends on the current network speed and traffic). You can then use the SysMan Station or SysMan Menu as described in 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 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. Consult 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

Note that running the Environment Configuration Utility (ECU) on the remote system will cause 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 already installed and configured for use.

1.11.1.1 Connecting the Modem to COMM1

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

Refer the hardware documents supplied with your modem for connecting the modem to your system. Consult the `modem`(7) reference page 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, so refer to your modem documentation to verify the interval.

The timer can be modified via 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
```

Where n = 0, 1, or 2

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

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

Where n = 0, 1, or 2

Note that 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.

Consult 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: Consult your system hardware documentation.

Note that if you change the baud rate, flow control, or modem setting (for example, using the getty command), the change will be propagated down to the console level and the environment variables will 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.

Once 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 will be displayed.

See the tip(1) reference page for more information.

1.11.2.1 Using the Console Port

Once 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. Note that the following features may be useful for remote administration:

- The UNIX-to-UNIX system copy utility, uucp for copying scripts and files to the remote system. See the uucp(1) reference page for more information.
- A kernel debugging tool, ikdebug can be invoked and used remotely. See the ikdebug(8) reference page 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 additional 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 the syslog(1) reference page.

1.11.2.1.2 Shutting Down the Remote System

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

```
# stty -hupcl
```

See the stty(1) reference page for more information.

When the shutdown is complete, you will 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, check the set up as follows:

- The local modem does not dial out.

  Check 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.

  Review the `modem`(7) reference page and verify the settings for dial-in access.

- The remote modem answers and then disconnects.

  This is most likely to be due to incorrect settings for dial-in access. Review the `modem`(7) reference page and verify the settings for dial-in access.

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

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

- The connection is dropped when the remote system is shut down via 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

Shutting down the system and then restarting it are routine tasks that you may need to perform periodically. On some installations, 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.

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.

You should 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 the 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.

This chapter contains the following information:

- Section 2.1 provides an overview of starting up and shutting down the system.
- Section 2.2 explains the boot operation.
- Section 2.3 describes how to prepare for booting your system.
- Section 2.4 explains how to boot your system.
- Section 2.5 describes the different system run levels.
- Section 2.6 explains how to change the system run level.
- Section 2.7 describes boot considerations for multiprocessor systems.
- Section 2.8 explains how to set the system date and time.
- Section 2.9 explains how to troubleshoot boot problems.

- Section 2.10 describes options for shutting down the system.
- Section 2.11 describes how to shut down the system from multiuser mode.
- Section 2.12 describes how to shut down the system from single user (root) mode.

2.1 Overview of Shutdown and Booting

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

A system can be shut down automatically or manually. The following shutdown methods and utilities are available:

- Configure system monitoring tools such as environmental monitoring to shut down the system automatically if certain system events occur. Refer to Chapter 13 for information on event management.
- Use the following utilities to manually shut down a system:
  - The SysMan Menu and SysMan Station enables 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. You can also invoke the task from the command line by typing the following command:

    ```
    # sysman shutdown
    ```

    Refer to Chapter 1 for more information.
  - The `/usr/sbin/shutdown` command line interface can be run from a character-cell terminal. Specify command options as documented in the `shutdown`(8) reference page.

Note that the Shutdown icon in the CDE Application Manager – DailyAdmin folder invokes the SysMan Menu task named `Shutdown the system`. In this release, `dxshutdown` has been moved to an obsolete interfaces subset and is replaced by the SysMan Menu task.

2.1.2 Boot Methods

Booting the operating system is performed from the system's console firmware. 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, this symbol is referred to as the console prompt. The console is sometimes called the System Reference Manual (SRM) console or the firmware console. Refer to 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:

- You can manually boot the local system from the console. You can also boot a remote system via a connection, such as the remote console method documented in Chapter 1.
- You can automatically boot the system after a shut down. For example, if you use SysMan Menu or the SysMan Station to initiate a shut down, you can specify that the system should automatically boot after the shutdown is completed.
- You can cause the system to boot automatically by setting the `auto_action` console variable. The system will then boot automatically 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

The following documentation contains information that is relevant to system shutdowns and reboots:

- Books
  - Refer to the Owner's Manual that came with your system for information on the console comands and variables. See also the `consvar`(8) reference page, which describes `consvar`, an interface that enables you to manipulate console environment variables from within the operating system, depending on the firmware revision.
  - Refer to the *AdvFS Administration* guide and *Logical Storage Manager* guide for information on file systems, should you need to check and repair damaged file systems before rebooting.
  - Refer to 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* guide provides information on analyzing crash dump files.
- Reference pages

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

- shutdown(8) – Describes how to invoke and use the shutdown command-line interface.
- sysman(8) and sms(8) – Provide information on using the SysMan options and describe how you invoke these utilities so that you can then run the Shutdown the system task.
- wall(1), rwall(1), fastboot(8), fasthalt(8), halt(8), reboot(8), fsck(8), init(8), rc0(8), rc2(8), and rc3(8) – Describe related commands and utilities.

- Online help

  The following help is available:

  - The shutdown -h command provides help on the command line options.
  - SysMan Menu and SysMan Station tasks provide an online help volume for each task. See also the introductory online help available at: /usr/doc/netscape/sysman/index.html

  See Chapter 1 for information on invoking on-line help.

This *System Administration* guide also contains the following topics of relevance to planning and managing shut downs and error recovery:

- Some systems support environmental monitoring, which can be used to shut down a system automatically in the event of a problem such as loss of a cooling fan. Refer to Chapter 12 for information on configuring this feature.
- Refer also to Chapter 12 for information on error conditions, log files, and crash dumps.
- The Event Manager, (EVM) and the SysMan Station provide integrated monitoring and event reporting facilities that enable you to monitor local and remote systems and clusters. Refer to Chapter 1 for information on invoking these features.
- Refer also to Section 1.11 in Chapter 1 for information on remote serial consoles if you administering systems at remote locations, or if there is a network failure and modem communications are required.
- Refer to Chapter 5 for information on diagnosing disk and bus problems.
- Refer to Chapter 9 for information on implementing a backup schedule, from which you can recover lost data if necessary.

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 – Is the custom UNIX kernel that is normally booted. The generic kernel/genvmunix can be booted if the custom kernel, /vmunix, is damaged.
- /sbin/rc0, /sbin/rc2, and /sbin/rc3 – Contain run level commands.

  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

The following utilities are also used during the boot operation

- fsck – The fsck command is a wrapper program for the ufs_fsck program, which checks and repairs UFS file systems. See advfs(4) and the *AdvFS Administration* guide for information on checking AdvFS file systems
- consvar – The consvar command to gets, sets, lists, and saves console environment variables while the operating system is still running. This feature is only available on some recent versions of the firmware. You may have to upgrade your firmware, as described in the *Installation Guide*.

  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 will be 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 should understand what is happening during the system boot operations, and be prepared to respond if problems occur. Although certain boot operations are dependent on the hardware, some features apply to all systems, as described in the following sections.

2.2.1 Booting Automatically or Manually

The system always boots either automatically or manually. In an automatic boot, the system controls the entire operation. When you boot the system to multiuser mode, or shut down the system with the reboot flag, or when the system panics and recovers, you are relying on an automatic boot. With an automatic boot, the system begins the initialization process and continues until completion or failure.

Manual intervention may be required if the automatic boot fails for some reason, for example, if the `fsck` command fails.

In a manual boot, the system controls the initial operation, turns control of the procedure over to you, 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 root 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 checking 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 might check new hardware, mount and check aberrant file systems, change disk partitions, or set the system clock. When you finish your work, you return control to the system, and `init` 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, module loading, and so on.

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 checking 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 UNIX operating system allows you to boot an alternate kernel. For example, if you cannot boot your system, you could boot `/genvmunix` to troubleshoot the problem with your system. You could also boot an alternate 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 discussed include the following:

- A powered-down system
- A powered-up, halted system
- A powered-up system in single-user mode
- A crashed system
- A networked system that has been taken out of the network

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 prompt, enter the following command:

```
# mount -u /
```

2.3.1 Preparing to Boot a Powered-Down System

A system is powered down when the hardware (processor, devices, and peripherals) is turned off. Administrators power down the hardware periodically for routine maintenance or to configure new devices.

If you are preparing to reboot your system from a powered-down state, follow these steps:

1. Confirm that the hardware and all peripheral devices are connected. Refer to the operator's guide for your hardware for information and instructions for interpreting diagnostic output.
2. Power up the hardware and peripheral devices. Remember to power up all devices that you powered down earlier. Refer to the operator's manual or the hardware user's guide for instructions on starting your hardware and peripherals.
3. Confirm that the hardware completed its restart and diagnostic operations. Most hardware provides a diagnostic check as a routine part of its startup operation. Refer to the operator's manual for your hardware for information about your hardware's restart and diagnostic operations.
4. Wait for the console prompt (>>>). If you have enabled your system to boot automatically when it is powered up, press the halt button to display the console prompt. Refer to the hardware operator's guide 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.
5. Decide which startup mode you want to initiate:
   - If you have tasks you need to accomplish and want the system to restrict access to all users but root, plan to boot to single-user mode.
   - If you do not require single-user access and you want the system to initialize all functions, plan to boot to one of the multiuser modes: multiuser without networking or multiuser with networking.
6. Enter the boot command that corresponds to the desired startup mode.

   Refer to 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:
   - If you have tasks you need to accomplish and you want the system to restrict access to all users but root, plan to boot to single-user mode.
   - If you do not require single-user access and you want the system to initialize full functionality, plan to boot to one of the multiuser modes: multiuser without networking or multiuser with networking.
2. Enter the boot command that corresponds to the desired startup mode.

   Refer to Section 2.4 for the commands and procedures required to boot your system.

2.3.3 Preparing to Transition from Single-User Mode

When your machine is powered up and enabled, the processor is running, and access is limited to root, the system is in single-user mode.

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

1. Decide if you should continue in single-user mode or if you should go to multiuser mode:
   - If you have additional tasks that you need to perform and you want the system to deny access to all users but root, plan to continue in single-user mode.
   - If you do not require single-user access, or if you have completed your tasks and you want the system to initialize full functionality, plan to go to one of the multiuser modes: multiuser without networking or multiuser with networking.
2. When you are ready to go to multiuser mode, press Ctrl/d. Refer to 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. Refer to Chapter 12 for information on saving crash dump files, and to check 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. Refer to the operator's guide for your hardware for information and instructions for interpreting diagnostic output:
   - In the unlikely event that 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. Refer to the hardware operator's guide for the location of the halt button on your system.
5. Decide which startup mode you want to initiate:
   - If you need to deny access to all users but root, plan to work in single-user mode. After a crash, it is wise to work initially in single-user mode. You should check all file systems thoroughly for inconsistencies and perform other post-crash operations before enabling system access to other users.
   - If you need to allow access to you and to all other users with login permission, plan to boot to one of the multiuser modes: multiuser without networking or multiuser with networking.
6. Enter the required boot command.

   Refer to 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 will try to start all the network services that are configured. This will result in the boot process hanging, or taking a very long time to test for the presence of services. If you take a system out of a network without unconfiguring the services, or if a system crashes and has to be disconnected from the network, you will need to perform the additional steps before rebooting the system.

There may also be times 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
   ```

   You may want to set other variables if you plan to do things such as boot from an alternate disk. See Section 2.4.1 for more information.

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

   ```
   >>> boot
   ```

3. When the single-user root (#) prompt is displayed, you will need to use an editor. Mount the root file system as follows:

   ```
   # mount -u /
   ```

   This enables you to use the ed line editor to edit system files if required, and also to access other 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 are important for startup operations and for system configuration. Avoid modifying these files with anything other than the rcmgr command; other subsystems or utilities may not be able to read the files correctly if modifications are not made using rcmgr. Refer to the rcmgr(8) reference page for more information. Refer to 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), you would 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 have completed the modifications, halt the system and reset any console environment variables. For example:

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

7. Your system will now reboot to multiuser mode, without attempting to start any network services.

Note that there might be slight variations in the console commands, depending on the system type. You should consult 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

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.

Note

This section provides examples of typical console settings. You should refer to the hardware documentation that came with your system. For example, you may be booting UNIX as an alternate operating system to Windows NT and other console commands may be required. See also the information on booting systems in the *Installation Guide* and *Installation Guide — Advanced Topics*.

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 (on and processors)
`cpu_enable`	Selectively enables particular processors from the console

To prepare the hardware, perform the following steps:

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

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

   The previous command will halt 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

	Action
`a`	Boot to multiuser mode. (By default, the kernel boots to single-user mode.)
`k`	Use the `kdebug` debugger to debug the kernel. Refer to the *Kernel Debugging* guide for more information.
`d`	Use full crash dumps. (By default, partial dumps are used.) Refer to Chapter 12 for information on crash dumps.
`i`	Prompt for the kernel and special arguments. (By default, no questions are asked.)

The options are concatenated into the `boot_osflags` variable to achieve the desired effect. 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
```

If you want to boot from the dual SCSI TURBOchannel option card (PMAZB or PMAZC), complete the following steps:

i. Identify the slot number of the PMAZB or PMAZC option card:

```
>>> show conf
```

This command displays the system configuration.

ii. Determine the unit number of your system's devices.

Use the `conf` command with the slot number to identify the unit numbers of the devices attached to that controller. For example, to look at the devices attached to the controller in slot 1, enter:

```
>>> t tc1 cnfg
```

A display appears identifying the unit number of each device attached to that controller. Identify the unit number of the device from which you want to boot.

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 syntax with the `bootdef_dev` variable to set a default boot device:

set bootdef_dev *device*

For example, to boot the system off of disk `dka0`, enter:

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

To boot the system from the first disk on the PMAZB or PMAZC option card in TURBOchannel slot 1, enter the following command. Note that the double quotes (") are necessary for the console to understand where it is booting from.

```
>>> set bootdef_dev "1/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 console variable `bootdef_dev`. During configuration of a dual-controller system, one of the controllers is designated as the preferred path. The boot devices on this controller must be specified as the first arguments to `bootdef_dev`.

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 console variable `bootdef_dev` 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 any spaces. If the console is unable to boot from the first device, it will try 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 file name. For example:

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

The system will display informational messages.

On some processors, you can also 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 `genvmunix` kernel, 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, turn the system power switch off and then back on again.

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* guide for 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 `bootdef_dev`

  To override the `bootdef_dev` variable, supply the desired boot device 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 `boot_osflags`

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

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

  To override the `boot_osflags` variables, pass the desired choices to 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
  ```

- Overriding `boot_file`

  To boot a kernel other than that specified by `boot_file`, enter the boot command with the following syntax:

  b -fi *kernel*

  For example, to boot the `genvmunix` kernel, enter:

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

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 are to run (and which processes are to be killed if the system is changing from one level to another) at a specific run level. Refer to the `init`(8) and `inittab`(4) reference pages and to 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, check 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. Refer to the `inittab`(4) reference page for more information about defining run levels. Refer to the `getty`(8) reference page for more information about defining terminal lines and access.

Before changing to a new run level, use the `wall` or `write` command to warn users that you intend to change the run level. Because a change in run level could result in termination of a user's `getty` process (which disables their login capability) as well as the termination of other processes that the user is running, you should communicate the change to each logged in user.

Check 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 will not terminate in response to a `kill` signal from `init`.

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 init 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 from Single-User Mode

Use the Bourne shell when working in single-user mode and press Ctrl/d to change run levels. The shell terminates in response to Ctrl/d and displays the following message if transitioning from single-user mode to multiuser mode during a boot operation:

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

If this transition is made from single-user mode with the previous state having been multiuser mode, then a prompt is issued for input of the desired run level. 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. To use the command, log in as root and use the following syntax:

init [0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | S | s | M | m | Q | q]

The init command invokes the following run levels:

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, and 5through 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 output.
M, m	Moves control to the console device and halts to single-user mode.

Run Level	System State
Q, q	Specifies that init 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, init 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, use Ctrl/d or enter exit 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, init reexamines the inittab file and starts new processes, if necessary. For example, if you recently added new terminal lines, init activates the getty process for these terminal lines in response to the init q command.

Refer to the getty(8) reference page for further 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.

From a system administrator's point of view, this additional computing power requires little to no additional system management work. All the administrator should see is additional available computing power. It may be that additional I/O capabilities will be required to more efficiently utilize the extra power.

2.7.1 Adding CPUs to an Existing System

At boot time, the system determines the number of CPUs available. Adding computing power to your multiprocessing systems is as simple as installing the processor board and rebooting the system. You do not have to reconfigure the kernel; you may have to modify any tuning that was done to limit the number of processors available, and you may need to install a Product Authorization Key (PAK). For more information on PAKs, 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 notes which processor failed, then automatically reboots the system. Although the operating system continues, you must manually restart the failed processor. 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 might need to reset the time or date. For example, with battery-powered clocks, you might 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 following syntax with the `date` command:

date *[[cc]yy]mmddHHMM[.ss]*

The sequence of date and time parameters can vary depending on what command options you use. Refer to the `date`(1) reference page for more information. The following table shows the value of the parameters:

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 AM Jan 7, 2000 using the mmddHHMM[[cc]yy][.ss] format, enter one of the following commands:

```
# date  010709342000
# date  0107093400.00
# date  010709342000.00
```

If you are changing the year, the system disk must be updated 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. Note also that the root file system will now be mounted read-write.

2.9 Troubleshooting Boot Problems

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

- Hardware failure

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

- Software failure

  Software can fail for the following reasons:

  – An incorrect boot path was specified

    Refer to Section 2.4 or your system's hardware guide for instructions on specifying the correct boot path.

  – The kernel is corrupt

    If you suspect that the kernel may be corrupt, try booting the generic kernel, `/genvmunix`. This will provide you with a fully functional system and you can begin debugging procedures using the `kdbx` or `dbx` utilities to analyze crash dumps. Refer to the `kdbx`(8) or `dbx`(1) reference pages for more information. Refer to 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 checks and repairs UNIX File Systems (UFS). If `fsck` finds something wrong, it prompts you for an action to take. Use extreme care under these circumstances so that you do not inadvertently overwrite or remove any files. Refer to the `fsck`(8) reference page 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. Refer to the *AdvFS Administration* guide for more information.

2.10 Shutting Down the System

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

There are several good reasons to stop the system in a controlled shutdown. For example:

- You need to upgrade your software or add new hardware to your configuration. You shut down the system to set up the new additions, make the necessary adjustments to your configuration files, and build a new kernel.

- You have been monitoring the hardware error log and have noticed repeated warnings. 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 check 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 some 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. Refer to 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. Refer to the shutdown(8) reference page for more information.

2.11.1 Using SysMan shutdown

The `sysman shutdown` command invokes a graphical user interface that you can use to shut down a host system. This interface can also be invoked from the SysMan Station or the SysMan Menu. Refer to 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.

Note

The command `sysman shutdown` replaces the `dxshutdown` utility, which has been removed to an obsolete utilities subset. If you type `dxshutdown` at the command line, the new `SysMan shutdown` interface will be invoked.

When you invoke `sysman shutdown`, a window titled `Shutdown Targeted on` *host name* is displayed, where *host name* is the local system name. This utility offers additional options if you are shutting down cluster members. Refer to 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 the single-user (standalone) command prompt
  - Message only – Broadcast a message to all current system users without shutting down the system
- Minutes until shutdown: – hold 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 using the slider bar. Note that in some environments, such as the curses interface 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, and at 30 seconds before shutdown, and at shutdown time.

- Broadcast message to NFS clients – Check this box if you want the message to be broadcast 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 will receive warning of the impending shutdown. The remote user's machine must be running the `rwalld` daemon

- Execute run-level transition scripts – Check this box if you want to run the existing run-level transition scripts in `/sbin/rc[`*023*`.d]/[K`*nn_name*`]`. For example, `/sbin/rc0.d/K45.syslog`. See the `-s` option in the `shutdown`(8) reference page 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 will complete any tasks that you specify prior to shutting down the system. Note that 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 using the `sync` operation.

Once you have initiated a shutdown 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 opt to cancel the shutdown at any time.

Refer to the online help for more information on the various options and the `shutdown`(8) reference page for more information on `shutdown` behavior.

2.11.2 Shutting Down the System and Warning Other Users

You can perform this task 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 following syntax with the `shutdown` command:

   /usr/sbin/shutdown [-bfhknrsc] [-chs] *time* [*warning-message*]

For example, to shutdown and halt the system in 10 minutes with a warning to users that the system is going 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 will be immediately echoed to the console terminal, and to the terminal window from which the command was run.

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

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

5 This final message warns that the system will shut down immediately and user processes will be 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.

Note that 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 (standalone) mode, displaying the console input prompt. Only the root user may now use the system and can perform standalone tasks or use the `halt` command to completely shut down the system.

8 Various messages are displayed as system components are initialized.

9 The console prompt (>>>) is displayed. You can now 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 can also 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 following syntax with the `shutdown` command:

shutdown –h *time* [*warning-message*]

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 standalone (single-user) 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 Owners 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 can also 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 following syntax with the `shutdown` command:

   shutdown -r *time* [*warning-message*]

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, `shutdown` automatically starts the reboot operation, which involves running `fsck` for a consistency check on all mounted file systems. If problems are not encountered, the system reboots to multiuser mode.

Note

If `fsck` finds file system inconsistencies, it displays a warning message, recommending that you run `fsck` 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. You can also invoke the SysMan Menu task `Shut down the system` to complete 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 Owners Manual for your system.

Note

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

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`. These commands should be invoked 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
```

In response to the `halt` command, the program logs the shutdown in the log file, kills all running processes, executes the `sync` system call and waits for all information to be written to disk, then halts the systems. Note that entering the `sync` command at least twice ensures that all data in memory is safely written to disk. Refer to the `halt`(8) reference page for a description of the command and its flags.

Refer to the fasthalt(8), fastboot(8), and reboot(8) reference pages 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, which has the following syntax:

reboot [[-dlnq]]

Use this command to reboot the system as follows:

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

When you run the reboot command without options, it stops all processes, synchronizes (sync) 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. Refer to the reboot(8) reference page for a description of the command and its flags.

2.12.2 Stopping Systems with the fasthalt Command

If you are working single-user mode, you can halt a system immediately using the fasthalt command, which has the following syntax:

fasthalt [-lqn]

Use this command to halt the system as follows:

```
# /usr/sbin/fasthalt
```

When you invoke fasthalt without options, it halts the system and flags a subsequent reboot to skip the execution of fsck. 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 using the syslogd command and places a record of the shutdown in the login accounting file, /var/adm/wtmp.

For a description of the fasthalt command, see the fasthalt(8) reference page.

2.12.3 Stopping Systems with the fastboot Command

If you are working in single-user mode, and do not need to check file systems, you can halt and reboot the systems with the `fastboot` command, which has the following syntax:

fastboot [-l -n -q]

Use this command to halt and reboot the system as follows:

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

When you invoke `fastboot` without options, it creates a file named `/fastboot`, halts the system, then immediately reboots the system without checking file systems with `fsck`.

For a description of the `fastboot` command, see the `fastboot`(8) reference page.

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 already performed some of these tasks. 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 correspondingly.

The following topics are covered in this chapter:

- Section 3.1 describes the system initialization files, which you use to initialize and control the system's run levels.
- Section 3.2 describes how you use the national language directories to provide support for language-specific and country-specific programs.
- Section 3.3 describes the internationalization features, which you tailor to support programmers and users developing and running programs for international audiences
- Section 3.4 describes the system time zone directories and environment variables, which you use to administer local and worldwide time zone information on your system
- Section 3.5 describes the Class Scheduler, a feature that enables you to customize the allocation of CPU resources to user processes.
- Section 3.6 describes power management, which you set up and use to control power consumption in Energy Star-compliant peripherals and processors.
- Section 3.7 describes how you customize swap space. Refer also to the *System Configuration and Tuning* guide as there are implications for performance tuning.

Refer to 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* guide is the principal source of security-related information for users, administrators, and programmers dealing with the security components.

- The *Network Administration* guide is the principal source 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 easily change or customize the system environment 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. Refer to the `init`(8) reference page 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 checking 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

A set of individual directories that correspond to the various run levels. Each directory 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*

The run command script that corresponds 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

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. Refer to the rcmgr(8) reference page and the *Network Administration* manual for more information.

/etc/sysconfigtab

The database file that contains information about the subsystems that can be dynamically configured. Chapter 4 describes this file.

/usr/sbin/getty

The executable file that sets and manages terminal lines. Section 3.1.1.3 and Section 3.1.1.4 describe this program. Refer to the getty(8) reference page for more information.

/etc/gettydefs

The file used by getty that contains entries to identify and define terminal line attributes. Refer to the gettydefs(4) reference page for more information.

/var/spool/cron/crontabs/*

The files that contain entries to identify and define the regular or periodic activation of specific processes. Refer to Section 3.1.3 for more information about these files.

/var/spool/cron/atjobs/*

The file that contains entries to identify and define the once-only activation of specific processes. See the at(1) reference page for more information.

The following files contain information on kernel configuration:

/usr/sys/conf/*NAME*

The 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 with four fields; each field is 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. As long as 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 the inittab(4) reference page for more information.

Command

This 1024-character field holds the sh command to be executed. 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.

If you intend to change or add entries to the /etc/inittab file, make certain that you are familiar with the function and contents of the associated files and run command scripts.

The following sections provide information that will 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 checking 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, the getty program for nonworksystems and the xdm program for worksystems must set up the process that runs the login and shell programs for each terminal and workstation, respectively. 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 with 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 executes at run levels 1, 2, 3, and 4. The respawn keyword tells init to re-create 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 re-created.

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 the terminfo(4) reference page 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 the getty(8), gettydefs(4), and inittab(4) reference pages 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, rc0.d, rc2.d, and rc3.d directories. 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
```

Notice that 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 the `rc0`(8) reference page for additional 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 to a nonnetworked multiuser state, run level 2. 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 2. For example:

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

Notice that 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.

Refer to the rc2(8) reference page 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
```

Notice that 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.

Refer to the rc3(8) reference page 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 2:00AM, 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. Note that 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.

One important function of the file /usr/spool/cron/crontabs/root is to back up and clean system log files to prevent them from growing too large. A set of default cron tasks is configured when you install the operating system.

You can add additional tasks, or modify the existing tasks, to suit your local system requirements. For example, the root crontab file /usr/var/spool/crontabs/root contains three entries that will clean up system log files at 2:00AM every Sunday. One compressed backup of each log file is retained until the next cleaning. These crontab entries are enabled by default and appear in the file as follows:

```
# To get the standard output by email remove the output redirection.
#
0 2 * * 0 /usr/lbin/logclean /var/adm/cron/log > /dev/null
0 2 * * 0 /usr/lbin/logclean /var/adm/wtmp > /dev/null
0 2 * * 0 /usr/lbin/logclean /var/adm/messages > /dev/null
```

Note that the output is directed to /dev/null by default. You can redirect it to e-mail to receive notification when a task finishes. These cron tasks backup the following log files and create a new empty file:

- /var/adm/cron/log logs the activity of cron.
- /var/adm/wtmp logs all user logins on the system
- /var/adm/messages logs the system boot and kernel error messages (including device probe results) that will not be captured in /var/adm/binary.errlog .

The preserved log files are located in /var/adm (/var/cluster/members/{memb}/adm), and are named with the suffix *.bak.gz. For example, wtmp.bak.gz. Use the command gunzip to decompress the files.

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 root crontab entries. You may also 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 only use the following command:

```
# crontab -e
```

The environment variable EDITOR should be set and exported beforehand if an editor other than /usr/bin/ed is desired.

Refer to the crontab(1) reference page 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, that 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, that contain translations of messages 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

Table 3–1: Locale Support Files (cont.)

Language/Territory	Locale Filename
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), character tables (.8859* files), and DEC variants (@DEC 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, refer to the following reference pages:

`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.

Note that this is not a definitive list of all the reference pages that document internationalization. Refer to the See Also section of each reference page, and the *Writing Software for the International Market* manual.

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. The `setlocale` function will then use 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
```

Note that 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 automatically set 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.
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 will 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 of 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 might exist in two versions to sort data two ways: in dictionary order and in telephone-book order. Suppose your site is in France, uses the default French locale, and the standard setup for this locale uses dictionary order. However, your site also needs to use a site-defined locale that collates data in telephone-book order. You might 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 will use 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* and *Installation Guide — Advanced Topics*.
- For information about setting up and maintaining an operating system environment for programmers who write internationalized software, see the guide to *Writing Software for the International Market*.
- To set up and maintain your system for users of internationalized applications, see the System Setup graphical 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, providing that at least one international support software subset is installed. You can also launch this option from the CDE Application Manager. Refer to Chapter 1 for information on using CDE.

3.4 Customizing Your Time Zone

Information about configuring your system's time zone is in Chapter 4. This section describes how to administer local and worldwide time zone information on your system.

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 Canada's Atlantic time zone:

```
# ln -sf /etc/zoneinfo/Canada/Atlantic /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. Refer to the zic(8) reference page for more information.

You can also 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 can also 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:

Jj
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, which is the default time.

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

You can also 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:

```
:US/Eastern
```

Refer to the `tzfile`(4) reference page 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 the offset stored in the kernel tz structure.

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.

Refer to the timezone(3) reference page for more information.

Refer to Chapter 4 for information about configuring a time zone for your system.

3.5 Customizing CPU Resource Allocation

The class scheduler provides you with a method of controlling the execution of tasks or applications by restricting the length of time that they can access the processor (CPU). For example, daemons such as the print spooler can be given less access time. The CPU will then have more time available to perform other tasks. To do this, you specify that the print daemon /usr/lbin/lpd is allowed to use no more than a certain percentage of the available CPU time. You can group resource user identifiers, such as a user's UID (user identification), into classes and assign the required CPU access time to each class.

This feature can help you to allocate system resources so that the most important work receives the required processing time. For example, you may want to run two versions of a production database on your system. One version is used as part of your business operations, while the other is a test copy, with different tuning parameters. The test database can be assigned to a different class so that your daily operations are not impacted by the testing.

To set up and use the class scheduler, you must complete the following steps:

- Plan the allocation of CPU resources
- Use class_admin to set up and maintain the class database
- Create classes and add members to the classes
- Verify class entries using the show command
- Save the entries to the database
- Enable class scheduling to start the daemon

You use the class scheduler commands to monitor and control scheduling as follows:

- Execute class_admin commands such as stat from the command line or a shell script without running an interactive session
- Use the runclass command to execute a task according to the priorities set for a particular class

The following sections suggest a systematic approach to using class scheduling, although it is not be necessary to perform tasks in a specific sequence. There are two methods of accessing the class scheduler:

Manual	By executing class_admin commands from the command line to configure a default database, add classes and class members, and enable the class scheduling daemon to create a quick fix to a CPU resource sharing problem.
Graphical Interface	By using the graphical user interface available as a SysMan Menu sub-option, Class Scheduling, which is available under the Monitoring and Tuning menu option. Refer to Chapter 1 for information on running the SysMan Menu. Section 3.5.6 describes how you use the graphical interface. Refer to the online help for additional information on valid data entries.

The following reference pages contain detailed information on using the class scheduler commands and options:

- class_scheduling(4)
- class_admin(8)
- runclass(1)
- classcntl(2)
- sysman(8)

The following command displays online help for the class_admin command:

```
# /usr/sbin/class_admin help
```

3.5.1 Class Scheduler Overview

To use the class scheduler, you must first create a database file and populate the file with one or more classes. Each class is assigned a CPU value that controls its access to processing time, expressed as a percentage of the total CPU time availability. One or more applications or groups of applications can be assigned to a class, identified according to a unique system process identifier such as:

- UID - User identifier, a unique number assigned to each user account (login)
- GID - Group identifier, a number or name assigned to several user accounts to indicate that they belong to the same group
- PID - Process identifier, a system-assigned number that is unique to each process
- PGID - Process group identifier, a system-assigned number that is unique to each process group
- SESS - Session identifier, a system-assigned number that is unique to each session

Note that the PID, PGID, and SESS identifiers are usually temporary and do not persist across a reboot, ceasing to exist when a task is completed. They are not stored in the database and have no effect when the system or task is restarted.

Once the database is established, you can enable class scheduling to start a class scheduling daemon and put the CPU access restrictions into effect. Other commands enable you to review classes, change contents or scheduling parameters, and delete components or entire classes. When a class scheduling database is configured and enabled, you can:

- Use `runclass` to execute a task (process) according to the CPU access value set for a specific class. For example, you might set a value for interactive operations that is much higher than background processes such as print daemons. To temporarily use the higher value for a print job, you can execute the `lpr` command in the same class as interactive operations.
- Use the `class_admin` command to execute class scheduling commands from within scripts.

3.5.1.1 Related Utilities

The following utilities are also available for use when monitoring and tuning processes:

- The `nice` command
- The Process Tuner (`dxproctuner`) graphical interface, available from the CDE MonitoringTuning folder in the Application Manager - System_Admin
- The `iostat` and `vmstat`, which can be used to monitor resource usage and can be invoked from the SysMan Menu

3.5.1.2 Invoking the Class Scheduler

The class scheduler is provided as both a command-line interface and a graphical user interface. You can invoke the class scheduler several ways , depending on what user environment you are working from:

- From the SysMan Menu, select the Monitoring and Tuning branch, then select the Class Scheduling task.
- From the command line, enter either of the following commands:

  ```
  # sysman class_sched
  ```

  ```
  # sysman -menu "Class Scheduling"
  ```

- From CDE (assuming your system is running a graphics environment with CDE) take the following steps:

  1. Select the Application Manager from the CDE front panel
  2. Select the System_Admin Software Management Group
  3. Select the Configuration Software Management Group
  4. Select the class scheduler icon

Note that the following sections focus on using the command-line method, and provide a brief introduction to using the graphical interface. Refer to the online help for more information in using the graphical interface.

3.5.2 Planning Class Scheduling

How you allocate CPU resources will depend on your system environment and what resources and priorities must be considered. A typical scenario is to assign a higher CPU percentage to interactive tasks so that users do not encounter long response times. Most batch or background processes will be assigned a lower CPU percentage, while some specific background processes may require a higher CPU percentage. For example, if a nightly back up is being performed, you might not want it to have such a low CPU percentage that it does not complete in a reasonable time.

If, however, your system is involved with critical real-time tasks that must take precedence over interactive processes, your course of action may be different. In such cases you should design a baseline that assigns processes to classes. You can then monitor processes and gather user feedback to tune the database by moving tasks from class to class or by changing the CPU access time of the classes.

3.5.3 Configuring Class Scheduling

Use the `class_admin` command to configure an initial database. This command provides:

- An interactive command with subcommands that enables you to create and administer a database of classes. The database is stored in the binary file /etc/class, which cannot be edited manually. Type help at the class> command prompt for a list of options.
- A command mode that allows you to execute class_admin commands at the command prompt, or include commands in shell scripts.

A database must be configured before you can enable class scheduling with the enable command. If a database does not exist when you enter the class_admin command, the command will invoke an interactive session and prompt you to configure a database. If the class_admin command is invoked by a script, a database is configured automatically, using the system defaults.

The following example shows an interactive configuration session using class_admin. Note that in the actual output, the lines will be formatted to fit in 80 columns:

```
# /usr/sbin/class_admin
                    Class Scheduler Administration

configure:

Shall processes that have not been explicitly
assigned to a defined class be assigned to a
'default' class?  Enter (yes/no) [no]: yes

Enforce class scheduling when the CPU is otherwise
idle? (yes/no) [no]: yes

How often do you want the system to reset class usage?
Enter number of seconds (1): 2
class>
```

The configuration values have the following effect:

- To be scheduled, a process must be assigned to a class. If you answer yes to the first prompt, a special class called the default class is created. Any process that has not been explicitly assigned to a defined class will be assigned to the default class.

  If you answer no to this prompt, then only those processes that are explicitly assigned to a defined class will be class scheduled.
- If you answer yes to the second prompt, you allow classes to exceed their allotted CPU time percentage when the system is otherwise idle. If you answer no , classes are restricted to their allotted percentage even if the CPU has no other work.

- The third prompt allows you to set the standard reset time for all classes. For example, if you choose the short default time of 1 second, each class will have more frequent, but shorter opportunities to access the CPU.

  Use a small number (several seconds) if there are interactive jobs subject to class scheduling to give them a quick response time. If only batch jobs are class scheduled, response time is not an issue and larger values may be used.

In the example, a default class was created and all current processes were assigned to that class. Class scheduling will be enforced even when the CPU is idle and class usage will be reset every five seconds.

To review the current configuration, use the following command:

```
class> show
Configuration:
 -Processes not explicitly defined in the database are
  class scheduled.
 -If the processor has some idle time, class scheduled
  processes are not allowed to exceed their cpu percentage.
 -The class scheduler will check class CPU usage every 2
  seconds.

Class scheduler status: disabled  current database: /etc/class

Classes:

 default targeted at 100%:
    class members:
    Every one not listed below
```

The next step in the process is to create classes and populate the classes with system processes such as tasks, daemons, or user accounts using the appropriate identifiers such as UID or SESS.

3.5.4 Creating and Managing Classes

When the database has been configured, you can administer classes as follows:

- Create a class:
  - Add processes to the class
  - Delete processes from a class
- Change the CPU access value (time percentage) of any class
- Destroy an entire class, whether empty or populated
- Show details of class members and configuration settings

- View statistics of actual CPU use against current priority settings

Some of these options are described briefly in the following sections, for detailed descriptions of command options, refer to the online help and reference pages.

3.5.4.1 Creating a Class

To create a class, either use the command mode or enter an interactive session as follows:

```
# class_admin
class> create high_users 50
```

The command mode version is entered as follows:

```
# class_admin create batch_jobs 10
batch_jobs created at 10% cpu usage


changes saved
```

The first command creates a class named `high_users` and assigns a CPU usage restriction of 50 percent. The second command creates a class named `batch_jobs` and assigns a CPU usage restriction of 10 percent. Note that in command mode the changes are automatically saved to the database in `/etc/class`. When making changes to classes interactively, you use the command `save` to commit changes to the database. If you attempt to end the session with the `quit` command and there are unsaved changes, you will be prompted to save or discard the changes before quitting the interactive session as follows:

```
class> quit
Class scheduler database modified.
Save changes? (yes/no) [yes]:yes

changes saved
```

3.5.4.2 Managing Identifier Types Within Classes

Processes that are members of a class are identified by unique system-assigned identifiers that the class scheduler recognizes, such as the PID, GID, or UID.

Once you have created classes, you can add UIDs and GIDs or processes to one or more classes using the `add` command. You must specify the type of identifier (id) used and enter one or more unique identifiers. UIDs and GIDs can be determined from the `/etc/passwd` and `/etc/group` files. Alternatively, you can use the graphical interface Account Manager (`dxaccounts`) to display UID and Group information.

Process identifiers can be obtained from system files or by using a command such as ps. With the ps command, you can determine the values of PID, PGID and SESS. Using the following command, you can display the PID for every process running on the system:

```
# /sbin/ps aj

USER   PID PPID  PGID  SESS JOBC S    TTY          TIME COMMAND
walt  5176 5162  5176  2908   1  S    ttyp1    0:01.30 -sh (csh)
root 12603 5176 12603  2908   1  R  + ttyp1    0:00.05 ps aj
```

See the ps(1) reference page for more information.

The following identifiers are supported:

gid

A group identification number from the /etc/group file. For example if you are adding members to a class, using this number will add all the users that are assigned to the group.

uid

A user identification number from the /etc/passwd file. For example if you are adding members to a class, this number will add only the specific user to which the UID is assigned.

pgrp

A process group identifier. In the output from the ps aj command, see the entries under the PGID table heading in the previous example.

session

A session identifier. In the output from the ps aj command, see the entries under the SESS table heading in the previous example.

pid

The process identifier. In the output from the ps aj command, see the entries under the PID table heading in the previous example.

It is most likely that you will use types uid and gid in your established classes, as these values will persist across a reboot or when class scheduling is stopped and restarted. You can use the account management tools, such as dxaccounts or the Accounts option of the SysMan Menu to list UIDs and GIDs for users and groups. The identifiers associated with types pgrp, session, and pid are temporary, and will not exist on reboot, or when a process terminates.

3.5.4.3 Enabling the Class Scheduler

To enable the class scheduler daemon, you execute the following command:

```
# class_admin enable
Class scheduling enabled and daemon \
/usr/sbin/class_daemon started.
```

To disable the daemon, enter the following command:

```
# class_admin disable
Class scheduling disabled.
```

3.5.4.4 Adding Members to a Class

To add a process to a class, you use the `add` command as shown in the following interactive mode example:

```
class> add batch_jobs uid 234 457 235
```

Note that you must use one of the unique identifiers previously specified and you cannot add the same identifier to a class more than once. The same procedure can be performed in command mode or from a script as follows:

```
# class_admin add batch_jobs uid 234 457 235
uid 234 457 235 added to high_users
```

In command mode, additions to a class are automatically saved to the `/etc/class` database.

3.5.4.5 Deleting Members From a Class

To delete one or more processes from a class, use the `delete` command in interactive or command mode. For example:

```
class> delete high_users uid 11
uid 11 deleted from high_users
```

This example deletes the single UID number 11 from class `high_users`.

3.5.4.6 Other Class Management Options

Consult the `class_admin`(8) reference page for information on the following options:

- Change the priority of a class. For example:

  ```
  class> change batch_jobs 20
  batch_jobs retargeted at 20%
  ```

- Destroy an entire class, whether empty or full. For example:

  ```
  class> destroy high_users
  high_users is not empty.
          to destroy anyway? [yes/no]:yes
  ```

```
high_users destroyed
```

- Loading and saving scheduling databases. For example:

```
class> load database_performance
current database modified and not saved
load new database anyway (destroys changes)? (yes/no) [yes]: \
yes
database database_performance loaded
```

  In this example the presence of unsaved modifications to the current database was detected, and the user was prompted to save the changes.

- View statistics of actual CPU use against current priority settings. For example:

```
class> stats
   Class scheduler status: enabled

class name   target percentage  actual percentage
  high_users      50%               40.0%
  batch_jobs      10%                2.0%
```

3.5.5 Using the runclass Command

Once you have established scheduler classes and enabled class scheduling, you can use the `runclass` command to execute a command in a particular class. If you want a higher CPU percentage than is currently assigned to you, you must have root privileges to use this command. The following command uses the `runclass` command to open a terminal window and assign it to the previously-created `high_users` class:

```
# runclass high_users xterm
```

The following command shows that the `pgrp` number for the terminal process is now identified as a member of that class:

```
# class_admin show
.
.
.
class members:
pgrp 24330     pgrp 24351     pgrp 24373
```

In this example, the identifier for the `xterm` process has been added to the class. You can use the following command to view the running process:

```
# ps agx | grep xterm
```

See the `runclass`(1) reference page for more information.

3.5.6 Using the Class Scheduling Graphical Interface

The class scheduler can be launched from the SysMan menu by selecting the `Class Scheduling` option from the `Monitoring and Tuning` tasks. Alternatively, you can launch it from the Common Desktop Environment (CDE) Application Manager. Refer to Chapter 1 for more information on using the SysMan Menu.

As for the command-line method of using the class scheduler described in preceding sections, the steps involved in initial configuration are as follows:

1. Plan your classes and the processes, users, or groups that will be in each class.
2. Configure and name a database by creating classes and adding them to the database.
3. Define the new database as the current database.
4. Start the class scheduling daemon.

You can complete these steps using the SysMan Menu Class Scheduling main menu option, where the following three suboptions are available:

Configure Class Scheduler

> This is the main option that you use to configure and initialize class scheduling. When you select this option, a window is displayed titled Configure Class Scheduler on *hostname*. From here you can select one of the following options:
>
> - Make Current... – Use this option to choose an existing database and make it the current database. When the system is first used, only the default database is available from the option list. This database is a placeholder and contains no classes. You can modify the default or create new databases, adding options to the list.
> - New... – Use this option to create a new database and add it to the list of optional databases. A data entry window will be displayed for you to name the database and select or create classes.
> - Copy... – Use this option to copy an existing database to a file so that you can use it as a starting point for a new database. You will be prompted to enter a file name and location for the copy.
> - Modify... – Use this option to change the configuration of an existing database. Note that if you want to preserve the original database before modifying it, you should use the Copy... option first.
> - Delete – Use this option to remove databases from the option list. You will not be able to recover these databases once removed.

The New... option is the main option and the one most frequently used. It is described in detail in Section 3.5.6.1. The Modify... option provides an identical interface, which allows you to change existing classes and databases.

The remaining menu options require only a confirmation and do not involve extensive data entry. For example, if you opt to delete a database, you will only be prompted to confirm that the database is to be destroyed.

[Re]Start Class Scheduler

Use this option to start the class scheduling daemon, or restart it if it has been stopped. You will be prompted to confirm your selection.

Stop Class Scheduler

Use this option to stop the class scheduling daemon. You will be prompted to confirm your selection.

3.5.6.1 Creating or Modifying a Database

When you select the New... or Modify... options, a screen is displayed titled Configure Class Scheduler: Create/Modify Scheduling Database. Use the following steps to create a new database:

1. In the Name: field, type the name of the database that you want to create. The name should reflect the function of the database, so that you can easily recognize it when it is displayed in a list of many options. For example, `served_applications`.
2. From the option list titled Available Scheduled Classes, you can select any existing classes. If you are setting up the first database, no classes will be listed and only the New.. option will be available for selection.
3. To create a new class, press the New... button to display the window titled Create a new class. In this window, you complete the following steps:
   - Enter a name for the class in the Class name field. The name should enable you to easily recognize the members of the class. For example, `principal_users`.
   - Move the slider bar adjacent to the CPU allocation label to assign a value for the percentage of CPU time allocated to this class.
   - From the pull-down menu in the Member type field, select the type of identifier you will use to allocate processes to this class. Note that only the Group ID and User ID will persist across reboots. Session, Process group and Process ID identifiers will not persist.

- In the member field, enter the name of the user from the `/etc/passwd` file, a group from the `/etc/group` file, or a process identifier from the output of the following command:

  ```
  # /sbin/ps aj
  ```

- Select the OK button to complete the class entry and return to the previous window, or the Apply button to complete this entry and retain the window to create further classes. Use the Cancel button if you do not want to proceed with the creation of a class.

4. When classes have been created, they appear as entries in the optional list of Available Scheduling Classes. Apart from the class name, the CPU time percentage allocation and member and type are also displayed. You can now select classes to add to the database as follows:
   - Click on a class to highlight it
   - Press the Select button to add the class to the database.
5. When all required classes are selected, press the OK button to create the new database. The new database will be added to the list of Available Scheduling Databases.

You also use the Configure Class Scheduler: Create/Modify Scheduling Database window to perform maintenance and administrative operations on classes as follows:

- Use the Copy... option to copy a class and use it as the base for a new class.
- Use the Modify... option to change characteristics of a class.
- Use the Delete... option to destroy a class and remove it permanently from the Available Scheduling Classes.

To begin using the newly created database, complete the following steps:

1. If the window titled Configure Class Scheduler on *hostname* is not already displayed, invoke the SysMan Menu and select the Configure Class Scheduler option.
2. Highlight the required database by clicking on it, then press the Make Current... button. You will be prompted to confirm or cancel your choice.
3. Press the OK button to return to the SysMan Menu, Class Scheduling options, and select the option titled [Re]Start Class Scheduler. You will be prompted to confirm your choice.

On completing these steps, the class scheduling daemon will be started, using the scheduling database that you specified. To verify and monitor that the database is working as anticipated, use the `show command` at the terminal command line. For example, to view scheduling statistics, enter the following command:

```
# class_admin stats

Class scheduler status: enabled \
current database: /etc/.cl_lab1

class name     target percentage  actual percentage
prio-tasks-lab        10                 10
```

Note that you may need to spend some time monitoring tasks and system performance, and you may need to tune your classes to obtain the required results.

3.6 Customizing Power Management

The operating system contains features that allow you to conserve power on certain systems that have the appropriate hardware. In this release, power management has been extended to additional systems and is enabled on these systems by default. Refer to the 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. Note that some systems may be 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, check whether this feature is enabled.
- Set the CPU power usage. This feature is available only on supported systems. The interface will only show and provide this option if 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, using the following utilities:

- Manage an individual workstation using the X11–compliant graphical user interface `/usr/bin/X11/dxpower` utility. Refer to the online help and the dxpower(8) reference page for information on invoking this interface.

- Use sysconfig and sysconfigdb to load and set kernel attributes. Refer to the sysconfig(8) and sysconfigdb(8) reference pages for a list of command options. Note that this method of power management will be retired in a future release.

3.6.1 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 only be used 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) can be damaged by the activation of the DPMS feature. It is important that you check the specifications for your monitor in the owners manual. Monitors that support DPMS and are put in a power savings state will 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 click on the Help button in the lower right-hand corner of the window.

3.6.2 Using the sysconfig Command and sysconfigdb

You can control power management attributes from the command line by using `sysconfig` to manage the `sysconfigdb` database. For example, you will need to use `sysconfig` if you are activating 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 3.6.2.1 for descriptions of these attributes.

Caution

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

3.6.2.1 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.

- 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 the sysconfigdb(8) reference page for more information.

3.6.2.2 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 the sysconfig(8) reference page for more information.

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

3.6.3 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 will enable a list of management actions for the selected device, one of which may be the power management application dxpower. Selecting this menu item will run dxpower on the device.

3.7 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.

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 have not been referenced before.

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 has not been recently referenced to the swap space. This operation is called modified page writing or a pageout. Note that 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 of its modified virtual pages to swap space. This operation is called a swapout. Swapouts typically occur on systems that are memory constrained.

3.7.1 Related Documentation and Utilities

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

- *Installation Guide* – Describes how to plan for initial swap space, and set up initial swap during installation of Tru64 UNIX.
- *System Configuration and Tuning* – Describes advanced concepts of virtual memory and swap, including strategies for performance tuning that involve swap space configuration.

- The swapon(8) and swapon(2) reference pages provide information on the swapon command, used for creating additional swap space.

The following utilities are used during swap space administration:

- /usr/sbin/diskconfig – This graphical user interface can be used to examine disks to locate unused partitions that can be assigned to swap. Refer to the diskconfig(8) reference page for information on invoking and using diskconfig.
- /usr/bin/X11/dxkerneltuner – This graphical user interface can be used to modify kernel swap attributes in the system configuration file. Refer to the dxkerneltuner(8) reference page for information on invoking and using dxkerneltuner.
- /sbin/sysconfig – This command-line interface can be used to modify kernel swap attributes in the system configuration file. Refer to the sysconfig(8) reference page for information on invoking and using sysconfig.
- /sbin/disklabel – This command-line interface can be used to modify kernel swap attributes in the system configuration file. Refer to the sysconfig(8) reference page for information on invoking and using sysconfig.

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 there is insufficient swap space allocation, the system will be unable to save a crash dump, which can contain valuable information that will assist you in recovering from errors. Refer to Chapter 12 for information on crash dump space requirements.

3.7.2 Allocating Swap Space

Swap space is initially 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 may be provided by system console warning messages, stating that available swap space is depleted. Before adding swap space, check that any sudden lack of space is not due to a system problem. Use the following command, or examine system log and event files to ensure that the swap space is not being used up by runaway processes or unusual user activity:

```
# ps agx
```

If the resulting list of processes looks normal, you may need to add swap space.

Swap space can be added temporarily by running swapon. 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 will verify a disk partition to ensure that you do not write over data or use overlapping partitions. If you have a choice of disks, you will probably 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 only 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
   ```

   Note that you can also 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 will see 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 will be automatically opened when the system is rebooted, or when you use the command:

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

See the swapon(8) reference page 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.

3.7.3 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 will terminate 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 that will be created on your system. Immediate mode requires significantly more swap space than deferred mode because it guarantees that there will be 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 that will 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, you should first use the default amount of swap space and adjust the swap space as needed. Typically, consider using a swap size of about half the size of physical memory.

You should 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 may result in one or more processes being involuntarily terminated.

For more information on virtual memory, refer to the *System Configuration and Tuning* guide.

3.7.4 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 `dxkerneltuner` or `sysconfig` to examine kernel attribute values. You will see 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` or `vm-swap-eager=` – 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 `dxkerneltuner` or `sysconfig` to dynamically modify the attribute.

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
```

4

Configuring the Kernel

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.

You also configure your kernel by tuning certain values stored in it. For example, the kernel contains values that you can adjust to make disk access faster. Modifying values to optimize disk access can improve your system's performance, however it can also affect performance in other areas. Detailed information on system tuning and the interaction of attributes is included in the *System Configuration and Tuning* guide.

The system provides two methods of configuring your kernel: the dynamic method and the static method. Dynamic system configuration entails using commands to configure the kernel. Static system configuration entails modifying system files and rebuilding 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 always be preserved across reboots.

This chapter contains the following information:

- Section 4.1 provides pointers to other relevant documentation, in particular the individual reference pages that document all the attributes for every available kernel subsystem.
- Section 4.2 describes how the kernel is configured at installation time.
- Section 4.3 explains how to determine whether you need to configure your kernel and which configuration method to use, static or dynamic.

- Section 4.4 explains how to configure your system dynamically, using commands or the Kernel Tuner (/usr/bin/X11/dxkerneltuner) graphical user interface.
- Section 4.5 explains how to configure your system statically, by modifying system files and rebuilding the kernel.
- Section 4.6 describes the configuration files.

4.1 Related Documentation and Utilities

The following information sources provide information on system attributes, configuration tools and utilities, and detailed reference information on configuration options:

- Books:
  - The *Installation Guide* and *Installation Guide — Advanced Topics* provide information about initial kernel configuration during installation.
  - The *Network Administration* guide provides information on configuring the network.
  - The *System Configuration and Tuning* guide provides detailed information on system configuration and tuning.
- Reference pages:
  - sys_attrs(5) – Contains information about the many configurable and nonconfigurable 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

    Although there are some attributes that are commonly configured, you should consult the appropriate reference page and the *System Configuration and Tuning* guide before you change 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_bsd_tty(5), sys_attrs_cam(5), sys_attrs_dli(5), sys_attrs_dlpi(5), sys_attrs_gpc_input(5), sys_attrs_inet(5), sys_attrs_io(5), sys_attrs_lfa(5), sys_attrs_lsm(5), sys_attrs_net(5), sys_attrs_netrain(5), sys_attrs_pci(5), sys_attrs_ppp(5), sys_attrs_presto(5), sys_attrs_pwrmgr(5), sys_attrs_sec(5), sys_attrs_snmpinfo(5), sys_attrs_socket(5), sys_attrs_streams(5), sys_attrs_ufs(5), and sys_attrs_vme_vba(7) – Describe the attributes for a single subsystem, such as pwrmgr for power management and ufs for the UFS file system.
- 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 that you use to maintain the kernel subsystem configuration and modify or display kernel subsystem attributes. The sysconfigtab reference page documents the file format of the configuration database. (Use the sysconfigdb utility to manage this configuration database.)
- sysconfigdb(8), stanza(4) – Describe the command line utility that you use to manage the subsystem configuration database. The stanza reference page documents the format of a configuration stanza file. This is a file fragment that will be built into the configuration database when you run sysconfigdb.
- autosysconfig(8) – Describes a utility that you use to maintain the list of dynamic kernel subsystems that are automatically configured.
- cfgmgr(8) – Describes a server that the sysconfig and other utilities use to manage kernel subsystems. See also the kloadsrv(8) reference page, which documents the kernel load server.
- dxkerneltuner(8) – Describes a graphical utility (the Kernel Tuner) that enables you to modify or display kernel subsystem attributes.

- sys_check(8) – Describes the sys_check utility, which checks various system attributes and makes recommendations for their appropriate settings. See Chapter 3 for more information.

4.2 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 4.5 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.

4.3 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 might be internet web servers or databases. System tuning requires that you fully understand the impact of changing kernel attributes so that you do not create an unusable kernel or degrade system performance.

  For example, you might 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 refer to the *System Configuration and Tuning* guide for information on potential impacts on other aspects of system performance before you modify an attribute's value.

Most devices are automatically recognized by the system and configured into the kernel at boot time. (See Chapter 5 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 4.4.

Some subsystems, such as required subsystems, are not dynamically loadable. However, these subsystems might allow you to dynamically configure the value of attributes. 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`), a graphical utility that performs most of the same display and set functions as the `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, invoke the Kernel Tuner from its icon. Refer to the `dxkerneltuner`(8) reference page and the application's online help for more information on using the Kernel Tuner.

  The `dxkerneltuner` utility 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 will be 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. use the `-Q` with the `sysconfig` as follows:

  ```
  # sysconfig -Q ddr
  ddr:
  # sysconfig -Q advfs
  advfs:
  AdvfsCacheMaxPercent = 7
  AdvfsMinFragGrps = 16
  AdvfsMaxFragGrps = 48
  AdvfsAccessMaxPercent = 25
  AdvfsMinFreeAccess = 128
  AdvfsMaxFreeAccessPercent = 80
  AdvfsSyncMmapPages = 1
  AdvfsMaxDevQLen = 24
  AdvfsFavorBlockingQueue = 1
  AdvfsReadyQLim = 16384
  AdvfsDomainPanicLevel = 1
  AdvfsCacheHashSize = 1024
  ```

  This example shows that `ddr` has no configurable attributes and `advfs` has several.

If you decide to configure attributes of these subsystems, use the procedures described in Section 4.4.8. 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 might also 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 refer to Section 4.4.

If the subsystem you want to add, remove, or configure does not support dynamic configuration, you must use the static configuration method. You must also use this method to configure system parameters that do not support dynamic configuration. For information about the static configuration method, see Section 4.5.

4.4 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. You must also 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 4.4.7

If you are writing a loadable device driver or other loadable subsystem, refer to 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.

4.4.1 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 (unconfigure) command option. For example, to configure the environmental monitoring envmon subsystem, enter the following command:

```
# /sbin/sysconfig -c envmon
```

4.4.2 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
```

4.4.3 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.

4.4.4 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 might 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 Chapter 5 and the dsfmgr(8) reference page for more information on device special files.

4.4.5 Maintaining the List of Automatically Configured Subsystems

The system determines which subsystems to configure into the kernel at system reboot time by checking 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 will configure 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
```

4.4.6 Managing Subsystem Attributes

To improve the performance or behavior of a subsystem, or of the system as a whole, you might modify the values of subsystem attributes. You can make such modifications using `sysconfig`, `sysconfigdb`, or the Kernel Tuner (`dxkerneltuner`). Under certain circumstances, such as recovering a crashed system, you may also need to use the debugger `dbx` to examine and change the attributes in a damaged kernel. Refer to the *Kernel Debugging* guide 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 4.4.8. 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 `dxkerneltuner` graphical utility, specifying and saving the change using the Boot Time Value field.
- For temporary modifications that will not persist across reboots, use `sysconfig -r` at the command line. Alternatively, use the `dxkerneltuner` graphical utility, 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, you should only use the sysconfigdb command or the dxkerneltuner interface to make changes.

See the sysconfig(8), sysconfigdb(8), sysconfigtab(4), and dxkerneltuner(8) reference pages 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 4.5.

You can manage dynamic subsystem attributes either locally or remotely. For information on how to use the /sbin/sysconfig command remotely, see Section 4.4.7.

4.4.6.1 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 4.4.2.

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.)

4.4.6.2 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 performed 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 initially loaded.
- R – You can modify the attribute while the subsystem is running.
- Q – You can query the attribute for information.

4.4.6.3 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 later store it in the subsystem attribute database as described in Section 4.4.8. (Refer to the dbx(1) reference page and the *System Configuration and Tuning* guide for information on using dbx.)

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 dxkerneltuner.

4.4.7 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 ability 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.

4.4.8 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 dxkerneltuner graphical utility to make the change. Refer to the online help for dxkerneltuner for more information on using that method.

Note

> The /etc/sysconfigtab database might 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 *Guide to Preparing Product Kits*. 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. (For information about stanza-format files, see stanza(4)). 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 4.4.6.3.)

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.

4.4.8.1 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
```

4.4.8.2 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
```

4.4.8.3 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
```

4.4.8.4 Updating Attributes in the Database

To update the entire definition of a subsystem that is already in the /etc/sysconfigtab database, enter the /sbin/sysconfigdb –u command.

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
```

4.4.8.5 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 in more than one subsystem from /etc/sysconfigtab database 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
```

4.4.8.6 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.

4.5 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 4.5.3. If you need to edit the target configuration file before you build a new kernel, refer to Section 4.5.1.
- 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 4.5.2.

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. You can also edit this file to change the value of system parameters. For details about the contents of the target configuration file, see Section 4.6.
- 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 4.6.2.

For information about running the `/usr/sbin/doconfig` program to build a kernel after editing system files, see Section 4.5.3.

For examples of adding and configuring devices, refer to Chapter 5.

4.5.1 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 once you have edited the target configuration file. The procedure assumes that you do not know the appropriate keyword to add. In some cases, you might 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 4.5.3.

Warning

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. Check that you have a copy of the generic kernel, `/genvmunix` , as you will 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. Refer to 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, check 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 4.5.3.

   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 the `strip`(1) reference page.

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 aleady knows the new device.

   To boot the generic kernel, enter the following command:

```
>>> boot -fi "genvmunix"
```

6. At the single-user mode prompt, check 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, check 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. For more information, see Section 4.2.) 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
```

   Check the differences between these files until you find the new device definition keyword. (The two files might 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. For more information, see inittab(4) and securettys(4).

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). Refer to 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. Check 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.

4.5.2 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 4.6

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. Check 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.

4.5.3 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 4.6

   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. Check 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:

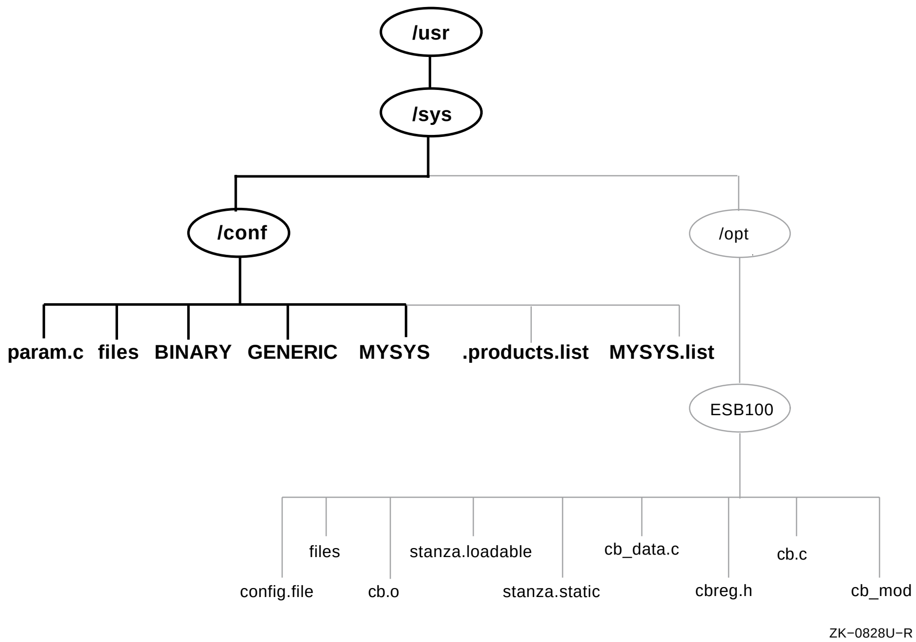
```
# shutdown -r now
```

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 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.

4.6.1 Configuration Files in /usr/sys/conf

The `/usr/sys/conf` directory contains two important system configuration files:

- 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

- 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. You can also 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` command. For more information about building a kernel using an existing configuration file, see Section 4.5.3.

Note

Never delete the `/usr/sys/conf/GENERIC` file.

4.6.2 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 will 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 also create the file by copying the .product.list file to the *NAME*.list file. You can then 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.

Refer to the device driver documentation for more information on the *NAME*.list and .product.list files.

4.6.3 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.

4.6.4 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. Refer to the *System Configuration and Tuning* guide for information on configuring a kernel and tuning it.

5

Hardware Management

This chapter describes the utilities available to assist you in administering the system hardware, which consists of the CPUs and all associated devices The utilities work on single systems and on systems joined into clusters. Hardware management involves viewing the status of system devices and performing administrative options on them if necessary. This includes adding and removing devices, troubleshooting any devices that are not working, and monitoring devices to prevent problems before they occur.

You may also need to administer the software that is associated with devices, such as drivers, kernel pseudodevices and device special files. This software enables devices to communicate and transfer data between system components. Information on administering the related software components is included in this chapter.

Most operations require root user privileges, however you can assign such privileges to nonroot users using the SysMan division of privileges (DOP) feature. See the `dop`(8) reference page for more information.

This chapter contains the following sections:

- Section 5.1 provides a conceptual overview of hardware management and relates it to the organization of information in this chapter.
- Section 5.2 lists other documentation resources that apply to hardware management, including reference pages for commands and utilities. It also identifies key system files and provides pointers to utilities that are associated with hardware administration.
- Section 5.3 describes the SysMan hardware management options.
- Section 5.4 describes the hardware manager command-line utility `hwmgr`. This utility provides full access to hardware management options.
- Section 5.5 describes how to use `dsfmgr` to manage device special files.
- Section 5.6 describes how to manually add devices that cannot be added using hardware manager, and how you create pseudodevices.
- Section 5.7 describes targeted utilities for hardware device management.

5.1 Understanding Hardware

A hardware device can be any part, or component, of a system. The system is organized in a hierarchy with the CPUs at the top, and discrete devices such as disks and tapes at the bottom. This is sometimes also referred to as the system topology. The following components are typical of the device hierarchy of most computer systems although it is not a definitive list:

- The central processing unit (CPU), which may be a single processor system, a multiprocessor system, or a set of processors joined into a cluster. The system is sometimes referred to as a host in the context of hardware management and has a designated host name and perhaps also a host address if the system is on a network. You will often specify commands using the host name. The CPUs are the top of the system hardware hierarchy, and all other system devices are organized under it in the hardware hierarchy.

  Typical administrative tasks associated with the CPU are many, such as bringing CPUs online, starting and stopping them, or sharing CPU resources. These tasks are documented throughout this guide, such as Chapter 2, which documents the options for shutting down the system.

- Buses – A system may have a number of main internal communication buses, which transfer data between all devices on the system. Adapters and controllers are physically plugged into buses and have both physical and logical addresses.

  Buses may have special software associated with the physical bus, but that software is usually managed within the context of the UNIX operating system. For example, when adding an option card such as a sound or network card to a PCI bus, you have to shut down the system add the hardware and reboot. Such devices are often automatically recognized and added to the system configuration on reboot, but you may need to run a firmware utility to install a driver for the device. Always consult your system documentation and the documentation that comes with the card for information on adding such devices.

- Controllers and Adapters – A system may have a number of controllers such as SCSI controllers, which control one or more devices. There may be other controllers, such as the floppy disk interface (`fdi`) which support only one kind of device and often only one physical disk attached to the controller. A network adapter may be connected to a bus, but will not have any other devices attached to it other than the network.

  Adapters occupy a physical slot on a bus, which gives them both a logical address and a physical location to administer. They may also provide slots for devices, which also have physical and logical addresses.

- Devices are usually the lowest entities in the system hierarchy, such as SCSI disks, CDROM readers, and tapes. They are typically attached to a controller or adapter, and often have both a physical location and a logical address to administer.

Devices can also be shared between hosts and between other system components. This means that a device may have different names and identifiers associated with it. Understanding how to identify a device and how that device appears to the rest of the hierarchy is an important aspect of hardware management and you often need to know the logical and physical locations of devices.

When referring to SCSI devices in this chapter, the SCSI disk is the device that is most frequently used as an example. Typically, it is the device that is most often the object of management tasks and may appear to the system as a single device, or as a group:

Single SCSI Disk or `RZ` devices

> Small Computer System Interface (SCSI) technology is an interface standard to which disks must conform if they are to be supported on the operating system. Note that not all SCSI devices closely conform to this standard and may not be automatically detected and added during a boot, or when using `hwmgr` to add a device dynamically. You may need to use `ddr_config` to add such devices as described in Section 5.6.

SCSI RAID or `HSG` and `HSZ` devices

> The Redundant Array of Inexpensive Disks (RAID) technology. These are storage boxes that contain several connected SCSI disks, appearing to the system as a single device. They may support features such as hot-swapping.

Refer to the `RAID`(7), `SCSI`(7) and `rz`(7) reference pages for more information on device characteristics. Refer to the `tz`(7) reference page for more information on tape devices. Refer to the *Technical Overview* and the *Software Product Description* for the current supported standards for RAID and SCSI.

Hardware management involves understanding how all the components relate to each other, how they are logically and physically located in the system topology, and how the system software recognizes and communicates with components. To better understand the component hierarchy of a system, refer to Chapter 1 for an introduction to the SysMan Station. This is a graphical utility that displays topological views of the system component hierarchy and allows you to manipulate such views.

Fortunately, the vast majority of hardware management is automated. When you add a device such as a SCSI disk to a system and reboot the system, it will find the device and recognize it, building in any device drivers that it needs. The system will automatically create the software components for that disk as device special files. It only remains for the administrator to partition the disk as needed and create a file system on the partitions (described in Chapter 6) before it can be used to store data. However, you will periodically need to perform some tasks manually, such as when a disk crashes and you need to bring a duplicate device online at the same logical location. You may also need to manually add devices to a running system or redirect the I/O for one disk to another disk. This chapter focuses on these manual tasks.

Many other hardware management tasks are part of regular system operations and maintenance, such as repartitioning a disk or adding an adapter to a bus. Often, such tasks are fully described in the hardware documentation that accompanies the device itself, but you will often need to perform tasks such as checking the system for the optimum (or preferred) physical and logical locations for the new device.

Another important aspect of hardware management is preventative maintenance and monitoring. You should be aware of the following operating system features that can facilitate a healthy system environment:

- The Event Manager, (EVM) – A utility for filtering and displaying all system events and then presenting those events to the administrator. It includes sophisticated features for warning you of problems by electronic mail or a pager. Refer to Chapter 13 for information on configuring EVM.
- The SysMan Station – A graphical utility that enables you to view and monitor the entire system (or cluster) hardware and launch applications to perform administrative tasks on any device. These applications can also be launched from the SysMan Menu, and some example applications are described later in this chapter (see Section 5.3). For information on using the SysMan utilities, refer to Chapter 1.
- The system census utility, `sys_check` – This utility provides you with data on your system's current configuration as an HTML document that you can read with a Web browser. You can use the data as a system baseline, perform tuning tasks, and check all log files. The Storage configuration section provides information on storage devices and file systems. Refer to Chapter 3 for information on running this utility, and on configuring it to run regularly.
- Insight Manager – An enterprise-wide, Web-based management tool that enables you to view system and component status anywhere in your local area network. It includes launch points for the SysMan Station, the SysMan Menu and the system census utility, `sys_check`. Refer to Chapter 1 for information on configuring and using Insight Manager.

The organization of this chapter reflects the hardware and software components that you manage as follows:

- Generic hardware management utilities – These utilities enable you to perform operations on all devices of a type, classes of devices such as SCSI tapes, or individual devices. The utilities may in some cases operate on all systems in a cluster. An example of such a utility is the SysMan Station, which provides you with a graphic display of the entire component hierarchy for all members of a cluster.
- Software management – This involves the administration of the software that is associated with hardware components on the system, principally managing the device special files. These are the files associated with a hardware device that enable any application to access the device driver or pseudo-driver for that device.
- Targeted hardware management utilities – These utilities enable you to perform operations that are targeted to a specific device and perform a specific task. An example is the disk configuration command line interface, `disklabel` and the analogous graphical interface, Disk Configuration (`diskconfig`). which enable you to partition a disk using the standard layouts or your own custom layouts.

Another way to think of this is that with a generic utility you can perform a task on many devices, while with a targeted utility you can only perform a task on a single device. Note that unless stated, most operations can be performed on a single system or a cluster. You should refer to the TruCluster documentation for additional information on managing cluster hardware.

5.1.1 Logical Storage Manager

The Logical Storage Manager (LSM) consists of physical disk devices, logical entities, and the mappings that connect both. LSM builds virtual disks, called volumes, on top of UNIX system disks. A volume is a special device that contains data managed by a UNIX file system, a database, or other application. LSM transparently places a volume between a physical disk and an application, which then operates on the volume rather than on the physical disk. A file system, for instance, is created on the LSM volume rather than a physical disk.

The LSM software maps the logical configuration of the system to the physical disk configuration. This is done transparently to the file systems, databases, and applications above it because LSM supports the standard block device and character device interfaces to store and retrieve data on LSM volumes. Thus, you do not have to change applications to access data on LSM volumes.

Refer to the manual *Logical Storage Manager* for more complete information on LSM concepts and commands.

5.2 Reference Information

The following sections contain reference information related to documentation, system files and other utilities. Some utilities described here are obsolete and will be removed in a future release. Consult the *Release Notes* for a list of utilities that are scheduled for retirement. If you are using one of these utilities, you should migrate to its replacement as soon as possible. Check your site-specific shell scripts for any calls that may invoke an obsolete utility.

5.2.1 Related Documentation

The following documentation contains information hardware management:

- Books
  - Device documentation – Consult the device documentation (Owners Manual or User Guide) for information on installing the device and for any required operating system settings. The device documentation will provide information that you may need, such as driver files and configuration settings.
  - *Network Administration* – Provides information on configuring or connecting network devices.
  - Device Driver Documentation Kit – Contains related documents such as: *Writing PCI Bus Device Drivers* and *Writing Device Drivers: Reference*.
- Reference pages
  - `hwmgr`(8) – Contains complete information on the command syntax for the hardware manager utility, `/sbin/hwmgr`.
  - `dsfmgr`(8) – Contains complete information on the command syntax for the device special file management utility, used to create device special files in the `/dev` directory. Refer also to Section 5.5.
  - `mknod`(8), `MAKEDEV`(8), `scsimgr`(8), `scu`(8), `ddr_config`(8), and `devswmgr`(8)

Note that most command line and graphical utilities also provide extensive online help.

5.2.2 Identifying Hardware Management System Files

The following system files contain static or dynamic information that is used to configure the device into the kernel. You should not edit these files manually even if they are ASCII text files. Some files may be Context Dependent Symbolic Links, as described in Chapter 6. If the links are accidentally broken, the files may not be usable in a clustered environment until the links are re-created.

- The /dev directory contains device special files. Refer to Section 5.5 for more information.
- /etc/ddr_dbase – The DDR (device dynamic recognition) device information database. The contents of this file is compiled into the binary file/etc/ddr.db, which is used by the system to obtain device information.
- /etc/dec_devsw_db – This is a binary database owned by the kernel dev switch code. This database keeps track of the driver major numbers and driver switch entries.
- /etc/disktab – This file specifies the disk geometry and partition layout tables. This file is useful for identifying disk device names and certain disk device attributes.
- /etc/dvrdevtab – This file specifies the database name and the mapping of driver names to special file handlers.
- /etc/gen_databases – A text file that contains the information required to convert a database name to a database file location and a database handler.
- /etc/dec_hw_db – This is a binary database that contains hardware persistence information. Generally, this refers to hardware such as buses or controllers.
- /etc/dec_hwc_ldb – This is a binary database that contains information on hardware components that are local to a cluster member.
- /etc/dec_hwc_cdb – This is a binary database that contains information on hardware components that are shared by all members of a cluster. Hardware components with unique cluster names or mapped dev_t are stored in this database.
- /etc/dec_scsi_db – This is a binary database owned by SCSI/CAM. It stores the world-wide identifier (WWID) of SCSI devices and enables CAM to track all SCSI devices that are known to the system.
- /etc/dec_unid_db – This is a binary database that stores the preceding highest hardware identifier (HWID) assigned to a hardware component. This database is used to generate the next HWID to be assigned to a newly-installed hardware component.

5.2.3 WWIDs and Shared Devices

SCSI device naming is based on the logical identifier (ID) of a device. This means that the device special filename has no correlation to the physical location of a SCSI device. UNIX uses information from the device to create an identifier called a world-wide identifier, which is usually written as WWID.

Ideally, the WWID for a device is unique, enabling the identification of every SCSI device attached to the system. However, some legacy devices (and even some new devices available from third-party vendors) do not provide the information required to create a unique WWID for a specific device. For such devices, the operating system will attempt to generate a WWID, and in the extreme case will use the device nexus (the SCSI bus/target/lun) to create a WWID for the device.

Consequently, devices that do not have a unique WWID should not be used on a shared bus. If a device that does not have a unique WWID is put on a shared bus, a different device special file will be created for each different path to the device. This can lead to data corruption if two different device special files are used to access the device at the same time. To determine if a device has a cluster unique WWID, use the following command:

```
# hwmgr -show components
```

If a device has the `c` flag set in the `FLAGS` field, then it has a cluster-unique WWID and can be placed on a shared bus. Such devices are cluster-shareable because they can be put on a shared bus within a cluster.

Note

An exception to this rule are HSZ devices. Although an HSZ device might be marked as cluster shareable some firmware revisions on the HSZ preclude multi-initiators from probing the device at the same time. Refer to the owners manual for the HSZ device and check the *Release Notes* for any current restrictions.

The following example displays all the hardware components of category disk that have cluster-unique WWIDs:

```
# hwmgr -show comp -cat disk -cs
HWID: HOSTNAME FLAGS SERVICE COMPONENT NAME
-----------------------------------------------
35:   pmoba    rcd-- iomap   SCSI-WWID:0410004c:"DEC  RZ28     ..."
36:   pmoba    -cd-- iomap   SCSI-WWID:04100024:"DEC  RZ25F    ..."
42:   pmoba    rcd-- iomap   SCSI-WWID:0410004c:"DEC  RZ26L    ..."
43:   pmoba    rcds- iomap   SCSI-WWID:0410003a:"DEC  RZ26L    ..."
48:   pmoba    rcd-- iomap   SCSI-WWID:0c000008:0000-00ff-fe00-0000
49:   pmoba    rcd-- iomap   SCSI-WWID:04100020:"DEC  RZ29B    ..."
50:   pmoba    rcd-- iomap   SCSI-WWID:04100026:"DEC  RZ26N    ..."
```

In some rare cases you may have a device that does not supply a unique WWID but you have a requirement that it must be available on a shared bus. Using such devices on a shared bus is not recommended but there is a manual command that will allow you to set up one of these devices to be used on a shared bus. See Section 5.4.4.10 for a description of how to use the `hwmgr -edit scsi` command option.

5.2.4 Related Utilities

The following utilities are also available for use in managing devices:

- The system exerciser utilities enable you to test devices for correct operation. See the `diskx`(8), `tapex`(8), `cmx`(8), `fsx`(8), and `memx`(8) reference pages. See also Chapter 12.
- The `scu` utility can be used to maintain and diagnose problems with SCSI peripherals and the CAM I/O subsystem. Refer to the `scu`(8) reference page and the online help for the command.
- The `sysconfig` command is used to query or modify the kernel subsystem configuration. You use this command to add subsystems to your running kernel, reconfigure subsystems already in the kernel, ask for information about (query) subsystems in the kernel, and unconfigure and remove subsystems from the kernel. You can use `sysconfig` to set some device attribute values. For information on using `sysconfig`, refer to Chapter 4 which also documents the Kernel Tuner, `dxkerneltuner`, a graphical utility that you can use to modify attribute values.
- CDE Application Manager – SysMan Applications pop-up and System_Admin folders contain several hardware management tools, for example:
  - Configuration – Graphical utilities used to configure hardware such as ATM, Disk devices, Network devices, PPP (modem) devices, and LAT devices.
  - DailyAdmin – A graphical utility for power management, which can be used to set power attributes for certain devices.
  - SysMan Checklist, SysMan Menu, and SysMan Station, provide interfaces to configure, monitor, and maintain system devices. The SysMan Menu and SysMan Station can be run from a variety of platforms, such as a personal computer or an X11–based environment. This enables you to perform remote monitoring and management of devices. Refer to Chapter 1 for information.

5.3 Using the SysMan Hardware Utilities

The SysMan Menu `Hardware` branch provides utilities for hardware management. You can also use the SysMan Station to obtain information about hardware devices and to launch hardware management utilities.

The SysMan utilities provide you with a subset of the many more hardware management features available from the command line when you use the hwmgr command. A more detailed discussion of the hwmgr command and its options can be found in Section 5.4. See also the hwmgr(8) reference page for a complete listing of the command syntax and options. Selecting the help option in one of the SysMan Menu hardware tasks will invoke the appropriate reference pages.

When you invoke the SysMan Menu as described in Chapter 1, hardware management options are available under the Hardware branch of the menu. Expanding this branch displays the following tasks:

- View hardware hierarchy
- View cluster
- View device information
- View central processing unit (CPU) information

These tasks launch SysMan Menu utilities that are described in the following sections. The first three utilities run instances of the /sbin/hwmgr command to obtain and display system data. Note that the utilities provide a method of finding the data that you use when specifying hardware management operations on system components. For example, finding out which disks are on which SCSI buses.

The following option buttons (or choices, in a terminal) are available in all the utilities:

- Rerun – Runs the utility again, updating the display.
- Stop – Stops the utility. Use the Rerun option to update the display or choose OK to exit the utility.
- OK – Ends the task and closes the window.
- Help – Displays the reference page.

5.3.1 Viewing the Hardware Hierarchy

The View hardware hierarchy task invokes the command /sbin/hwmgr -view hierarchy, directing the output to the SysMan Menu window (or screen, if a terminal). The following example shows output from a single-CPU system that is not part of a cluster:

```
                View hardware hierarchy

 HWID:  hardware component hierarchy
 ---------------------------------------------------
    1:  platform AlphaServer 800 5/500
    2:    cpu CPU0
    4:    bus pci0
    5:      connection pci0slot5
   13:        scsi_adapter isp0
```

```
 14:          scsi_bus scsi0
 30:            disk bus-0-targ-0-lun-0 dsk0
 31:            disk bus-0-targ-4-lun-0 cdrom0
  7:      connection pci0slot6
 15:        graphics_controller trio0
  9:      connection pci0slot7
 16:        bus eisa0
 17:          connection eisa0slot9
 18:            serial_port tty00
 19:          connection eisa0slot10
 20:            serial_port tty01
 21:          connection eisa0slot11
 22:            parallel_port lp0
 23:          connection eisa0slot12
 24:            keyboard PCXAL
 25:            pointer PCXAS
 26:          connection eisa0slot13
 27:            fdi_controller fdi0
 28:              disk fdi0-unit-0 floppy0
 11:      connection pci0slot11
 29:        network tu0
```

Use this task to display the hardware hierarchy for the entire system or cluster. The hierarchy shows every bus, controller, and device on the system from the CPUs down to the individual peripheral devices such as disks and tapes. On a system or cluster that has many devices, the output can be lengthy and you may need to scroll the display to see devices at the beginning of the output.

The output is useful because it provides you with information that is used in many `hwmgr` command options to perform hardware management operations such as viewing more device detail and adding or deleting devices. The following items shown in the hierarchy can be used as command input:

- HWID – The hardware identifier (or `id`), an integer that is unique to every individual entry in the hierarchy.
- The device name, such as `pci` for the Peripheral Component Interconnect (PCI) bus.
- The device basename, a mnemonic followed by an integer that identifies the device such `cdrom0`, which relates to the device special file for the device (`/dev/disk/cdrom0`). More information on device special file names can be found in Section 5.5.
- The physical location attribute specifies the address or path to a device, such as `bus-0-targ-0-lun-0`, sometimes written as `0/0/0`, which provides the following information:
  – `scsi-0` is the `bus` and provides number of the bus to which the device is attached.

– targ-0 is the target number for this device on the bus, in this case the first target on bus 0.

– lun-0 is the logical unit number or lun, in this case the first logical unit number at target 0 on bus 0.

- The hardware category of a device, such as a bus or ide_controller.
- Connections to slots, which show the slot number for a device, such as pci0slot5 and eisa0slot9.
- Bus, controller, and device relationships, such as the following section showing two disk devices on controller scsi_adapter isp0 which is on the bus scsi_bus scsi0:

```
13:        scsi_adapter isp0
14:          scsi_bus scsi0
30:            disk bus-0-targ-0-lun-0 dsk0
31:            disk bus-0-targ-4-lun-0 cdrom0
```

Note that because the same device might be shared (for example, on a shared bus) it may appear in the hierarchy more than once and will have a unique identifier each time it appears. An example of this is given in Section 5.4.4.7.

You can use the information from the -view hierarchy command output in other hwmgr commands when you want to focus an operation on a specific hardware component, as shown in the following command, which gets the value of a device attribute named device_starvation_time for the device with the HWID (id) of 30. Device 30 is the disk device at bus 0, target 0 and lun 0 in the example hierarchy:

```
# /sbin/hwmgr -get attr -id 30 -a device_starvation_time
30:
  device_starvation_time = 25 (settable)
```

The output shows that the value of the device_starvation_time attribute is 25. The label (settable) indicates that this is a configurable attribute that you can set using the following command option:

```
# /sbin/hwmgr -set attr
```

5.3.2 Viewing the Cluster

Selecting the View cluster task invokes the command /sbin/hwmgr -view cluster, directing the output to the SysMan Menu window (or screen, if a terminal) as follows:

```
                                     View cluster

Starting /sbin/hwmgr -view cluster ...

/sbin/hwmgr -view cluster run at Fri May 21 13:42:37 EDT 1999
```

```
        Member ID         State          Member HostName
        ---------         -----          ---------------
             1             UP                  rene (localhost)
            31             UP                  witt
            10             UP                  rogr
```

If you attempt to run this command on a system that is not a member of a cluster, the following message is displayed instead of the system listing:

```
hwmgr: This system is not a member of a cluster.
```

The `Member ID` and the `HostName` can be specified in some `hwmgr` commands when you want to focus an operation on a specific member of a cluster, as shown in the following example:

```
# /sbin/hwmgr -scan scsi -member witt
```

5.3.3 Viewing Device Information

Selecting the `View device information` task invokes the command `/sbin/hwmgr -view devices`, directing the output to the SysMan Menu window (or screen, if a terminal).

Use this option to display the device information for the entire system or cluster. The output shows every device and pseudodevice (such as `/dev/kevm`) on the system. The following example shows the output from a small single-CPU system that is not part of a cluster:

```
                            View device information

Starting /sbin/hwmgr -view devices ...

/sbin/hwmgr -view devices run at Fri May 21 14:20:08 EDT 1999

HWID:  Device Special File  Mfg Model          Location
                      Name
------------------------------------------------------------------
 3:           /dev/kevm
28:   /dev/disk/floppy0c        3.5in floppy    fdi0-unit-0
30:      /dev/disk/dsk0c   DEC  RZ1DF-CB(C)DEC  bus-0-targ-0-lun-0
31:    /dev/disk/cdrom0c   DEC  RRD47  (C)DEC   bus-0-targ-4-lun-0
```

For the purpose of this command, a “device” is considered to be any entity in the hierarchy that has the attribute `dev_base_name` and as such will have an associated device special file (DSF). The output from this utility provides the following information which can be used with the `hwmgr` command to perform hardware management operations on the device:

- HWID – The hardware identifier (or `id`), an integer that is unique to every individual entry in the hierarchy,

- The DSF Name, such as /dev/disk/cdrom0c. In the case of disk devices, this is the name of the device special file associated with the c partition that maps to the entire capacity of the disk. For a tape, it will show the device special file name that maps to the default density for the device. See Section 5.5 for a description of these names.
- The model, which specifies a manufacturer model number or a generic description such as 3.5in floppy.
- The physical location of a device, such as the SCSI bus-0-targ-0-lun-0, sometimes written as 0/0/0, which specifies the following:
  - bus-0 – The number of the bus to which the device is attached, in this case, it is SCSI bus 0.
  - targ-0 – The target number for this device on the bus, in this case the first target on the bus.
  - lun-0 – The logical unit number, or lun), in this case the first on the bus.

  The previous output also shows a floppy disk attached to the floppy disk interface, fdi as device 0, unit 0.

You can specify this information to certain hwmgr commands to perform hardware management operations on a particular device. The following example of disk location specifies a device special file for a disk, causing the light (LED) on that disk to flash for 30 seconds. This tells you exactly which device special file is associated with that disk.

```
# /sbin/hwmgr -flash light -dsf /dev/disk/dsk0c
```

5.3.4 Viewing CPU Information

Selecting the View central processing unit (CPU) information task invokes the command /usr /sbin/psrinfo -v, directing the output to the SysMan Menu window (or screen, if a terminal). Use this option to display the CPU status information, as shown in the following sample output for a single-processor system.

The output from this utility describes the processor and tells you how long it has been running, as follows:

```
                               /usr/sbin/psrinfo
Starting /usr/sbin/psrinfo -v ...

/usr/sbin/psrinfo -v run at Fri May 21 14:22:05 EDT 1999

Status of processor 0 as of: 05/21/99 14:22:05
  Processor has been on-line since 05/15/1999 14:42:28
   The alpha EV5.6 (21164A) processor operates at 500 MHz,
    and has an alpha internal floating point processor.
```

5.3.5 Using the SysMan Station

The SysMan Station is a graphical utility that runs under various windowing environments or from a web browser. Refer to Chapter 1 and the online help for information on launching and using the SysMan Station.

Features of the SysMan Station that assist you in hardware management are as follows:

Monitoring systems and devices

> The SysMan Station provides a live view of system and component status. You can customize views to focus on parts of a system or cluster that are of most interest to you. You will be notified when a hardware problem occurs on the system. System views are hierarchical, showing the complete system topology from CPUs down to discrete devices such as tapes. You can observe the layout of buses, controllers, and adapters and see their logical addresses. You can see what devices are attached to each bus or controller, and their slot numbers. Such information is useful for running `hwmgr` commands from the command prompt.

Viewing device properties (or attributes)

> You can select a device and view detailed attributes of that device. For example, if you select a SCSI disk device and press the right mouse button, a menu of options is displayed. You can choose to view the device properties for the selected disk. If you opt to do this, an extensive table of device properties will be displayed. This action is the same as using the `hwmgr` command, as shown in the following (truncated) sample output:

```
# hwmgr -get attr -id 30
30:
  name = SCSI-WWID:0c000008:0060-9487-2a12-4ed2
  category = disk
  sub_category = generic
  architecture = SCSI
  phys_location = bus-0-targ-0-lun-0
  dev_base_name = dsk0
  access = 7
  capacity = 17773524
  block_size = 512
  open_part_mask = 59
  disk_part_minor_mask = 4294967232
  disk_arch_minor_mask = 4290774015
<display truncated>
```

Launching hardware management utilities

When you select a device, you can also choose to launch a utility and perform configuration or daily administrative operations on the selected device. For example, if you select a network adapter, you can configure its settings or perform related tasks such as configure the domain name server (DNS). You can launch the Event Viewer to see if any system events (such as errors) pertaining to this device have been recently posted.

Note that you can also run the SysMan Station from within Insight Manager and use it from a PC, enabling you to remotely manage system hardware. Refer to Chapter 1 for more information on remote management options.

5.4 Using hwmgr to Manage Hardware

The principal generic utility used for managing hardware is the `hwmgr` command line interface (CLI). Other utilities, such as the SysMan utilities only provide a limited subset of the features provided by `hwmgr`. For example, you can use `hwmgr` to set an attribute for all devices of a particular type (such as SCSI disks) on all SCSI adapters in all members of a cluster.

Most hardware management is performed automatically by the system and you need only intervene under certain circumstances, such as replacing a failed device so that the replacement device takes on the identity of the failed device. The following sections provide information on:

- Understanding the hardware management model
- Understanding the principal user options available in `hwmgr`
- Performing administrative tasks using `hwmgr`

5.4.1 Understanding the Hardware Management Model

Within the operating system kernel, hardware data is organized as a hardware set managed by the kernel set manager (KSM). Application requests are passed by library routines to KSM kernel code, or remote code. The latter deals with requests to and from other systems. The hardware component module (HWC) resides in the kernel, and contains all the registration routines to create and maintain hardware components in the hardware set. It also contains the device nodes for device special file management, which is performed using `dsfmgr`.

The hardware set consists of data structures that describe all of the hardware components that are part of the system. A hardware component (or device) becomes part of the hardware set when registered by its driver. Devices have various attributes that describe their function and content.

Each attribute is assigned a value. You can read or manipulate these attribute values using hwmgr.

Hardware management using the hwmgr utility is organized into three parts, referred to as subsystems by the hwmgr utility. The subsystems are identified as component, scsi and name. The subsystems are related to the system hardware databases as follows:

- The component subsystem references all hardware devices specified in the (binary) /etc/dec_hwc_ldb and /etc/dec_hwc_cdb databases. This includes most devices on a system.
- The name subsystem references all hardware components in the binary /etc/dec_hw_db database, often referred to as the hardware topology. The database contains hardware persistence information, maintained by the kernel driver framework and includes data for buses, controllers and devices.
- The scsi subsystem references all SCSI devices in the binary /etc/dec_scsi_db database. The SCSI database contains entries for all devices managed by the SCSI/CAM architecture.

The specific features of hwmgr are as follows:

- It is specific to the hardware management subsystem in the kernel, and uses only the KSM hardware set and functions provided by the enhanced management subsystem in the kernel.
- It provides a wide range of hardware management features, managing many hardware databases instead of just one or two. (Previous utilities were often focused on a single database.)
- It enables you to manage hardware components that are currently unregistered in the KSM hardware set. These may be hardware components seen on a previous system boot, but not currently seen in the active configuration.
- It enables you to propagate a management request to multiple members of a cluster.

5.4.2 Understanding hwmgr Command Options

The hwmgr utility works with the KSM hardware set and the kernel hardware management module, providing you with the ability to manage hardware components. A hardware component can be a storage peripheral, such as a disk or tape, or a system component such as a CPU or a bus. Use the hwmgr utility to manage hardware components on either a single system or on a cluster.

The hwmgr utility provides two types of commands, internal and generic. Internal commands do not specify a subsystem identifier on the command line. Generic commands are characterized by a subsystem identifier after the command name.

5.4.2.1 Using Generic Hardware Manager Commands

Generic hwmgr commands have the following synopsis:

/sbin/hwmgr [component | name | scsi] [*parameter*]

Refer to the hwmgr(8) reference page and use the -help command option to obtain information on the command syntax, as shown in the following example:

```
# hwmgr -help component
```

Note that some hwmgr commands are duplicated in more than one subsystem and not all commands are usable across all subsystems. You should use the subsystem most closely associated with the type of operation you want to perform. The following are examples of commands. Refer to the hwmgr(8) reference page for a definitive list of commands and for additional examples.

- -add – Use this command to add items to certain databases. For example, a hardware persistence entry:

  ```
  # /sbin/hwmgr -add name -component_name \
  scsi -component_number 1
  ```

- -delete – Use this command to delete information from databases. For example, the following command will get the hardware component for the entry being deleted, and pass the request to the component subsystem handler to finish the deletion. Note that if the entry is not registered in the kernel with HWC (only under unusual circumstances) the -delete option will remove the entry from the CAM database without calling the component subsystem.

  ```
  # /sbin/hwmgr -delete name \
  -entry scsi1 -member witt
  ```

- -scan – Use the scan command to check databases for new device information. For example, the following command probes the scsi subsystem for new hardware:

  ```
  # /sbin/hwmgr -scan scsi
  ```

- -show – Use the show command to display information from databases, such as the hardware components from the component subsystem. For example, the following command will display all hardware components, including hardware components that were previously registered but may not be currently registered:

  ```
  # /sbin/hwmgr -show component
  ```

5.4.2.2 Using Internal Hardware Manager Commands

Internal hwmgr commands have the following typical synopsis:

/sbin/hwmgr -get attribute [saved | default | current] [-a *attribute*] [-a *attribute=value*] [-a *attribute!=value*] [-id *hardware-component-id*] [-category *hardware-category*] [-member *cluster-member-name*] [-cluster]

The -get attribute command option is only one of many command options available. Obtain a complete listing of command options using the following command:

```
# /sbin/hwmgr -help
```

Refer also to the hwmgr(8) reference page for a complete list of supported command combinations, and optional flags. Examples of commands are shown in the following list:

- -view – Use this command option to display information. For example, the following command displays the cluster status:

  ```
  # /sbin/hwmgr -view cluster
  Member ID    State    Member HostName
  ---------    -----    ---------------
  1            UP       rene(localhost)
  2            UP       witt
  3            UP       freu
  4            UP       rogr
  ```

  The output from this command provides identifiers that can be used to specify operations in other hwmgr command options. Use the HostName whenever a command option allows you to specify -member*hostname*

  Other supported options are:

  – env (environment) – Use this option to display the current values of environment variables such as HWMGR_DATA_FILE, the environment variable that you use to set the location of the main hwmgr data file.

  – transaction – Use this option to display information on the most recent hardware management transaction.

  – devices – Use this option to display all devices on the system. All devices on the local host will be returned by default, but you can specify parameters to filter the output.

  ```
  # /sbin/hwmgr -view devices
   HWID:             DSF Name  Mfg  Model            Location
   -----------------------------------------------------------------
      3:            /dev/kevm
     28:   /dev/disk/floppy0c       3.5in floppy     fdi0-unit-0
     30:      /dev/disk/dsk0c  DEC  RZ1DF-CB (C) DEC bus-0-targ-0-lun-0
     31:    /dev/disk/cdrom0c  DEC  RRD47   (C) DEC  bus-0-targ-4-lun-0
  ```

  The output provides the hardware identifier (HWID) number assigned to the device, which you use as a parameter in other hwmgr commands. The device special file (DSF) name for the device is identified and can also be used to specify devices in certain hwmgr command options. The hardware vendor's model number, is specified, as shown on the device or its casing. Finally, the physical location of the device is listed, by bus, target, and logical unit number (lun).

  – hierarchy – Use this option to display the current hardware component hierarchy in the KSM set. All devices on the local host will be returned by default, but you can specify parameters to filter the output. The output provides the hardware identifier (HWID) for each device, the device category, such as disk or bus, and the persistence name, such as isp0. This information can be used to specify operations in other hwmgr commands.

- - flash light – Use this option to flash the display light (LED) on a SCSI disk device for a default time period of 30 seconds. Note that this operation may not work on all SCSI devices, and you may have to open a cover to access the light on some systems, particularly where the disk is installed in an internal bay.

- -get attribute or -set attribute – Use the -get and -set commands to return or configure (set) attribute values for a device. You can specify the device attributes to manipulate, according to their type and one or more optional matching parameters. The type of an attribute can be identified as follows:

  – saved – This is the value of the attribute that has been configured and stored in the database using the -set saved command option. When you set a saved attribute value, you change its default value and save that value in the database. That value will be read in on all subsequent system starts (boots).

  – default – This is the usual value of an attribute that has not been assigned a saved or current. When you add a new device and boot the system, all the device attributes will have their default values.

  – current – This is a temporary value of the attribute, assigned for the current boot session only. If you set an attribute using -set current, the saved value is unchanged. When you shut down and reboot the system, the value of the attribute reverts to the saved value in the database. If you want the value to persist, you must use the -set saved command option. When using the -get command, the current values are returned by default.

  Not all attributes can always have a current or saved value. Attribute values may be assigned to a device by the system at startup, so that the saved value shows 0, but the current value may be different. You may only be able to set a few attributes for a given device and these attributes are identified as (settable) in the output from the -get attr command option.

For each attribute status (saved, default, or current) you can specify the following optional parameters for -get. If the attribute can also be set, it is noted in the definition.

– -a attribute . . . – Use this option to return values of an individual attribute, such as path_fail_limit, which is a SCSI disk attribute defining the limit for path failures.

  You must specify at least one attribute for -set operations. For -get operations, if you do not specify at least one attribute, the operation will get all attributes.

– -a attribute=value . . . – Use this option to return attributes that match the specified value. For example, to search for devices that support power management, where the saved value of power management is enabled (1), use the following command:

```
# /sbin/hwmgr -get attribute -a power_mgmt_capable=1
```

  When setting attribute values with -set, use this parameter to specify the new value as follows:

```
# /sbin/hwmgr -set attribute current \
-a user_name=disk_5_bay_4 -id 18
```

– -a attribute!=value . . . – Use this option to return attributes that do not match the specified value. For example:

```
# hwmgr -get attribute saved -a power_mgmt_capable!=1
```

  This option is supported for -get operations only.

– -id hardware-component-id – Use this option to return the attribute values for the specified hardware device identifier (HWID). For example, the following command returns the current attribute values for device 18:

```
# hwmgr -get attribute current -id 18
```

  This option is supported for -get operations only.

– -category hardware-category – Use this option to specify a hardware category, such as disk or tape on which the operation should be performed. This option is supported for -get operations only. Note that you can display all the available categories using the -get cat command option.

– -member cluster-member-name – Use this option to specify the host name of a cluster member on which the operation should be performed. This option is supported for both -get and -set operations.

– -cluster – Use this option to specify that the operation should be performed cluster-wide. If this option is not specified, only data for the local host is returned. This option is supported for both -get and -set operations.

- - get category – Use this option to return a list of all Hardware Categories available on the system, such as platform, scsi-bus, or disk. Use the hardware category to specify other hwmgr operations, such as in the following example:

  ```
  # hwmgr -view devices -cat disk
  ```

Section Section 5.4.4 contains examples of how you use hwmgr to perform administrative tasks.

5.4.3 Configuring the hwmgr Environment

The hwmgr utility has some environment settings that you can use to control the amount of information displayed. The settings of the environment can be viewed using the following command, which displays the system default settings:

```
# hwmgr -view env

  HWMGR_DATA_FILE = "/etc/hwmgr/hwmgr.dat"
  HWMGR_DEBUG = FALSE
  HWMGR_HEXINTS = FALSE
  HWMGR_NOWRAP = FALSE
  HWMGR_VERBOSE = FALSE
```

As for other environment variables, you can set the values in login script, or at the command line as shown in the following example:

```
# HWMGR_VERBOSE=TRUE
# export HWMGR_VERBOSE
```

You usually only need to define the HWMGR_HEXINTS HWMGR_NOWRAP, and the HWMGR_VERBOSE values as follows:

- If HWMGR_HEXINTS is defined as TRUE, any numerical data output from a hwmgr command is displayed in hexadecimal numbers.
- If HWMGR_NOWRAP is defined as TRUE, the output from hwmgr will be truncated at 80 characters. In some cases it can be difficult to read the output from hwmgr command options as it wraps off the screen. Setting HWMGR_NOWRAP to TRUE makes the output more legible at the console. A horizontal ellipsis marks truncated lines as follows: "..."
- If HWMGR_VERBOSE is defined as TRUE, the output from hwmgr contains more detailed information. The normal output mode of the hwmgr utility hides any errors that are not critical. To view more verbose information on the status of command completion, you can also append the -verbose switch to any of the hwmgr command options.

For example, if you do a query for a KSM attribute that does not exist for all hardware components, by default the hwmgr utility will only display the output from hardware components that support the attribute, as shown in the following example:

```
# /sbin/hwmgr -get attr -a type
  6:
   type = local
  7:
   type = local
  9:
   type = MOUSE
```

Not all hardware components on the system support the attribute type, so there are errors generated by this command which are suppressed if HWMGR_VERBOSE is not defined as TRUE. To see the errors from hardware components that do not support this attribute, use the -verbose switch with the command line as follows:

```
# hwmgr -get attr -a type -verbose
  1: Attribute "type" not defined.
  2: Attribute "type" not defined.
  4: Attribute "type" not defined.
  5: Attribute "type" not defined.
  6:
   current type = local
  7:
   current type = local
  8: Attribute "type" not defined.
  9:
   current type = MOUSE
  10: Attribute "type" not defined.
  11: Attribute "type" not defined.
            .
            .
  (long display, output truncated)
```

The -verbose switch can be used with all hwmgr commands, although it does not always produce additional output.

5.4.4 Using hwmgr to Manage Hardware

The following sections contain examples of tasks that you may need to perform using hwmgr. Some of these examples may not be useful for managing a small server with a few peripheral devices. However, when managing a large installation with many networked systems or clusters with hundreds of devices they become very useful. Using hwmgr enables you to connect to an unfamiliar system, obtain information about its device hierarchy, and then perform administrative tasks without any previous knowledge about how the system is configured and without consulting system logs or files to find devices.

5.4.4.1 Locating SCSI Hardware

On systems with many SCSI peripherals, it can often be difficult to identify a particular device and associate that device with its logical address or device special file. The `-flash light` option, which currently only works for some SCSI devices, enables you to identify a device. This option has the following syntax:

/sbin/hwmgr -flash light [-dsf *device-special-file*] [-bus *N*] [-target *N*] [-lun *N*] [-seconds *number*] [-nopause]

You might use this command when you are trying to physically locate a SCSI disk. For example, a service engineer has arrived and asks where the system root disk is located. You know from your `/etc/fstab` file that you are using `/dev/disk/dsk4a` as your root device, but you do not know where that disk is physically located. The following command will flash the LED (light emitting diode) light on the root device for a minute:

```
# /sbin/hwmgr -flash light -dsf dsk4 -seconds 60
```

You can then check the disk bays for the device that is flashing its light.

The LED on the device may be the same LED that is used to indicate normal disk I/O activity (reads and writes). If there is much activity on all the disks, it may not be easy to see which disk is flashing. In this case, you can specify the `-nopause` option. Using `-nopause` will cause the target disk to turn on the LED constantly for the determined time (the default is 30 seconds). This option is also very useful on SCSI RAID devices where you have more than one disk contained in a RAIDSET and you want all of the disks to turn on their LEDs.

See also the `-locate component` option.

5.4.4.2 Viewing the System Hierarchy

The `-view` command can be used to view the hierarchy of hardware within a system. Use this command to find what adapters are controlling devices, and discover where adapters are installed on buses. The command syntax is as follows:

/sbin/hwmgr -view hierarchy [-id *hardware_component_id*] [-instance *instance_number*]

The following example shows typical output on a small system that is not part of a cluster:

```
# hwmgr -view hier
```

```
HWID: Hardware component hierarchy
  ----------------------------------
   147:  platform DEC 3000 - M400
     2:    cpu CPU0
   148:    bus tc0
   149:      connection tc0slot7
     6:        serial_port tty00
     7:        serial_port tty01
   150:        keyboard LK401
   151:        pointer VSXXXAA
   154:        network ln0
   153:      connection tc0slot6
   152:        scsi_adapter tcds0
   155:      connection tc0slot0
   156:        graphics_controller fb0
```

Note that some devices may appear as multiple entries in the hierarchy. For example, if a disk is on a SCSI bus that is shared between two adapters, the hierarchy will show two entries for the same device.

You can obtain similar views of the system hardware hierarchy using the SysMan Station.

5.4.4.3 Viewing System Categories

To perform hardware management options on all devices of the same category, or to select a particular device in a category, you may need to know what categories of devices are available. The hardware manager `-get category` command fetches all the possible values for hardware categories, and has the following syntax:

/sbin/hwmgr -get category

This command is useful when used in conjunction with the `-get/-set attributes` options, which you can use to display and configure the attributes (or properties) of a particular device. Once you know the hardware categories you can limit your attribute queries to a specific type of hardware.

The command produces output similar to the following:

```
Hardware Categories
-------------------
category = undefined
category = platform
category = cpu
category = pseudo
category = bus
category = connection
category = serial_port
category = keyboard
category = pointer
```

```
  category = scsi_adapter
  category = scsi_bus
  category = network
  category = graphics_controller
  category = disk
  category = tape
```

Your attribute query can then be focused as follows:

```
# hwmgr -get attr -category platform
  1:
   name = DEC 3000 - M400
   category = platform
```

This output informs you that the system platform has a hardware ID of 1, and that the platform name is DEC 3000 - M400. See also the `-get attribute` and `-set attribute` options.

5.4.4.4 Obtaining Component Attributes

Any device driver that controls a hardware device will register and maintain the KSM (kernel set manager) attributes for that component. Attributes are characteristics of the device that may simply be information, such as the model number of the device, or they may control some aspect of the behavior of the device, such as the speed at which it operates.

The `-get attribute` command fetches and displays KSM (kernel set manager) attributes for a component. The hardware manager utility is specific to managing hardware and only fetches KSM attributes only from the hardware set. All hardware components are identified by KSM using a unique hardware identifier, otherwise known as the hardware ID or HWID. The syntax of the command was given as an example in Section 5.4.2.2 to show typical `hwmgr` internal command syntax.

The following command will fetch all attributes for all hardware components on the local system and direct the output to a file which you can then search for information:

```
# hwmgr -get attr > sysattr.txt
```

However, if you know which device category you want to query, as was demonstrated in Section 5.4.4.3, you can focus your query on that particular category.

Querying a hardware component category for its attributes can provide useful information. For example, you may not be sure if the network is working for some reason. You may not even know what type of network adapters are installed in a system or how they are configured. Use the `-get attribute` option to determine the status of network adapters as shown in the following example:

```
# hwmgr -get attr -category network
  203:
   name = ln0
   category = network
   sub_category = Ethernet
   model = DE422
   hardware_rev =
   firmware_rev =
   MAC_address = 08-00-2B-3E-08-09
   MTU_size = 1500
   media_speed = 10
   media_selection = Selected by Jumpers/Switches
   media_type =
   loopback_mode = 0
   promiscuous_mode = 0
   full_duplex = 0
   multicast_address_list = CF-00-00-00-00-00 \
    01-00-5E-00-00-01
   interface_number = 1
```

This output provides you with the following information:

- The number 203 is the hardware ID (HWID) for this ethernet adapter.
- The fields and values listed below the HWID are the attribute names and their current values. Some values may be blank if they are not initialized by the driver. Using this information, you know that the system has a model DE422 ethernet adapter that has a device name of `ln0`.
- You can then check the status of this network adapter using the `ifconfig` command, as follows:

  ```
  # ifconfig ln0
  ln0: flags=c62 inet XX.XXX.XXX.XX netmask ffffff00 \
  broadcast XX.XXX.XX.XXX ipmtu 1500
  ```

In some cases, you can change the value of a device attribute to modify device information or change its behavior on the system. Setting attributes is described in Section 5.4.4.5. To find which attributes are settable, you can use the `-get` option to fetch all attributes and use the `grep` command to search for the for the `(settable)` keyword as follows:

```
# hwmgr -get attr | grep settable

   device_starvation_time = 25 (settable)
   device_starvation_time = 0 (settable)
   device_starvation_time = 25 (settable)
   device_starvation_time = 25 (settable)
   device_starvation_time = 25 (settable)
   device_starvation_time = 25 (settable)
```

The output shows that there is one settable attribute on the system, `device_starvation_time`. Having found this, you can now obtain a list of devices that support this attribute as follows:

```
# hwmgr -get attr -a device_starvation_time
  23:
   device_starvation_time = 25 (settable)
  24:
   device_starvation_time = 0 (settable)
  25:
   device_starvation_time = 25 (settable)
  31:
   device_starvation_time = 25 (settable)
  34:
   device_starvation_time = 25 (settable)
  35:
   device_starvation_time = 25 (settable)
```

The output from this command displays the HWID of the devices which support the `device_starvation_time` attribute. Reading the HWID in the hierarchy output, it can be further determined that this attribute is supported by SCSI disks.

See also the `-set attribute` and `-get category` options.

5.4.4.5 Setting Component Attributes

The `-set attribute` command option allows you to set (or configure) the value of settable KSM attributes (within the hardware set). Not all device attributes can be set. When you use the `-get attribute` command option, the output will flag any attributes that can be configured by labeling them as `(settable)` next to the attribute value. Finding such attributes is described in Section 5.4.4.4.

The command syntax for setting attribute values is as follows:

/sbin/hwmgr -set attribute [saved | current] {-a *attribute*} {-a attribute=...}... [-id *hwid*] [-member *cluster-member-name*] [-cluster]

As demonstrated in Section 5.4.4.4, the value of `device_starvation_time` is an example of a settable attribute supported by SCSI disks. This attribute controls the amount of time that must elapse before the disk driver determines that a device is unreachable due to SCSI bus starvation (no data transmitted). If the `device_starvation_time` expires before the driver is able to determine that the device is still there, the driver will post an error event to the binary error log.

Using the following commands, you can change the value of the `device_starvation_time` attribute for the device with the HWID of 24, and then verify the new value:

```
# hwmgr -set attr -id 24 -a device_starvation_time=60
# hwmgr -get attr -id 24 -a device_starvation_time
  24:
   device_starvation_time = 60 (settable)
```

This action does not change the saved value for this attribute. All attributes have three possible values, a current value, a saved value and a default value. The default value is a constant and can never be set. If you never set a value of an attribute, the default value applies. The saved value can be set and persists across boots. You can think of it as a permanent override of the default.

The current value can be set but does not persist across reboots. You can think of it as a temporary value for the attribute. When a system is rebooted, the value of the attribute will revert to the saved value (if there is a saved value). If there is no saved value the attribute value will revert to the default value. Setting an attribute value always changes the current value of the attribute. The following examples show how you get and set the saved value of an attribute:

```
# hwmgr -get attr saved -id 24 -a device_starvation_time
  24:
   saved device_starvation_time = 0 (settable)

# hwmgr -set attr saved -id 24 -a device_starvation_time=60
    saved device_starvation_time = 60 (settable)
# hwmgr -get attr saved -id 24 -a device_starvation_time
  24:
   saved device_starvation_time = 60 (settable)
```

See also the -get attribute and -get category command options.

5.4.4.6 Viewing the Cluster

If you are working on a cluster, you often need to focus hardware management commands at a particular host on the cluster. The -view cluster command option enables you to obtain details of the hosts in a cluster. The following sample output shows a typical cluster:

```
  Member ID      State   Member HostName
  ---------      -----   ---------------
    1            UP      ernie.zok.paq.com (localhost)
    2            UP      bert.zok.paq.com
    3            DOWN    bigbird.zok.paq.com
```

This option can also be used to verify that the hwmgr utility is aware of all cluster members and their current status. The command has the following syntax:

/sbin/hwmgr -view cluster

The preceding example indicates a three member cluster with one member (bigbird) currently down. The (localhost) marker tells us that hwmgr is currently running on cluster member ernie. Any hwmgr commands that you enter using the -cluster option will be sent to members bert and ernie, but not to bigbird as that system is unavailable. Additionally, any hwmgr commands that you issue using the -member bigbird option will fail because the cluster member state for that host is DOWN.

Note that this command only works if the system is a member of a cluster. If you attempt to run it on a single system an error message is displayed. See also the clu_get_info command, and refer to the TruCluster documentation for more information on clustered systems.

5.4.4.7 Viewing Devices

You can use hwmgr to display all devices that have a device special file name, such as /dev/disk/dsk34 using the -view devices option. The hardware manager considers any hardware component that has the KSM attribute dev_base_name to be an accessible device. (See Section 5.4.4.4 for information on obtaining the attributes of a device.) This command has the following syntax:

/sbin/hwmgr -view devices [-category *hardware_category*] [-member *cluster-member-name*] [-cluster]

This command option enables you to determine what devices are currently registered with hardware management on a system, provides information that enables you to access these devices through their device special file. For example, if you load a CD-ROM into a reader, this output could be used to determine that the CD-ROM reader should be mounted as /dev/disk/cdrom0. The -view devices option is also useful to find the hardware identifiers (HWID) for any registered devices. When you know the HWID for a device, you can use other hwmgr command options to query KSM attributes on the device, or perform other operations on the device.

Typical output from this command is shown in the following example:

```
# hwmgr -view dev


 HWID:            DSF Name Mfg      Model       Location

--------------------------------------------------------------------
    3:           /dev/kevm
   22:      /dev/disk/dsk0c DEC      RZ26        bus-0-targ-3-lun-0
   23:    /dev/disk/cdrom0c DEC      RRD42       bus-0-targ-4-lun-0
   24:      /dev/disk/dsk1c DEC      RZ26L       bus-1-targ-2-lun-0
   25:      /dev/disk/dsk2c DEC      RZ26L       bus-1-targ-4-lun-0
   29:     /dev/ntape/tape0 DEC      TLZ06       bus-1-targ-6-lun-0
   35:     /dev/disk/dsk8c  COMPAQ   RZ1CF-CF    bus-2-targ-12-lun-0
```

The listing of devices shows all hardware components that have the dev_base_name attribute on the local system. The hardware manager attempts to resolve the dev_base_name to the full path location to the device special file, such as /dev/ntape/tape0. It always uses the path to the device special file with partition c because that partition is usually used to represent the entire capacity of the device, except in the case of tapes. See Section 5.5 for information on device special file names and functions.

If you are working on a cluster, you can view all devices registered with hardware management across the entire cluster with the -cluster option, as follows:

```
# hwmgr -view devices -cluster


  HWID:            DSF Name    Model       Location        Hostname

-------------------------------------------------------------------
   20:   /dev/disk/floppy0c    3.5in          fdi0-unit-0   tril7e
   34:    /dev/disk/cdrom0c    RRD46   bus-0-targ-5-lun-0   tril7e
   35:      /dev/disk/dsk0c    HSG80   bus-4-targ-1-lun-1   tril7d
   35:      /dev/disk/dsk0c    HSG80   bus-6-targ-1-lun-1   tril7e
   36:      /dev/disk/dsk1c    RZ26N   bus-1-targ-0-lun-0   tril7e
   37:      /dev/disk/dsk2c    RZ26N   bus-1-targ-1-lun-0   tril7e
   38:      /dev/disk/dsk3c    RZ26N   bus-1-targ-2-lun-0   tril7e
   39:      /dev/disk/dsk4c    RZ26N   bus-1-targ-3-lun-0   tril7e
   40:      /dev/disk/dsk5c    RZ26N   bus-1-targ-4-lun-0   tril7e
   41:      /dev/disk/dsk6c    RZ26N   bus-1-targ-5-lun-0   tril7e
   42:      /dev/disk/dsk7c    RZ26N   bus-1-targ-6-lun-0   tril7e
   43:      /dev/disk/dsk8c    HSZ40   bus-3-targ-2-lun-0   tril7d
   43:      /dev/disk/dsk8c    HSZ40   bus-3-targ-2-lun-0   tril7e
   44:      /dev/disk/dsk9c    HSZ40   bus-3-targ-2-lun-1   tril7d
   44:      /dev/disk/dsk9c    HSZ40   bus-3-targ-2-lun-1   tril7e
   45:     /dev/disk/dsk10c    HSZ40   bus-3-targ-2-lun-2   tril7d
   45:     /dev/disk/dsk10c    HSZ40   bus-3-targ-2-lun-2   tril7e
```

Note that some devices, such as the disk with the HWID of 45:, appear more than once in this display. These are devices that are on a shared bus between two cluster members. The hardware manager displays the device entry as seen from each cluster member.

See also the following hwmgr command options: -show scsi, -show components, and -get attributes.

5.4.4.8 Viewing Transactions

Hardware management operations are transactions that need to be synchronized across a cluster. The -view transaction command option displays the state of any hardware management transactions that have occurred since the system was booted. This option can be used to check for failed hardware management transactions. The command option has the following syntax:

/sbin/hwmgr -view transactions

If you do not specify the `-cluster` or `-member` option, the command displays status on transactions that have been processed or initiated by the local host (the system on which the command is entered). Note that the `-view transaction` command is primarily for debugging problems with hardware management in a cluster, and you will not need to use this command very often, if ever. The command has the following typical output:

```
# hwmgr -view trans

   hardware management transaction status
  -----------------------------------------------------
  there is no active transaction on this system
   the last transaction initiated from this system was:
    transaction = modify cluster database
    proposal    = 3834
    sequence    = 0
    status      = 0
   the last transaction processed by this system was:
    transaction = modify cluster database
    proposal    = 3834
    sequence    = 0
    status      = 0

 proposal                      last status  success  fail
 ----------------------------  -----------  -------  ----
             Modify CDB/ 3838  0            3        0
               Read CDB/ 3834  0            3        0
           No operation/ 3835  0            1        0
            Change name/ 3836  0            0        0
            Change name/ 3837  0            0        0
              Locate HW/ 3832  0            0        0
                Scan HW/ 3801  0            0        0
   Unconfig HW - confirm/ 3933  0            0        0
    Unconfig HW - commit/ 3934  0            0        0
     Delete HW - confirm/ 3925  0            0        0
      Delete HW - commit/ 3926  0            0        0
   Redirect HW - confirm/ 3928  0            0        0
   Redirect HW - commit1/ 3929  0            0        0
   Redirect HW - commit2/ 3930  0            0        0
          Refresh - lock/ 3937  0            0        0
```

From this output you can tell that the last transaction that occurred was a modification of the cluster database.

5.4.4.9 Deleting a SCSI Device

Under some circumstances, you may want to remove a SCSI device from a system, such as when it is logging device errors and must be replaced. Use the `-delete scsi` command option to remove a SCSI component from all hardware management databases cluster-wide. This option unregisters the component from the kernel, removes all persistent database entries for the device, and removes all device special files. When you delete a SCSI component it is no longer accessible and its device special files will be removed from the appropriate `/dev` subdirectory. Note that you cannot delete a SCSI component that is currently open, and all connections to the device (such as mounts) must be terminated.

Usually, you might delete a SCSI component if it was being removed from your system and you did not want to have any information about the device remaining on the system. You might also want to delete a SCSI component if there were software, rather than hardware problems. For example, if the device was operating properly but could not be accessed through the device special file for some reason. In this case you could delete the component and use the `-scan scsi` command option to find and register it as if it were a newly installed device.

To replace the SCSI device (or bring the old device back) you can use the `-scan scsi` command option to find the device again. However, when you delete a component and then perform a `-scan` operation to bring the component back on line, it may not be assigned the device special file name that it previously held. To replace a device as an exact replica of the original, you need to perform the additional operations described in Section 5.4.4.11. In addition, there is also no guarantee that the subsequent `-scan` operation will find the device if it is not actively responding during the bus scan.

The `-delete scsi` command option has the following syntax:

/sbin/hwmgr -delete scsi [-did *scsi-device-identifier*]

Note that the SCSI device identifier `-did` is not equivalent to the hardware identifier (HWID).

The following examples show how you check the SCSI database and then delete a SCSI device:

```
# hwmgr -show scsi


      SCSI            DEVICE DEVICE   DRIVER NUM   DEVICE FIRST
HWID: DEVICEID HOST-  TYPE   SUBTYPE  OWNER  PATH  FILE   VALID
               NAME                                        PATH
  ----------------- ----------------------------------------
23:   0       bert   disk   none     2       1   dsk0    [0/3/0]
24:   1       bert   cdrom  none     0       1   cdrom0  [0/4/0]
25:   2       bert   disk   none     0       1   dsk1    [1/2/0]
30:   4       bert   tape   none     0       1   tape2   [1/6/0]
31:   3       bert   disk   none     0       1   dsk4    [1/4/0]
34:   5       bert   disk   none     0       1   dsk7    [2/5/0]
35:   6       bert   disk   none     0       1   dsk8
```

In this example, component ID 23 is currently open by a driver. You can see this because the DRIVER OWNER field is not zero, Any number other than zero in the DRIVER OWNER field means that a driver has opened the device for use. Therefore, you cannot delete SCSI component 23 because it is currently being used.

However, component ID 35 is not open by a driver, and it currently has no valid paths shown in the FIRST VALID PATH field. This means that the device is not currently accessible and can be safely deleted. The /dev/disk/dsk8* and /dev/rdisk/dsk8* device special files will also be deleted.

To delete the SCSI device, specify the SCSI DEVICEID value with the -delete option, and then review the SCSI database as follows:

```
# hwmgr -del scsi -did 6
   hwmgr: The delete operation was successful.
# hwmgr -show scsi

      SCSI            DEVICE  DEVICE  DRIVER NUM  DEVICE FIRST
HWID: DEVICE HOSTNAME TYPE    SUBTYPE OWNER  PATH FILE   VALID
      ID                                                  PATH
  -------------------------------------------------------------
23:   0      bert     disk    none    2      1   dsk0    [0/3/0]
24:   1      bert     cdrom   none    0      1   cdrom0  [0/4/0]
25:   2      bert     disk    none    0      1   dsk1    [1/2/0]
30:   4      bert     tape    none    0      1   tape2   [1/6/0]
31:   3      bert     disk    none    0      1   dsk4    [1/4/0]
34:   5      bert     disk    none    0      1   dsk7    [2/5/0]
```

The device /dev/disk/dsk8 has been successfully deleted.

5.4.4.10 Creating a User-Defined SCSI Device Name

Most devices have an identification attribute that is a unique to the device. This can be read as the serial_number or name attribute of a SCSI device. For example, the following hwmgr command will return both these attributes for the device HWID: 30, a SCSI disk:

```
# hwmgr -get attributes -id 30 -a serial_number -a name
30:
  serial_number = SCSI-WWID:0c000008:0060-9487-2a12-4ed2
  name = SCSI-WWID:0c000008:0060-9487-2a12-4ed2
```

This string is known as a world-wide identifier (WWID) because it is unique for every device on the system.

Some older devices do not provide a unique identifier, so the operating system will create such a number for the device using valid path bus/target/lun data that describes the physical location of the device. Because a device can be shared by more than one system (or more than one bus) each system that has access to the device will see a different path and will create its own unique WWID for that device. This creates the possibility of concurrent access to a device, and data on the device could be corrupted. To check for such devices, use the following command:

```
# hwmgr -show comp -cshared

 HWID:  HOSTNAME   FLAGS SERVICE COMPONENT NAME
-----------------------------------------------
   40:  joey       -cd-- iomap   SCSI-WWID:04100026:"DEC \
 RZ28M    (C) DEC00S846590H7CCX"
   41:  joey       -cd-- iomap   SCSI-WWID:04100026:"DEC \
 RZ28L-AS (C) DECJEE019480P2VSN"
   42:  joey       -cd-- iomap   SCSI-WWID:0410003a:"DEC \
 RZ28     (C) DECPCB=ZG34142470  ; HDA=000034579643"
   44:  joey       rcd-- iomap   SCSI-WWID:04100026:"DEC \
 RZ28M    (C) DEC00S735340H6VSR"
.
.
.
```

You can use `hwmgr` to create a user-defined unique name that will in turn enable you to create a WWID that is common to all systems that are sharing the device. This means that the device will have a common WWID and one set of device special file names.

The process for creating a user-defined name is as follows:

- Choose the name that you want to assign. This name should be unique within the scope of all systems that have access to the device. Although it need not be as long and complex as the WWIDs shown in the preceding example, it should be sufficiently long to provide the information that you need to recognize the renamed device and differentiate it from others.
- Decide what devices will use this name. When renamed, the device will be seen as the same device on all systems. You must update the systems so that the device can be seen.
- Each system that shares the device will create a new WWID using the string and use this new WWID for all subsequent registrations with the system. Internally, the device will still be tracked by its default WWID (if one existed). However, all external representations will display the new WWID based on the user defined name. On a cluster you must run the `-edit scsi` command option on every cluster member that has access to the device.

Caution

All systems with access to the device should be updated. Otherwise, the access controls which ensure data coherency may not be valid and data may be corrupted.

The `-edit scsi` command option has the following syntax:

/sbin/hwmgr -edit scsi [-did *device-id*] [-uwwid *user-defined-name*] [-member *cluster-member-name*]

The following examples shows how you assign a user-defined name:

```
# hwmgr -show scsi

      SCSI           DEVICE DEVICE   DRIVER NUM   DEVICE FIRST
HWID: DEVICEID HOST  TYPE   SUBTYPE OWNER  PATH FILE   VALID
      ID       NAME                                    PATH
 ------------------------------------------------------------
  22: 0       ftwod  disk   none     0      1   dsk0   [0/3/0]
  23: 1       ftwod  cdrom  none     0      1   cdrom0 [0/4/0]
  24: 2       ftwod  disk   none     0      1   dsk1   [1/2/0]
  25: 3       ftwod  disk   none     2      1   dsk2   [2/4/0]
```

This command displays which SCSI devices are on the system. On this system the administrator knows that there is a shared bus and that hardware components 24 and 25 are actually the same device. The WWID constructed for this device is constructed using the bus/target/lun address information. Because the bus/target/lun addresses are different, the device is seen as two separate devices. This can cause data corruption problems because two sets of device special files can be used to access the disk (`dev/disk/dsk1` and `/dev/disk/dsk2`).

The following command shows how you can rename the device, and demonstrates how it appears after being renamed:

```
# hwmgr -edit scsi -did 2 -uwwid "this is a test"
    hwmgr: Operation completed successfully.

# hwmgr -show scsi -did 2 -full

        SCSI              DEVICE    DEVICE  DRIVER NUM  DEVICE FIRST
  HWID:  DEVICEID HOSTNAME   TYPE      SUBTYPE OWNER  PATH FILE   VALID PATH
  -------------------------------------------------------------------------
    24:  2        ftwod      disk      none    0      1    dsk1   [1/2/0]

      WWID:0910003c:"DEC    (C) DECZG41400123ZG41800340:d01t00002l00000"
      WWID:ff10000e:"this is a test"

      BUS   TARGET  LUN   PATH STATE
      ------------------------------
      1     2       0     valid
```

The operation is repeated on the other device path and the same name is given to the device at address 2/4/0. When this is done hardware management will use the user defined name to track the device and recognize it as an alternate path to the same device:

```
# hwmgr -edit scsi -did 3 -uwwid "this is a test"
    hwmgr: Operation completed successfully.

# hwmgr -show scsi -did 3 -full

        SCSI               DEVICE    DEVICE  DRIVER NUM  DEVICE FIRST
  HWID:  DEVICEID HOSTNAME   TYPE      SUBTYPE OWNER  PATH FILE   VALID PATH
  ------------------------------------------------------------------------
    25:  3        ftwod     disk      none    0      1    dsk1   [2/4/0]

      WWID:0910003c:"DEC    (C) DECZG41400123ZG41800340:d02t00004l00000"
      WWID:ff10000e:"this is a test"

      BUS   TARGET  LUN   PATH STATE
      ------------------------------
      2     4       0     valid
```

Both of these devices now use device special file name /dev/disk/dsk1 and there is no longer a danger of data corruption as a result of two sets of device special files accessing the same disk.

5.4.4.11 Replacing a Failed SCSI Device

When a SCSI device fails, you may want to replace it in such a way that the replacement disk takes on hardware characteristics of the failed device, such as ownership of the same device special files. The -redirect command option enables you to assign such characteristics. For example, if you have an HSZ (RAID) cabinet and a disk fails, you can hot-swap the failed disk and then use the -redirect command option to bring the new disk on line as a replacement for the failed disk.

Note

The replacement device must be of the same device type for the -redirect operation to work.

This command has the following syntax:

/sbin/hwmgr -redirect scsi [-src *scsi-device-id*] [-dest *scsi-device-id*]

The following examples show how you use the -redirect option:

```
# /sbin/hwmgr -show scsi
      SCSI          DEVICE DEVICE DRIVER NUM  DEVICE   FIRST
HWID: DEVICE- HOST- TYPE   SUB-   OWNER  PATH FILE     VALID
      ID      NAME         TYPE                        PATH
  ---------------------------------------------------------
 23:   0      fwod  disk   none   2      1    dsk0    [0/3/0]
```

```
 24:   1     fwod  cdrom  none   0      1    cdrom0 [0/4/0]
 25:   2     fwod  disk   none   0      1    dsk1   [1/2/0]
 30:   4     fwod  tape   none   0      1    tape2  [1/6/0]
 31:   3     fwod  disk   none   0      1    dsk4
 37:   5     fwod  disk   none   0      1    dsk10  [2/5/0]
```

This output shows a failed SCSI disk of HWID 31. The device has no valid paths. To replace this failed disk with a new disk that has device special file name /dev/disk/dsk4, and the same dev_t information, use the following procedure:

1. Install the device as described in the hardware manual.
2. Use the following command to find the new device:

   ```
   # /sbin/hwmgr -scan scsi
   ```

   This command probes the SCSI subsystem for new devices and registers those devices. You can then repeat the -show scsi command and obtain the SCSI device id of the replacement device.
3. Use the following command to reassign the device characteristics from the failed disk to the replacement disk. This example assumes that the SCSI device id (did) assigned to the new disk is 36:

   ```
   # /sbin/hwmgr -redirect scsi -src 3 -dest 36
   ```

5.4.4.12 Viewing the name Persistence Database

The name persistence database stores information about the hardware topology of the system. This data is maintained by the kernel and includes data for controllers and buses in addition to devices. Use the -show -name command option to display persistence data which you can then manipulate using other hwmgr commands. The command has the following syntax:

/sbin/hwmgr -show name [-member *cluster-member-name*]

The following example shows typical output from the -show -name command option on a small system:

```
# hwmgr -show name -member ychain

 HWID:  NAME    HOSTNAME   PERSIST TYPE    PERSIST AT
-----------------------------------------------------
   13:  isp0    ychain     BUS             pci0 slot 5
    4:  pci0    ychain     BUS             nexus
   14:  scsi0   ychain     CONTROLLER      isp0 slot 0
   29:  tu0     ychain     CONTROLLER      pci0 slot 11
```

The following information is provided by the output:

- HWID: – The unique hardware identifier for this device. This can also be determined by the -view hierarchy command option.

- NAME – The device name and the instance number such as pci0 for personal computer interconnect (PCI) bus 0. Each additional PCI bus will have a different instance number.
- HOSTNAME – The host on which the command was run. When working in a cluster you can specify the cluster name on which the command is to operate.
- PERSIST TYPE – The type of hardware component, which can be a bus, controller, or device.
- PERSIST AT – The logical address of the device, which may map to a physical location in the hardware. For example, the SCSI controller scsi0 persists at slot 0 of the bus isp0.

5.4.4.13 Deleting and Removing a Name from the Persistence Database

One of the options for manipulating the name subsystem is to remove devices from the persistence database. The hwmgr utility offers two methods of removal:

- -remove – Use this option to take an entry out of the persistence database. This will not affect the running system, but at the next reboot, the device will no longer be seen.
- -delete – Use this option to take an entry out of the persistence database and delete it from the running system. This command will unregister and unconfigure the device, removing it from all hardware management databases.

These commands have the following syntax:

/sbin/hwmgr -remove name [-entry *name*]

/sbin/hwmgr -delete name [-entry *name*]

Where *name* is the device name shown in the output from the -show name command option described in Section 5.4.4.12

The following example shows typical output from the -show name command option on a small system:

```
# hwmgr -show name
 HWID:  NAME    HOSTNAME  PERSIST TYPE    PERSIST AT

------------------------------------------------------
   33:  aha0    fegin     BUS             eisa0 slot 7
   31:  ln0     fegin     CONTROLLER      eisa0 slot 5
    8:  pci0    fegin     BUS             ibus0 slot 0
   34:  scsi1   fegin     CONTROLLER      aha0 slot 0
   17:  scsi0   fegin     CONTROLLER      psiop0 slot 0
   15:  tu0     fegin     CONTROLLER      pci0 slot 0
```

Note that there are two scsi adapters shown. If scsi0 is the target of a -remove operation then scsi1 would not become scsi0. The information of where the adapter is located persists at aha0 slot 0 and the name scsi1 is saved across boots.

To remove scsi0 and rename scsi1 you would use the following commands:

```
# hwmgr -remove  name -ent scsi0
# hwmgr -edit name -ent scsi1 -parent_num 0
```

5.5 Device Naming and Device Special Files

Devices are made available to the rest of the system through device special files located in the /dev directory. A device special file enables an application (such as a database application) to access a device through its device driver, which is a kernel module that controls one or more hardware components of a particular type. For example, network controllers , graphics controllers, and disk devices (including CD-ROM devices). See Section 5.4 for a discussion of system components.

Device special files are also used to access pseudodevice drivers that do not control a hardware component, for example, a pseudoterminal (pty) terminal driver, which simulates a terminal device. The pty terminal driver is a character driver typically used for remote logins; it is described in Section 5.6. (For detailed information on device drivers refer to the device driver documentation.)

Normally, device special file management is performed automatically by the system. For example, when you install a new version of the UNIX operating system, there is a point at which the system probes all buses and controllers and all the system devices are found. The system then builds databases that describe the devices and creates device special files which make them available to users. The most common way that you use a device special file is to specify it as the location of a UFS file system in the system /etc/fstab file, which is documented in Chapter 6.

You only need to perform manual operations on device special files when there are problems with the system or when you need to support a device that cannot be handled automatically. The following sections describes the way that devices and device special files are named and organized in Version 5.0 or higher. See Appendix B for information on other supported device mnemonics for legacy devices and their associated device names.

Note the following:

- A current device special file for a SCSI device has the format /dev/disk/disk13a for SCSI disks and /dev/ntape/tape0_d0 for SCSI tapes. A SCSI device special file in the format /dev/rz10b is a legacy device special file. The following sections differentiate between **current** and **legacy** device special files. You may also see these referred to as old (legacy) and new (current) device names in some scripts and utilities. First time users of the operating system need not be concerned with legacy device special file names except where there is a need to use third-party drivers and devices that do not support the current naming model. (The structure of a device special file will be described in detail later in this section.)

- There is currently one device special file naming model for SCSI disk and tape devices and a different model for all other devices. The naming system for SCSI disk and tape devices will be extended to the other devices in future releases. This ensures that there is continued support for legacy devices and device names on a nonclustered system. Applications and utilities will support all device names or will display an error message informing you of which device name format is supported.

Legacy device names and device special files will be maintained for some time and their retirement schedule will be announced in a future release.

5.5.1 Related Documentation and Utilities

The following documents contain information about device names:

- Books:
  - Chapter 6 contains information about context dependent symbolic links (CDSLs). Some directories that contain device special files are CDSLs; you should be familiar with this concept before you read this section.
- Reference pages and utilities:
  - The `dsfmgr`(8) reference page describes the utility used to manage device special files. The `MAKEDEV`(8) reference page describes the utility used to manage legacy device special files, if you need to create `rz*` format device special files.
  - The `disklabel`(8) reference page describes the utility used to maintain disk pack labels.
  - The `diskconfig`(8) reference page describes how to invoke the Disk Configuration interface, a graphical disk management tool that provides additional features over `disklabel` in that you can use it to partition disks and create file systems on the disks in a single operation. You can also launch the Disk Configuration interface from the CDE Application Manager - System_Admin folder. The Disk icon is located in the Configuration folder. Online help describes how to use this interface.

5.5.2 Device Special File Directories

You should be familiar with the file system hierarchy described in Chapter 6, in particular the implementation of *Context Dependent Symbolic Links* (CDSLs). CDSLs enable some devices to be available cluster-wide, when a system is part of a cluster.

For device special files, a /devices directory exists under / (root). This directory contains subdirectories that each contain device special files for a class of devices. A class of device corresponds to related types of devices, such as disks or nonrewind tapes. For example, the directory /dev/disk contains files for all supported disks, and /dev/ntape contains device special files for nonrewind tape devices. Currently, only the subdirectories for certain classes have been created. The available classes are defined in Appendix B. Note that in all operations you will need to specify paths using the /dev directory and not the /devices directory.

From the /dev directory, there are symbolic links to corresponding subdirectories to the /devices directory. For example:

```
lrwxrwxrwx 1 root system 25 Nov 11 13:02 ntape ->
../../../../devices/ntape

lrwxrwxrwx 1 root system 25 Nov 11 13:02 rdisk ->
../../../../devices/rdisk

lrwxrwxrwx 1 root system 24 Nov 11 13:02 tape ->
../../../../devices/tape
```

This structure enables certain devices to be host-specific when the system is a member of a cluster. It enables other devices to be shared between all members of a cluster. In addition, new classes of devices can be added by device driver developers and component vendors.

5.5.2.1 Legacy Device Special File Names

According to legacy device naming conventions, all device special files are stored in the /dev directory. The device special file names indicate the device type, its physical location, and other device attributes. Examples of the file name format disk and tape device special file names that use the legacy conventions are /dev/rz14f for a SCSI disk and /dev/rmt0a. The name contains the following information:

```
/path/prefix{root_name}{unit_number}{suffix}
/dev/          rmt          0          a
/dev/   r       rz          4          c
/dev/   n      rmt          12         h
```

This information is interpreted as follows:

The *path* is the directory for device special files. All device special files are placed in the /dev directory.

The *prefix* differentiates one set of device special files for the same physical device from another set, as follows:

- r – Indicates a character (raw) disk device. Device special files for block devices have no prefix.
- n – Indicates a no rewind on close tape device. Device special files for rewind on close tape devices have no prefix.

The *root_name* is the two or three-character driver name, such as rz for SCSI disk devices, or rmt for tape devices.

The *unit_number* is the unit number of the device, as follows:

- For SCSI disks, the unit number is calculated with the formula:

  ```
  unit = (bus * 8) + target
  ```

  For HSZ40 and HSZ10 disk devices, a letter can precede the unit number to indicate the LUN, where a is LUN 0, b is lun 1, and so on. You do not need to include the letter a for LUN 0, it is the default.

- For tapes, the prefix is a sequential number from 0 - 7.

The *suffix* differentiates multiple device special files for the same physical device, as follows:

- Disks use the letters a through h to indicate partitions. In all, 16 files are created for each disk device: 8 for character device partitions a through h, 8 for block device partitions a through h.
- Tapes use suffixes to indicate tape densities. Up to 8 files are created for each tape device: two for each density, using the suffixes defined in Table 5–1.

Table 5–1: Tape Device Suffix for Legacy Device Special Files

Suffix	Description
a	QIC-24 density for SCSI QIC devices.
l	The lowest density supported by the device, or QIC-120 density for SCSI QIC devices.
m	Medium density when a drive is triple density, or QIC-150 density for SCSI QIC devices.
h	The highest density supported by the device, or QIC-320 density for SCSI QIC devices.

Legacy device naming conventions are supported so that scripts will continue to work as expected. However, features available with the current device naming convention may not work with the legacy naming convention. When Version 5.0 or higher is installed, none of the legacy device special files (such as rz13d) will be created during the installation. If you determine that legacy device special file naming is required, you will need to create the legacy device names using the appropriate commands described in dsfmgr(8). Note that some devices will not support legacy device special files.

5.5.2.2 Current Device Special File names

Current device special files imply abstract device names and convey no information about the device architecture or logical path to the device. The new device naming convention consists of a descriptive name for the device and an instance number. These two elements form the basename of the device as shown in Table 5–2.

Table 5–2: Sample Current Device Special File Names

Location in /dev	Device Name	Instance	Basename
/disk	dsk	0	dsk0
/rdisk	dsk	0	dsk0
/disk	cdrom	1	cdrom1
/tape	tape	0	tape0

A combination of the device name, with an system-assigned instance number creates a basename such as dsk0.

The current device special files are named according to the basename of the devices, and include a suffix that conveys more information about the device being addressed. This suffix will differ depending on the type of device, as follows:

- Disks – These device file names consist of the basename and a suffix from a through z. For example, dsk0a. Disks use a through h to identify partitions. By default, CD-ROM and floppy disk devices use only the letters a and c only. For example, cdrom1c and floppy0a.

  The same device names exist in the class directory /dev/rdisk for raw devices.

- Tapes – These device file names have the basename and a suffix comprised of the characters _d followed by an integer. For example tape0_d0. This suffix determines the density of the tape device, according to the entry for the device in the /etc/ddr.dbase file. For example:

Device	Density
tape0	Default density
tape0c	Default density with compression
tape0_d0	Density associated with entry 0 in /etc/ddr.dbase
tape0_d1	Density associated with entry 1 in /etc/ddr.dbase

Note that with the new device special file naming, there is a direct mapping from the legacy tape device name suffix to the current name suffix as follows:

Legacy Device Name Suffix	Current Suffix
l (low)	_d0
m (medium)	_d2
h (high)	_d1
a (alternate)	_d3

There are two sets of device names for tape that both conform to the current naming convention. The /dev/tape directory for rewind devices and the /dev/ntape directory (for no rewind). To determine which device special file to use, you can look in the /etc/ddr.dbase file.

5.5.2.3 Converting Device Special File Names

If you have shell scripts that use commands which act on device special files, you should note that any command or utility supplied with the operating system operates on current and legacy file names in one of the following ways:

- The program will accept both forms of device name.
- Only the current device names will be supported by the program. This means that if you use legacy device names, you will not be able to use these utilities.
- Only the old device names will be supported. This means that if you use current device names, you will not be able to use these utilities.

Note however than no device can use both forms of device names simultaneously. You should test any shell scripts, and if necessary refer to the individual reference pages or on-line help for a utility.

If you want to update scripts, translating legacy names to the equivalent current name is a simple process. Table 5–3 shows some examples of legacy device names and corresponding current device names. Note that there is no relationship between the instance numbers. A device that was associated with device special file /dev/rz10b may be associated with /dev/disk/dsk2b under the current system.

Using these names as examples, you should be able to translate most device names that appear in your scripts. You can also use the utility dsfmgr(8) to convert device names.

Table 5–3: Sample Device Name Translations

Legacy Device Special File Name	New Device Special File Name
/dev/rmt0a	/dev/tape/tape0
/dev/rmt1h	/dev/tape/tape1_d1
/dev/nrmt0a	/dev/ntape/tape0_d0
/dev/nrmt3m	/dev/ntape/tape3_d2
/dev/rz0a	/dev/disk/dsk0a
/dev/rz10g	/dev/disk/dsk10g
/dev/rrz0a	/dev/rdisk/dsk0a
/dev/rrz10b	/dev/rdisk/dsk10b

5.5.3 Managing Device Special Files

In most cases, the management of device special files is undertaken by the system itself. During the initial full installation of the operating system, the device special files are created for every SCSI disk and SCSI tape device found on the system. If the system was updated from a previous version using the update installation procedure, both the current device special files and the legacy device files will exist. However, if you subsequently add new SCSI devices dsfmgr will only create new device special files by default. When the system is rebooted, dsfmgr is called automatically during the boot sequence to create the new device special files for the device. The system also automatically creates the device special files that it requires for pseudodevices such as ptys (pseudoterminals).

When you add a SCSI disk or tape device to the system, the new device will be located automatically, added to the hardware management databases, and its device special files will be created. On the first reboot after installation of the new device, dsfmgr is called automatically during the boot sequence to create the new device special files for that device.

However, under certain circumstances, you may need to perform manual administration of device special files, such as creating legacy device special files or verifying the device databases. The utility named dsfmgr enables you to manage device special files. Some devices or some system configuration changes may require the manual creation of a device special file.

To support applications that will only work with legacy device names, you may need to manually create the legacy device special files, either for every existing device, or for only for recently-added devices. Note however that some recent devices using features such as Fibre Channel will only support the current special device file naming convention.

The following sections describe some typical uses of dsfmgr. Refer to the dsfmgr(8) reference page for detailed information on the command syntax. The system script file /sbin/dn_setup, which runs at boot time to create device special files, provides an example of a script that uses dsfmgr command options.

5.5.3.1 Using dn_setup to Perform Generic Operations

The script /sbin/dn_setup script runs automatically at system start up to create device special file names. Normally, you will not need to use dn_setup options, however they will be useful if you need to troubleshoot device name problems or restore a damaged special device file directory or database files. See also Section 5.5.3.3.

If you frequently change your system configuration or install different versions of the operating system you may see device-related error messages at the system console during system start up. These messages might indicate that the system was unable to assign device special file names. This problem can occur when the saved configuration does not map to the current configuration. Adding or removing devices between installations can also cause the problem.

The command syntax is as follows:

/sbin/dn_setup [-sanity_check] [-boot] [-default] [-clean] [-default_config] [-init]

The dn_setup script has the following functions. Generally, only the -sanity_check option is useful to administrators. The remaining options should be used under the guidance of technical support for debugging and problem solving:

-sanity_check

Verifies the consistency and currency of the device special files and the directory hierarchy. The message Passed is displayed if the check is successful.

-boot

Runs at boot time to create all the default device special databases, files, and directories.

-default

Creates only the required device special directories.

-clean

Deletes everything in the device special directory tree and re-creates the entire tree (including device special files).

-default_config

Creates only the class and category databases.

-init

Removes all the default device special databases, files, and directories and re-creates everything.

5.5.3.2 Displaying Device Classes and Categories

Any individual type of device on the system is identified in the Category to Class-Directory, Prefix Database file, /etc/dccd.dat. You can display information in these databases using dsfmgr This information enables you to find out what devices are on a system, and obtain device identification attributes that can be used with other dsfmgr command options. For example, a class of devices have related physical characteristics, such as being disk devices. Each class of devices has its own directory in /dev such as /dev/ntape for nonrewind tape devices. Device classes are stored in the Device Class Directory Default Database file, /etc/dcdd.dat.

To view the entries in these databases, you use the following command:

```
# /sbin/dsfmgr -s
```

```
dsfmgr: show all datum for system at /

Device Class Directory Default Database:
     # scope mode  name
    --  ---  ----  -----------
     1   l   0755  .
     2   c   0755  disk
     3   c   0755  rdisk
     4   c   0755  tape
     5   c   0755  ntape
     6   l   0755  none

Category to Class-Directory, Prefix Database:
 #   category       sub_category   type        directory  iw  t mode prefix
--   -------------- -------------- ----------  ---------  --  - ---- --------
 1   disk           cdrom          block       disk        1  b 0600 cdrom
 2   disk           cdrom          char        rdisk       1  c 0600 cdrom
 3   disk           floppy         block       disk        1  b 0600 floppy
 4   disk           floppy         char        rdisk       1  c 0600 floppy
 5   disk           floppy_fdi     block       disk        1  b 0666 floppy
 6   disk           floppy_fdi     char        rdisk       1  c 0666 floppy
 7   disk           generic        block       disk        1  b 0600 dsk
 8   disk           generic        char        rdisk       1  c 0600 dsk
 9   parallel_port  printer        *           .           1  c 0666 lp
10   pseudo         kevm           *           .           0  c 0600 kevm
11   tape           *              norewind    ntape       1  c 0666 tape
```

```
12   tape           *              rewind      tape        1  c 0666 tape
13   terminal       hardwired      *           .           2  c 0666 tty
14   *              *              *           none        1  c 0000 unknown

Device Directory Tree:
   12800    2 drwxr-xr-x  6 root system 2048 May 23 09:38 /dev/.
     166    1 drwxr-xr-x  2 root system  512 Apr 25 15:58 /dev/disk
    6624    1 drwxr-xr-x  2 root system  512 Apr 25 11:37 /dev/rdisk
     180    1 drw-r--r--  2 root system  512 Apr 25 11:39 /dev/tape
    6637    1 drw-r--r--  2 root system  512 Apr 25 11:39 /dev/ntape
     181    1 drwxr-xr-x  2 root system  512 May  8 16:48 /dev/none

Dev Nodes:
 13100  0 crw-------  1 root system 79,  0 May  8 16:47 /dev/kevm
 13101  0 crw-------  1 root system 79,  2 May  8 16:47 /dev/kevm.pterm
 13102  0 crw-r--r--  1 root system 35,  0 May  8 16:47 /dev/tty00
 13103  0 crw-r--r--  1 root system 35,  1 May  8 16:47 /dev/tty01
 13104  0 crw-r--r--  1 root system 34,  0 May  8 16:47 /dev/lp0
   169  0 brw-------  1 root system 19, 17 May  8 16:47 /dev/disk/dsk0a
  6627  0 crw-------  1 root system 19, 18 May  8 16:47 /dev/rdisk/dsk0a
   170  0 brw-------  1 root system 19, 19 May  8 16:47 /dev/disk/dsk0b
  6628  0 crw-------  1 root system 19, 20 May  8 16:47 /dev/rdisk/dsk0b
   171  0 brw-------  1 root system 19, 21 May  8 16:47 /dev/disk/dsk0c
    .
    .
    .
```

This display provides you with information that can be used with other `dsfmgr` commands. (Refer to the `dsfmgr`(8) reference page for a complete description of the fields in the databases). For example:

- `class` – The device class such as `disk` (a block device), `rdisk` (a character device), or `tape`, a rewind device. This information can be used with the `dsfmgr -a` (add) or `dsfmgr -r` (remove) command options to add or remove classes.
- `category` – The primary description of a device. For example, SCSI disks, CD-ROM readers and floppy disk readers are all in the `disk` category. This information can be used with the `dsfmgr -a` (add) or `dsfmgr -r` (remove) command options to add or remove categories.

5.5.3.3 Verifying and Fixing the Databases

Under unusual circumstances, the device databases may be corrupted or device special files may accidentally be removed from the system. You may see errors indicating that a device is no longer available, but the device itself does not appear to be faulty. If you suspect that there may be a problem with the device special files, you can check the databases using the `dsfmgr -v` (verify) command option.

Caution

If you see error messages at system start up that indicate a device naming problem, you should use the verify command only to enable you to proceed with the boot. Check your system configuration before and after verifying the databases. The verification procedure will fix most errors and enable you to proceed, however it will not cure any underlying device or configuration problems.

Such problems are rare and usually arise when performing unusual operations such as switching between boot disks. Errors generally mean that the system was unable to recover and use a good copy of the previous configuration, and errors usually arise because the current system configuration no longer matches the database.

As for all potentially destructive system operations, you should always be able to restore the system to its identical previous configuration, and to restore the previous version of the operating system from your backup.

For example, if you attempted to configure the floppy disk device to use the `mtools` utilities, and you found that you could not access the device, you would use the following command:

```
# /sbin/dsfmgr -v

dsfmgr: verify all datum for system at /

Device Class Directory Default Database:
    OK.

Device Category to Class Directory Database:
    OK.

Dev directory structure:
    OK.

Dev Nodes:
    ERROR: device node does not exist: /dev/disk/floppy0a
    ERROR: device node does not exist: /dev/disk/floppy0c
  Errors:   2

Total errors:   2
```

This output shows that the device special files for the floppy disk device are missing. To correct this problem, use the same command with the `-F` (fix) flag to correct the errors as follows:

```
# /sbin/dsfmgr -v -F

dsfmgr: verify all datum for system at /

Device Class Directory Default Database:
    OK.
```

```
Device Category to Class Directory Database:
    OK.

Dev directory structure:
    OK.

Dev Nodes:
    WARNING: device node does not exist: /dev/disk/floppy0a
    WARNING: device node does not exist: /dev/disk/floppy0c
    OK.

Total warnings:   2
```

Notice that the ERROR changes to a WARNING, which indicates that the device special files for the floppy disk were created automatically. Repeating the dsfmgr -v command will then show no errors.

5.5.3.4 Deleting Device Special Files

If a device is permanently removed from the system, you may want to remove its device special file so that it can be reassigned to another type of device. Use the dsfmgr -D command option to remove device special files as shown in the following example:

```
# cd /dev/disk
# ls
cdrom0a   dsk0a     dsk0c     dsk0e     dsk0g     floppy0a
cdrom0c   dsk0b     dsk0d     dsk0f     dsk0h     floppy0c

#  /sbin/dsfmgr -D cdrom0*
 -cdrom0a -cdrom0a -cdrom0c -cdrom0c
# ls
dsk0a     dsk0c     dsk0e     dsk0g     floppy0a
dsk0b     dsk0d     dsk0f     dsk0h     floppy0c
```

Notice that the output from ls shows that there are device special files for cdrom0. Running dsfmgr -D on all cdrom devices, as shown by the wildcard symbol (*), causes all device special files for that sub_category to be permanently deleted. The message that follows repeats the basename (cdrom0) twice, because it also deletes the device special files from the /dev/rdisk directory where the raw or character device special files were located.

Note that if device special files are deleted in error, and no hardware changes are made then they can be recreated as follows:

```
#  /sbin/dsfmgr -n cdrom0a

  +cdrom0a +cdrom0a
#  /sbin/dsfmgr -n cdrom0c
```

```
+cdrom0c +cdrom0c
```

5.5.3.5 Moving and Exchanging Device Special File Names

You may want to reassign the device special files between devices using the `dsfmgr -m` (move) command option. It is also possible to exchange the device special files of one device for those of another device using the `-e` option. The syntax for this command option is as follows:

/sbin/dsfmgr [-e-m *basename_1* || *basename_2* **|** *instance*||]

Where:

- *basename_1* is the prefix and instance number of the source device such as `dsk0` or `tape7`
- *basename_2* is the prefix and instance number of the target device, such as `dsk0` or `tape7`
- *instance* is just the device name and instance number of the target device.

For example:

```
# /sbin/dsfmgr -m dsk0 dsk10
# /sbin/dsfmgr -e dsk1 15
```

5.6 Manually Configuring Devices Using ddr_config

Most device management is automatic. A device added to a system will be recognized, mapped, and added to the device databases as described in Section 5.4. However, you may sometimes need to add devices that cannot be detected and added to the system automatically. These devices may be old, or new prototypes, or they may not adhere closely to supported standards such as SCSI. In these cases, you must manually configure the device and its drivers in the kernel, using the `ddr_config` utility described in this section.

The following sections describe how to create pseudoterminals (`ptys`), a terminal pseudodevice that enables remote logins.

There are two processes you use to effect the reconfiguration and rebuilding of a kernel: a static method and a dynamic method.

- The dynamic method uses the `ddr` utility to rebuild the kernel and effect the disk configuration changes without shutting down the operating system.
- The static method uses the `MAKEDEV` and `config` utilities and requires that you shut down the operation and restart it in order to rebuild the kernel and effect the changes.

The MAKEDEV command or the mknod command is used to create device special files instead of the dsfmgr utility. The kmknod command creates device special files for third-party kernel layered products. Refer to MAKEDEV(8), mknod(8), and kmknod(8) for more information.

For loadable drivers, the sysconfig command creates the device special files by using the information specified in the driver's stanza entry in the /etc/sysconfigtab database file.

5.6.1 Dynamic Method to Reconfigure the Kernel

The following sections explain how to use the ddr_config utility to manage the DDR database for your system.These sections introduce DDR, then describe how you use the ddr_config utility to:

- Add SCSI devices to the DDR database
- Convert a customized cam_data.c file

5.6.1.1 Understanding Dynamic Device Recognition

Dynamic Device Recognition is a framework for describing the operating parameters and characteristics of SCSI devices to the SCSI CAM I/O subsystem. You can use DDR to include new and changed SCSI devices into your environment without having to reboot the operating system. You do not disrupt user services and processes, as happens with static methods of device recognition.

DDR is preferred over the static method for recognizing SCSI devices. The current, static method, as described in Chapter 4, is to edit the /sys/data/cam_data.c data file and include custom SCSI device information, reconfigure the kernel, and shut down and reboot the operating system.

Note

Support for the static method of recognizing SCSI devices will be retired in a future release.

Both methods can be employed on the same system, with the restriction that the devices described by each method are exclusive to that method (nothing is doubly-defined).

The information DDR provides about SCSI devices is needed by SCSI drivers. You can supply this information using DDR when you add new SCSI devices to the system, or you can use the /sys/data/cam_data.c data file and static configuration methods. The information provided by DDR and the cam_data.c file have the same objectives. When compared to the static method of providing SCSI device information, DDR minimizes the amount of information that is supplied by the device driver or subsystem to the operating system and maximizes the amount of information that is supplied by the device itself or by defaults specified in the DDR databases.

5.6.1.1.1 Conforming to Standards

Devices you add to the system should minimally conform to the SCSI-2 standard, as specified in *SCSI-2, Small Computer System Interface-2 (X3.131-1994)*, or other variants of the standard documented in the *Software product Description*. If your devices do not comply with the standard, or if they require exceptions from the standard, you store information about these differences in the DDR database. If the devices comply with the standard, there is usually no need to modify the database. Note however that such devices should be automatically recognized or configurable using `hwmgr`.

5.6.1.1.2 Understanding DDR Messages

Following are the most common DDR message categories and the action, if any, that you should take.

- Console messages are displayed during the boot sequence.

  Frequently, these messages indicate that the kernel cannot read the DDR database. This error occurs when the system's firmware is not at the proper revision level. Upgrade to the correct revision level of the firmware.

- Console messages warn about corrupted entries in the database. Recompile and regenerate the database.
- Runtime messages generally indicate syntax errors that are produced by the `ddr_config` compiler. The compiler runs when you use the `–c` option to the `ddr_config` utility and does not produce an output database until all syntax errors have been corrected.

Use the `–h` option to the `ddr_config` command to display help on command options.

5.6.2 Changing the DDR Database

When you make a change to the operating parameters or characteristics of a SCSI device, you must describe the changes in the `/etc/ddr.dbase` file. You must compile the changes by using the `ddr_config -c` command.

Two common reasons for changes are:

- Your device deviates from the SCSI standard or reports something different from the SCSI standard

- You want to optimize device defaults, most commonly the TagQueueDepth parameter, which specifies the maximum number of active tagged requests the device supports

You use the ddr_config -c command to compile the /etc/ddr.dbase file and produce a binary database file, /etc/ddr.db. When the kernel is notified that the file's state has changed, it loads the new /etc/ddr.dbase file. In this way, the SCSI CAM I/O subsystem is dynamically updated with the changes that you made in the /etc/ddr.dbase file and the contents of the on-disk database are synchronized with the contents of the in-memory database.

Use the following procedure to compile the /etc/ddr.dbase database:

1. Log in as root or become the superuser.
2. Enter the ddr_config -c command, for example:

   ```
   # /sbin/ddr_config -c
   ```

Note that there is no message confirming successful completion. When the prompt is displayed, the compilation is complete. If there are syntax errors, they are displayed at standard output and no output file is compiled.

5.6.3 Converting Customized cam_data.c Information

You use the following procedure to transfer customized information about your SCSI devices from the /sys/data/cam_data.c file to the /etc/ddr.dbase text database. In this example, *MACHINE* is the name of your machine's system configuration file.

1. Log on as root or become the superuser.
2. To produce a summary of the additions and modifications that you should make to your /etc/ddr.dbase file, enter the ddr_config -x command. For example:

   ```
   # /sbin/ddr_config -x MACHINE > output.file
   ```

   This command uses as input the system configuration file that you used to build your running kernel. The procedure runs in multiuser mode and requires no input after it has been started. You should redirect output to a file in order to save the summary information. Compile errors are reported to standard error and the command terminates when the error is reported. Warnings are reported to standard error and do not terminate the command.
3. Edit the characteristics that are listed on the output file into the /etc/ddr.dbase file, following the syntax requirements of that file. Instructions for editing the /etc/ddr.dbase database are found in ddr.dbase(4).

4. Enter the `ddr_config -c` command to compile the changes.

See Section 5.6.2 for more information.

You can add pseudodevices, disks, and tapes statically, without using DDR, by using the methods described in the following sections.

5.6.4 Adding Pseudoterminals and Devices Without Using DDR

System V Release 4 (SVR4) pseudoterminals (ptys) are implemented by default and are defined as follows:

/dev/pts/N

The variable *N* is a number from 0-9999.

This implementation allows for more scalability than the BSD ptys (tty[a-zA-Z][0-9a-zA-Z]). The base system commands and utilities have been modified to support both SVR4 and BSD ptys. To revert back to the original default behavior, create the BSD ptys using `MAKEDEV`. See also the `SYSV_PTY`(8), `pty`(7), and `MAKEDEV`(8) reference pages.

5.6.4.1 Adding Pseudoterminals

Pseudoterminals enable users to use the network to access a system. A pseudoterminal is a pair of character devices that emulate a hardware terminal connection to the system. Instead of hardware, however, there is a master device and a slave device. Pseudoterminals, unlike terminals, have no corresponding physical terminal port on the system. Remote login sessions, window-based software, and shells use pseudoterminals for access to a system. By default, SVR4 device special files such as `/dev/pts/`<*n*> are created. You must use `/dev/MAKEDEV` to create BSD pseudoterminals such as `/dev/ttyp/`<*n*>. Two implementations of pseudoterminals are offered: BSD STREAMS and BSD `clist`.

For some installations, the default number of `pty` devices is adequate. However, as your user community grows, and each user wants to run multiple sessions of one or more timesharing machines in your environment, the machines may run out of available `pty` lines. The following command enables you to review the current value:

```
# sysconfig -q pts
pts:
nptys = 255
```

You can dynamically change the value with the `sysconfig` command, although this change will not be preserved across reboots:

```
# sysconfig -r pts nptys=400
```

To modify the value and preserve it across reboots, use the following procedure:

1. Log in as root.
2. Add or edit the pseudodevice entry in the system configuration file `/etc/sysconfigtab`. By default, the kernel supports 255 pseudoterminals. If you add more pseudoterminals to your system, you must edit the system configuration file entry and increment the number 255 by the number of pseudoterminals you want to add. The following examples show that 400 pseudoterminals have been added.

   ```
   pts:
   nptys=400
   ```

   The pseudodevice entry for `clist`-based pseudoterminals is as follows:

   `pseudo-device pty 655`

   For more information on the configuration file and its pseudodevice keywords, refer to Chapter 4.
3. For `clist`-based pseudoterminals, you also need to rebuild and boot the new kernel. Use the information on rebuilding and booting the new kernel in Chapter 4.

   When the system is first installed, the configuration file contains a pseudodevice entry with the default number of 255 pseudoterminals. If for some reason the number is deleted and not replaced with another number, the system defaults to supporting the minimum value of 80 pseudoterminals. The maximum value is 131072.

If you want to create BSD terminals, use the `/dev/MAKEDEV` command as follows:

1. Log in as root and change to the `/dev` directory.
2. Create the device special files by using the `MAKEDEV` command, which has the following syntax:

   ./MAKEDEV pty#

   The number sign (#) represents the set of pseudoterminals (0 to 101) you want to create. The first 51 sets (0 to 50) create 16 pseudoterminals for each set. The last 51 sets (51 to 101) create 46 pseudoterminals for each set. You can use the following syntax to create a large number of pseudoterminals:

   ./MAKEDEV PTY_#

   The number sign (#) represents the set of pseudoterminals (1 to 9) you want to create. Each set creates 368 pseudoterminals, except the

PTY_3 and PTY_9 sets, which create 356 and 230 pseudoterminals, respectively. (Refer to the Software Product Description (SPD) for the maximum number of supported pseudoterminals).

Note

> By default, the installation software creates device special files for the first two sets of pseudoterminals, pty0 and pty1. The pty0 pseudoterminals have corresponding device special files named /dev/ttyp0 through /dev/ttypf. The pty1 pseudoterminals have corresponding device special files named /dev/ttyq0 through /dev/ttyqf.

If you add pseudoterminals to your system, the pty# variable must be higher than pty1 because the installation software sets pty0 and pty1. For example, to create device special files for a third set of pseudoterminals, enter:

```
# ./MAKEDEV pty2
```

The MAKEDEV command lists the device special files it has created. For example:

```
MAKEDEV: special file(s) for pty2:
ptyr0 ttyr0 ptyr1 ttyr1 ptyr2 ttyr2 ptyr3 ttyr3 ptyr4 ttyr4
ptyr5 ttyr5 ptyr6 ttyr6 ptyr7 ttyr7 ptyr8 ttyr8 ptyr9 ttyr9
ptyra ttyra ptyrb ttyrb ptyrc ttyrc ptyrd ttyrd ptyre ttyre
ptyrf ttyrf
```

3. To remove BSD ptys, use the /dev/SYSV_PTY command.
4. If you want to allow root logins on all pseudoterminals, make sure an entry for ptys is present in the /etc/securettys file. If you do not want to allow root logins on pseudoterminals, delete the entry for ptys from the /etc/securettys file. For example, to add the entries for the new tty lines and to allow root login on all pseudoterminals, enter the following lines in the /etc/securettys file:

```
/dev/tty08     # direct tty
/dev/tty09     # direct tty
/dev/tty10     # direct tty
/dev/tty11     # direct tty
ptys
```

   Refer to the securettys(4) reference page for more information.

5.6.4.2 Adding Other Devices

When you add new SCSI devices to your system, they are automatically detected and configured by the Hardware Manager hwmgr and the Device Special File Manager dsfmgr. However, you may want to manually create device names for other devices using /dev/MAKEDEV. For example, you may also need to recreate device special files that were incorrectly deleted from the system.

For new devices, you must physically connect the devices and then make the devices known to the system. There are two methods, one for static drivers and another for loadable drivers. You will need the documentation that came with your system processor and any documentation that came with the device itself. You may also require a disk containing the driver software.

Appendix D provides an outline example of adding a PCMCIA modem to a system, and shows you how to create the device special files.

Note that it is not necessary to use /dev/MAKEDEV if you simply want to create legacy rz or tz device special files in /dev such as /dev/rz5. The dsfmgr utility provides a method of creating these device names.To add a device for a loadable driver, see the device driver documentation.

To add a device for a static driver, see Section 5.6.4.1.

Next, make the device special files for the device, by following these steps:

1. Change to the /dev directory.
2. Create the device special files by using the MAKEDEV command. Use the following syntax to invoke the MAKEDEV command:

   ./MAKEDEV *device*#

   The *device* variable is the device mnemonic for the drive you are adding. Appendix B lists the device mnemonics for all supported disk and tape drives. The number sign (#) is the number of the device. For example, to create the device special files for two PCMCIA modem cards, use the following command:

   ```
   # ./MAKEDEV ace2 ace3

   MAKEDEV: special file(s) for ace2:
   tty02
   MAKEDEV: special file(s) for ace3:
   tty03
   ```

   The generated special files should look like this:

   ```
   crw-rw-rw-   1 root     system    35,  2 Oct 27 14:02 tty02
   crw-rw-rw-   1 root     system    35,  3 Oct 27 14:02 tty03
   ```

3. Stop system activity by using the shutdown command and then turn off the processor. Refer to Chapter 2 for more information.
4. Power up the machine. To ensure that all the devices are seen by the system, power up the peripherals before powering up the system box.

5. Boot the system with the new kernel. Refer to Chapter 2 for information on booting your processor.

5.7 Using Device Utilities

The preceding sections described generic hardware management tools that are used to manage many aspects of all devices, such as the `hwmgr` utility described in Section 5.4. The following sections describe hardware management tools that are targeted at a particular kind of device and perform specific task. The topics covered in these sections are:

- Finding device utilities
- Using SCSI utilities
- Disk partitioning
- Copying and Cloning disks
- Monitoring disks

5.7.1 Finding Device Utilities

Many of the device utilities are documented elsewhere in this guide or in other volumes of the documentation set. For example, utilities that enable you to configure network devices are documented in detail in the *Network Administration* guide. Table 5–4 provides references to utilities documented in the guides, including those listed in this chapter. Other utilities are documented only in reference pages. Table 5–5 provides references to utilities documented in the reference pages and also provides pointers to reference data such as the Section 7 interface reference pages.

Table 5–4: Device Utilities Documented in the Guides

Device	Task	Location
Processor	Starting or stopping	Chapter 2
	Sharing resources	Chapter 3, Class Scheduler.
	Monitoring	Chapter 3 and Chapter 12 (Environmental)
	Power Management	Chapter 3, `dxpower`.
	Testing memory	Chapter 12
	Error and Event handling	Chapter 12 and Chapter 13
SCSI buses	Managing	Section 5.7.2.1, `scsimgr`. (Note that `hwmgr` supersedes this utility)

Table 5–4: Device Utilities Documented in the Guides (cont.)

Device	Task	Location
	Configuring	Section 5.7.2.2, scu.
Disks	Partitioning and Cloning	Section 5.7.3, diskconfig
	Copying	Section 5.7.5, dd
	Monitoring usage	Section 5.7.7, df and du
	Power Management	Chapter 3
	File systems status	Chapter 6
	Testing and exercising	Chapter 12
Tapes (and Disks)	Archiving	Chapter 9
	Testing and exercising	Chapter 12
Clock	Setting	Chapter 2
Modem	Configuring	Chapter 1

Table 5–5: Device Utilities Documented in the Reference Pages

Device	Task	Location
Devices (General)	Configuring	hwmgr(8), devswmgr(8), dsfmgr(8).
	Device Special Files	kmknod(8) , mknod(8), MAKEDEV(8), dsfmgr(8).
	Interfaces	atapi_ide(7), devio(7), emx(7).
Processor	Starting/Stopping	halt(8), psradm(8), reboot(2).
	Allocating CPU resources	class_scheduling(4), processor_sets(4), runon(1).
	Monitoring	dxsysinfo(8), psrinfo(1).
SCSI buses	Managing	sys_attrs_cam(5), ddr.dbase(4) ddr_config(8).
Disks	Partitioning	diskconfig(8), disklabel(4), disklabel(8), disktab(4).
	Monitoring	dxsysinfo(8) , diskusg(8), acctdisk(8), df(1), du(1), quota(1).

Table 5–5: Device Utilities Documented in the Reference Pages (cont.)

Device	Task	Location
	Testing and Maintenance	diskx(8), zeero(8).
	Interfaces	ra(7), radisk(8), ri(7), rz(7).
	Swap Space	swapon(8).
Tapes (and Disks)	Archiving	bttape(8), dxarchiver(8), rmt(8).
	Testing and Maintenance	tapex(8).
	Interfaces	tz(7), mtio(7), tms(7).
Floppy	Tools	dxmtools(1), mtools(1).
	Testing and Maintenance	fddisk(8).
	Interfaces	fd(7).
Terminals, Ports	Interfaces	ports(7).
Modem	Configuring	chat(8).
	Interfaces	modem(7).
Keyboard, Mouse	Interfaces	dc(7), scc(7).

See Appendix A for a list of the utilities provided by SysMan.

5.7.2 SCSI and Device Driver Utilities

The following sections describe utilities that you use to manage SCSI devices and device drivers.

5.7.2.1 Using the SCSI Device Database Manager, scsimgr

The `scsimgr` utility is used to manage entries for SCSI devices in the `/etc/dec_scsi_db` database. This is a binary database that stores the logical identification assignments for SCSI devices, and preserves these identifications across system reboots. Most of the business of managing SCSI devices is managed automatically by the system. For example, you can add a new SCSI device (such as a disk) to a system and the system will detect the device on reboot, create database entries and create the device special files in `/dev`. Entries in the `/etc/dec_scsi_db` database are used to translate from a logical identifier (ID) of a device to a physical address.

This information ensures that once a device is associated with a device identifier, it retains that identifier on the next reboot.

Note

You can now use hwmgr to perform all scsimgr operations. The scsimgr utility will be retired in a future release of the operating system.

5.7.2.2 Using the SCSI Configuration Utility, scu

The SCSI/CAM Utility Program, scu, provides commands necessary for normal maintenance and diagnostics of SCSI peripheral devices and the CAM I/O subsystem. The scu program has an extensive help feature which describes utility's commands and conventions. Refer also to the scu(8) reference page for detailed information on using this command.

You can use scu to:

- Format disks
- Reassign a defective disk block
- Reserve and release a device
- Display and set device and program parameters
- Enable and disable a device

DSA Disks

For Digital Storage Architecture (DSA) disks, use the radisk program. See the radisk(8) reference page for information.

Examples of scu usage are:

```
# scu
scu> set nexus bus 0 target 0 lun 0
Device: RZ1CB-CA, Bus: 0, Target: 0, Lun: 0, Type: Direct Access
scu> show capacity

Disk Capacity Information:

                  Maximum Capacity: 8380080 (4091.836 megabytes)
                      Block Length: 512
scu> show scsi status 0
SCSI Status = 0 = SCSI_STAT_GOOD = Command successfully completed
```

5.7.2.3 Using the Device Switch Manager, devswmgr

The `devswmgr` command enables you to you manage the device switch table by displaying information about the device drivers in the table. You can also use the command to release device switch table entries. Typically, you release the entries for a driver after you have unloaded the driver and do not plan to reload it later. Releasing the entries frees them for use by other device drivers.

Examples of `devswmgr` usage for device data are:

```
# devswmgr -display
device switch database read from primary file
  device switch table has 200 entries
# devswmgr -getnum

Device switch reservation list
                          (*=entry in use)
  driver name             instance   major
-----------------------   --------   -----
                    pfm          1      71*
                    fdi          2      58*
                    xcr          2      57
                   kevm          1      56*
               cam_disk          2      55*
                    emx          1      54
                  TMSCP          2      53
                   MSCP          2      52
                    xcr          1      44
                    LSM          4      43
                    LSM          3      42
                    LSM          2      41*
                    LSM          1      40*
                    ace          1      35*
          parallel_port          1      34*
               cam_uagt          1      30
                   MSCP          1      28
                  TMSCP          1      27
                    scc          1      24
                 presto          1      22
                cluster          2      21*
                cluster          1      19*
                    fdi          1      14*
               cam_tape          1       9
               cam_disk          1       8*
                    pty          2       7
                    pty          1       6
                    tty          1       1
                console          1       0
```

5.7.3 Partitioning Disks Using diskconfig

The Disk Configuration graphical user interface (`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 the `diskconfig`(8) reference page for information on invoking the Disk Configuration utility (`diskconfig`). An online help volume describes how you use the graphical interface. See the `disklabel`(8) reference page for information on command options.

The Disk Configuration utility provides a graphical interface to several disk maintenance tasks that can also be done manually, using the following commands:

- `disklabel` – This command can be used 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 also the `/etc/disktab` file, described in the `disklabel`(4) reference page.
- `newfs` – This command creates a new UFS file system on the specified device. The `newfs` command cannot be used to create Advanced File System (AdvFS) domains. Instead, use the `mkfdmn` command, as described in the`mkfdmn`(8) reference page.
- `mkfdmn` and `mkfset` – These commands are used to create Advanced File System (AdvFS) domains and filesets.

An example of using manual methods is provided in Section 5.7.4.

The Disk Configuration interface can be invoked as follows:

- At the system prompt, type `diskconfig`.
- From the CDE Front Panel, SysMan Applications pop-up menu, choose Configuration. Then select the Disk icon from the SysMan Configuration folder.

Caution

The Disk Configuration will display 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* will be displayed. This is the main window for DiskConfig, and lists the following information for each disk:

- The disk basename, such as `dsk10`. See Section 5.5 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 Section 5.4 for information on the device location.

Select a device by double-clicking on the list item (or press configure when a disk is highlighted) . The following windows will be displayed:

Disk Configuration: Configure Partitions: *`device name device type`*

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, that 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 using the bar-chart. You can also 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 item.

Disk Configuration: Partition Table: *device name* *device type*

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 was started. If you make errors on a manual partition change, you can use this window to reset the partition table.

Refer to 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:

- Invoke the SysMan Menu, as described in Chapter 1
- Expand the Storage options, and select Basic File System Utilities - Mount File Systems
- In the Mount Operation window, select the option to mount a specific file system and press Next
- In the Name and Mount Point window:
  - Type a mount point, such as `/usr/newusers`
  - Type the partition name, such as `/dev/disk/dsk0g` or a domain name, such as `newusr_domain#usr`.

Your new file system is now accessible.

5.7.4 Manually Partitioning Disks

This section provides 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.

The disk label provides detailed information about the geometry of the disk and the partitions into which the disk is divided. You can change the label with the `disklabel` command. You must be the root user to use the `disklabel` command.

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 directly access the label on the disk instead of going through the in-memory label.

Note

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 only one 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, you must then use partition `c` to change the label. Note that partition `c` also must begin at sector 0.

Caution

If partition `a` is mounted and you attempt to edit the disk label using device partition `a`, you will not be able to change the label. Furthermore, you will not receive an error message that would indicate that the label was 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 syntax:

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 also 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*)
```

You must be careful 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 being used as swap space.
3. Examine the disk's label by using the `disklabel` command with the –r option. Refer to the `rz`(7) and `ra`(7) reference pages and to the `/etc/disktab` file 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. You can also 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:

   disklabel -e [-r] *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.

5.7.4.1 Checking for Overlapping Partitions

Commands to mount or create file systems, add a new swap device, and add disks to the Logical Storage Manager first check whether the disk partition specified in the command already contains valid data, and whether it overlaps with a partition that is already marked for use. The fstype field of the disk label is used 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 like mount, newfs, fsck, voldisk, mkfdmn, rmfdmn, and swapon also modify the disk label, so that the fstype field specifies how the partition is being used. 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 the check_usage(3) and set_usage(3) reference pages for more information.

5.7.5 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 was not meant for copying multiple files, you should copy a disk or a partition only on a disk that is used as a data disk or one 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 *AdvFS Administration*, to copy disks or partitions that contain an AdvFS fileset.

UNIX 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. Note that 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 syntax:

disklabel -r *target_device*

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 following `dd` command syntax:

dd if=*original_disk* of=*target_disk* skip=16 seek=16 bs=*block_size*

Specify 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
```

5.7.6 Cloning a System Disk

This section suggests a procedure that can be used to clone a system disk. For example, you could move your system disk from a small disk to one with larger capacity without reinstalling the operating system. Cloning involves recreating the entire file system of one disk (target) on a new disk (clone). Note that this is not presented as a definitive method, and your local system may require additional steps. The operation is best undertaken while in single-user mode.

The process assumes that you have installed the new disk as described in the hardware documentation supplied with the disk.

1. Identify the device special files for the source and target disks (`dev/disk/dskNx`). Use `dsfmgr` or `hwmgr` to identify and check disk characteristics. See Section 5.4 for information on using `hwmgr` and Section 5.5 for information on using `dsfmgr`.
2. Examine and copy the `/etc/fstab` file. This file describes the partitions and file systems you will need to clone.

3. Examine and copy the /etc/sysconfigtab file, which lists the swap partitions that you will need to re-create on the target disk. See Chapter 3 and the swapon(8) reference page.

4. Use diskconfig as described in Section 5.7.3 to label and partition a target disk to receive the clone copy. The size of partitions may differ, but the layout and file system information must be identical to the source disk. For cloning a boot disk, you must write a boot block to the target disk.

   It is possible to change partition layouts if you do not want all source partitions, but you will need to modify the target fstab file.

5. If you have AdvFS domains complete this step. Otherwise, go to step 6.

   Create domains for /, usr and var ensuring that the partitions are of equal or greater size. The following example assumes that the /var file system exists in /usr:

```
# mkfdmn /dev/disk/dsk1a root_tmp
# mkfdmn /dev/disk/dsk1g usr_tmp

# mkfset root_tmp root
# mkfset usr_tmp usr
# mkfset usr_tmp var

# mkdir /clone
# mount root_tmp#root /clone
# vdump -0 -f - / (cd /clone ; vrestore -x -f -)

# mount usr_tmp#usr /clone/usr
# vdump -0 -f - /usr (cd /clone/usr ; vrestore -x -f -)

# mount usr_tmp#var /clone/var
# vdump -0 -f - /var (cd /clone/var ; vrestore -x -f -)
```

   Next, correct the links in etc/fdmns. The copied version will be pointing to the original device special file. Change these links to point to the device special files for the newly created domains. For example:

```
# cd /clone/etc/fdmns/root_domain
# rm -r *
# ln -s /dev/disk/dsk1a .

# cd /clone/etc/fdmns/usr_domain
# rm -r *
# ln -s /dev/disk/dsk1g .
```

   If UFS is not in use on the source disk, go to step 7

6. If you have UFS file systems on the source disk, complete this step. Otherwise go to step 7.

Create a /clone mount point and mount the UFS partition (for example, a) of the target disk on /clone, as shown in the following example:

```
# mount /dev/disk/dsk1a /clone
```

Next, dump the partition as follows:

```
# dump -0u -f - /dev/disk/disk0a | \
(cd /clone ; restore -r -f -)
```

7. Verify file ownerships and that all required file system branches were dumped. The following diff command sequence will help you do this and provide a record of the dump:

```
# ls -R -l /clone > /newfiles
# cd /
# umount /clone
# ls -R -l  > /newfiles
# diff /newfiles /oldfiles > files.diff
```

8. If differences occur, remount the source and correct them. You can edit the files.diff file to create a script that you run to correct errors.

9. If you used this process to create a bootable clone disk, examine the /etc/fstab file before booting off the new disk. Make any necessary changes to partition mounts. Similarly, make any changes to swap in /etc/sysconfigtab.

10. To test the clone, shut down and halt the system, then reboot specifying the new boot disk as follows:

```
>>> show devices
```

Determine the SCSI address of the target, and its configuration device name, such as DKxNNN.

Boot from the cloned disk as follows:

```
>>> boot DKA200
```

11. If the boot is successful, and all system features appear to be functioning correctly, you can permanently swap the source and target disks by changing the appropriate console environment variables, physically swapping the devices, or using hwmgr.

The bootable tape utility described in Chapter 9 provides a method of creating a bootable standalone kernel on a magnetic tape. This method may enable faster recovery if you have problems with the root disk. Consider also some of the features offered by the Logical Storage Manager (LSM) that enable you to create a disk mirror as a copy of the root disk.

5.7.7 Monitoring Disk Use

To ensure an adequate amount of free disk space, you should regularly monitor the disk use of your configured file systems. You can do this in any of the following ways:

- Check available free space by using the `df` command
- Check disk use by using the `du` command or the `quot` command
- Verify disk quotas (if imposed) by using the `quota` command

You can use the `quota` command only if you are the root user.

5.7.7.1 Checking Available Free Space

To ensure sufficient space for your configured file systems, you should regularly use the `df` command to check the amount of free disk space in all of 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*

With no arguments or options, the `df` command displays the amount of free disk space on all of 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 path name to display the amount of available disk space on the file system that contains the file.

Refer to the `df`(1) reference page for more information.

Note

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.

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/dsk2g     101372   5376  85858     5%  /var
/dev/disk/dsk3c     394796     12 355304     0%  /usr/users
/usr/share/mn@tsts  557614 449234  52620    89%  /usr/share/mn
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.

5.7.7.2 Checking Disk Use

If you determine that a file system has insufficient space available, check 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. Refer to the du(1) reference page 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/morse
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
```

Note

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
```

Note

The character device special file must be used to return the information, because when the device is mounted the block special device file is busy.

Refer to the quot(8) reference page for more information.

6 Administering File Systems

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, you will find that AdvFS is configured as the default file system. Consult the *AdvFS Administration* guide for information on administering AdvFS.

If you installed the operating system yourself, you may have opted to create one or more UFS file systems. Even if your system arrived configured for AdvFS, you can still create UFS file systems. Both file systems can coexist on a system and many administrators opt to use the familiar UFS file system on system disks or in instances where the advanced features of AdvFS are not required. This chapter discusses system administration tasks related to the following file system topics:

- Section 6.1 provides an introduction to the file systems that are available.
- Section 6.2 describes Context-Dependent Symbolic Links (CDSLs), which facilitate the joining of systems into clusters.
- Section 6.3 describes how you create UFS file systems manually, using the command line.
- Section 6.4 describes how you create UFS file systems using the SysMan Menu tasks.
- Section 6.5 describes how you control UFS file system resources by assigning quotas to users.
- Section 6.6 provides pointers to methods of backing up UFS file systems.
- Section 6.7 briefly describes features for monitoring and tuning file systems.
- Section 6.8 provides information for troubleshooting UFS file system problems.

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

The UNIX operating system supports current versions of several file systems, including:

- Advanced File System (AdvFS). This file system has its own documentation and advanced interfaces. Refer to the *AdvFS Administration* guide and the `advfs`(4) reference page for more information. There are advanced administrative utilities available for AdvFS. When these utilities are available, there will be a launch icon named Advanced File System in the CDE Application Manager – Storage_Management folder. Consult the AdvFS documentation for information on installing and using the advanced administrative utilities.

  Basic AdvFS utilities are provided as SysMan Menu tasks. Refer to 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 also the `ufs_fsck`(8), `sys_attrs_ufs`(5), and `tunefs`(8) reference page for information on attributes and utilities.
- ISO 9660 Compact Disk File System (CDFS). Refer to the `cdfs`(4) reference page for information.
- Memory File System (mfs). Refer to the `newfs`(8) reference page for information on mfs.
- File on File Mounting file system (ffm). Refer to the `ffm`(4) reference page for information on mfs.

You may also need to refer to the following volumes:

- The *Logical Storage Manager* guide for information about using the Logical Storage Manager (LSM) with both the AdvFS and UFS file systems.
- The *AdvFS Administration* guide for information on converting file systems from UFS to AdvFS, and from AdvFS to UFS.
- The *System Configuration and Tuning* guide for information on advanced UFS file system tuning.

The rest of this section, and 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 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 a hard disk. The administration of such devices is described in Chapter 5. A disk is divided into logical partitions, which may be the whole disc (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 will 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 will be 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, invoke the appropriate reference page as follows:

- Regular files – Refer to the `file`(1) reference page.
- Directories – Refer to the `ls`(1) and `dir`(1) reference pages.
- Device Special Files – Refer to Chapter 5.
- Sockets – Refer to the `socket`(2) reference page, the *Network Administration* guide, and the *Network Programmer's Guide*.
- Pipes – Refer to the `pipe`(2) reference page.
- Symbolic links – Refer to the `link`(1) and `ln`(1) reference pages

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 the `hier`(5) reference page 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 will attach. 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. Refer to `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 also 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.

For example, the following example shows the default partitioning for a model RZ1DF-CB disk, using the following command:

```
# disklabel -r /dev/rdisk/dsk0a
```

Note that only the disk table part of the output is shown here. Also listed is an example of and HSZ RAID disk, taken from the rz(7) reference page.

Example 6–1: Default Partitions for RZ1DF-CB Disk and HSZ RAID Devices

```
(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. Refer to disklabel(8) for more information on the command-line interface. Refer to Chapter 5 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 newfs or diskconfig to create file systems

> When you create new file systems using newfs or diskconfig, the new file systems are always created as Version 3.0 (32k subdirectories or hard links) to minimize any incompatability problems.

Using fsck to check file systems

> When you use fsck to check a dirty file system (such as one not unmounted normally, or perhaps after a system crash), the file system will be marked as either Version 3.0 or Version 4.0, depending on the maximum number of subdirectories found. If fsck finds a directory with more than 32k subdirectories, the file system will be marked as Version 4.0. Otherwise, if fsckdoes not find a directory with more than 32k hard links, the file system will be marked as Version 3.0. A file system will normally be converted to Version 4.0 as soon as the 32k subdirectory limit is exceeded by a user.
>
> A new fsck otion, -B, has been 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 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 will be permenently deleted 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 will fail because of a superblock mismatch with alternate superblocks. When you are prompted to specify an alternate superblock, always respond `n`. Even if you inadvertantly enter `y`, the Version 4.0 file system will remain 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 will not exist. Any directories with more than 32k subdirectories will be permanently deleted.

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 UNIX. If you attempt to create more than 32k hard links on a Version 3.0 file system, it will be automatically converted to Version 4.0. The following example system message will be 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 a 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

This section discusses the structure of the UFS. The structure of the AdvFS is discussed in *AdvFS Administration*.

A UFS file system has four major parts:

- Boot block

  The first block of every file system (block 0) is reserved for a boot, or initialization, program.

- 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 of 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

- 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 have a maximum number of inodes; therefore there is a 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.

- 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 that processes use 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 are typically 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, refer to Chapter 5.

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 which are not currently in 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 which 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 the `hier`(5) reference page 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. The `installupdate`(8) reference page describes the update installation process.

  The TruCluster documentation describes the process of adding a system to a cluster and further explains how CDSLs are utilized on a running cluster. Note that this documentation is not part of the base documentation set.

- The `local`(4), `ls`(1), `ln`(1), and `hier`(5) reference pages provide reference information 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 must always 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 will 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 re-create 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/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 automatically set to zero. 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

Refer to 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 will 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 accidentally removed 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 Checking CDSL Inventory

Use the script /usr/sbin/cdslchkinv to check the CDSL inventory on a single system. Periodically, revise the inventory and check the CDSLs against it. See cdslchkinv(8) for information on cdslchkinv.

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:

- Check the value of {memb}, as defined by the sysconfigtab variable generic:memberid
- Check 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. Note that 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, refer to the *AdvFS Administration* guide.

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 Chapter 5. Refer to the hwmgr(8) and dsfmgr(8) reference pages 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 the disklabel(8), and fstab(4) reference pages for more information.)

2. Having identified which unused raw (character) disk partition you will use, you can determine the special device file name for the partition. For example, partition g on disk 2 will have a special device file named /dev/rdisk/dsk2g. (See Chapter 5 for information on device special file names.)

3. Use the newfs command to create a file system on the target partition. (See to the newfs(8) reference page 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 the mount(8) reference page for more information.)

5. Use the chmod command to check and adjust any access control restrictions. (See the chmod(1) reference page 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 also 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).

Refer to 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 will tell you what disk devices are attached to a system and what 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 physically known to the system 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 will be created automatically when a new disk device is added to the system. You will only need to create them manually under the circumstances described in Chapter 5.

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, the `c` partition cannot be used as 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` and cannot be used.

Table 6–1: Disk Partition Tables (cont.)

Partition Table	Description
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 and cannot be used
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 and cannot be used.

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 labelled for five partitions, and the f partition is unused and available to be used for 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 will not be able to create the file system and will return 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. Refer to the fsck(8) reference page 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 additional 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.

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 will be 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 automatically processed 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 checking during startup.

If you do not specify a pass number or if you specify 0 (zero), the file system is not checked. 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 will be processed in parallel based on the pass number assigned (with some exceptions). All entries with a pass number of 2 will be processed before pass number 3, pass number 3 will be 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
```

Refer to Chapter 5 and Chapter 12 and the swapon(8) reference page for more information on swapping and swap partitions.

6.3.4 Using the mount Command

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 will be temporary and will 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 must already 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
```

Refer to mount(8) for more information on general options and options specific to a file system type.

6.3.5 Using the umount Command

Use the umount command to unmount a file system. You must unmount a file system if you want to check it with the fsck command. If you want to change its partitions with the disklabel command, you must unmount the file system. Be aware, however, that changing partitions could destroy the 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.

6.4 Administering UFS File Systems Using SysMan

In addition to the manual method of file system creation and administration, the operating system provides some graphical tools, and also some SysMan tasks, which can be used in different user environments. Refer to 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. Click on 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 interface. refer to the *AdvFS Administration* guide for more information.

  See also the `dtadvfs`(8) reference page for information on launching the Advfs graphical interface from the command line.

- Bootable Tape – Select this icon to invoke the SysMan Bootable Tape Creation interface. Use this interface to create a bootable system image on tape. This image will contain a standalone kernel and copies of selected file systems that you specify during creation. You can recover the image using the `btextract` utility. Refer to Chapter 9 for information on using the bootable tape interfaces. See also the `btcreate`(8), `btextract`(8), and `bttape`(8)reference pages. The `bttape` command is used to launch the bootable tape graphical 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 interface. Logical Storage Management enables you to create virtual disk volumes that appear as a single device to the system and any applications. Refer to the *Logical Storage Manager* guide for more information and the `lsm`(8) reference page for a list of LSM commands.

  Tp nvoke this interface from the command line, use the `dxlsm` command. Refer to the `dxlsm`(8) reference page for more information.

- 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. The stored data is then written to disk asynchronously as needed or when the machine is halted. Refer to the *Guide to Prestoserve* for more information and the `presto`(8) reference page for information on the command-line interface.

  To invoke this interface from the command line, use the `dxpresto` command. Refer to the `dxpresto`(8) reference page 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
         - Logical Storage Manager (LSM) Utilities
            | Initialize the Logical Storage Manager (LSM)
         - UNIX File System (UFS) Utilities
            | Create a New UFS File System
```

Each option provides a step-by-step interface to perform basic file system administrative tasks. Refer to Chapter 1 for information on invoking and using the SysMan Menu. You can also launch the file system utilities from the SysMan Station. 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. Refer to Chapter 1 for information on invoking and using the SysMan Station. Refer to 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 – Refer to the *AdvFS Administration* guide.
- Logical Storage Manager (LSM) Utilities – Refer to the *Logical Storage Manager* guide.

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. Refer to the `mount`(8) and `umount`(8) reference pages 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 will be 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.

Press the Apply button to dismount the file system and continue dismounting other file systems, or press 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. Refer to Chapter 5 for information on device names and device special files.
  - An NFS (Network File System) mounted file share, possibly mounted using the automount utility, which automatically mounts exported networked file systems when a local user accesses (imports) them. Refer to the *Network Administration* guide for information on NFS. 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
    ```

    Where 19serv: is the host name identifier followed by a colon, /share/19serv/tools/tools is the pathname to the exported directory and /tmp_mnt/19serv/tools is the temporary mount point that is automatically created by NFS.
  - An AdvFS domain name such as usr_domain#var. Refer to *AdvFS Administration* or the advfs(4) reference page 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. Note that 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 will not be included in this list.

The following option buttons 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 will be prompted to confirm the dismount request. Note that you may be unable to 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 file systems that were dismounted. Note that if you mount file systems using the command line, or if NFS mounts are established, these newly mounted systems will not be displayed until you exit the utility and invoke it again.
- OK – press this button 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. Refer to the mkdir(1) reference page for information on this command.
- The file system must be created on a disk partition, and the disk must be on line. Refer to 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. Refer to the newfs(8) reference page for information on this command. Information on creating AdvFS file systems is located in the *AdvFS Administration* guide.

  The diskconfig graphical utility provides a way to customize disk partitions and write a file system on the partition in a single operation. Refer to Chapter 5 for information on the diskconfig command, and see the diskconfig(8) reference page for information on launching the utility. You can also launch this utility from the SysMan Menu or SysMan Station and form the CDE Application Manager.
- 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 check the status of disk devices and partitions for existing disks (if necessary). Refer to the hwmgr(8) reference page for information on this command. Refer to Chapter 5 for information on administering devices.

  Normally, the device special files for a disk partition, such as /dev/disk/dsk5g, are automatically created and maintained by the system. However if you do not find the device special file, you may need to create it.

  Refer to Chapter 5 for information on the dsfmgr command, and see the dsfmgr(8) reference page 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 the mount(8) reference page. 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 options, which you select by clicking on the button:

1. Mount a specific file system

   Select this option to mount a single specific file system. The File System name and Mount Point window will be 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 will be 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 online, and all the mount points have been created.

   The File System Mounting Options window will be 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 online, and all the mount points have been created.

   You specify the file system type in the File System Mounting Options window, which will be 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 will be 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 – Click on the appropriate button for 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. Refer to *AdvFS Administration* or the `advfs`(4) reference page for more information.
  - UFS – Select this option to specify a UNIX File System type. Refer to Section 6.1.4 for a description of this file system.
  - NFS – Select this option to specify a Networked File System. Refer to the *Network Administration* guide and the `nfs`(4) reference page for more information.
  - CDFS – Select this option to specify a Compact Disk Read Only Memory File System. Refer to the `cdfs`(4) reference page 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). Refer to the `mount`(8) reference page for more information on supported file systems, and see the individual file system reference pages, such as `mfs`(4) 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. Refer to the `mount`(8) reference page for more information on the various options.

When you have entered the options you want, use the Finish button to process the mount operation and return to the SysMan Menu options. Use the Back button to return to the Mount Operation window and process new mount operations, or the Cancel button to abort the mount operation.

If data in any field is incomplete or incorrect, you will be 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 via NFS (Network File System). Note that if the Advanced Server for UNIX (ASU) is installed and running, you may have further options to share file systems with PC clients. Refer to the ASU *Concepts and Planning Guide*.

You may also have 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`. Refer to the *Network Administration* guide for information on configuring your system to allow incoming connections to shared file systems.

You can also manage shared file systems using the `dxfileshare` graphical interface, which can be launched 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. Refer to the `dxfileshare`(8) reference page for more information on invoking the interface.

The only prerequisite for shared file systems is that you should have already 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). 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).
```

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. Press the Add... button to add a directory to the list.

2. A window titled Share Local Directory: Add Local Directory is displayed next. Complete the fields as follows:

   - In the field labeled Share This Directory: type the directory pathname, such as `/usr/users/share/tools`.
   - Choose whether to share the directory with read/write access or read-only access. The Read/Write check button is selected by default.
   - Choose whether to share the directory with all qualified hosts (remote systems) or only with named hosts as follows. For all hosts, check the All button. For selected hosts, check the Selected button and then add hosts to the Selected Hosts With Access list as follows:
     – Enter the host name and address, such as `dplhst.xxx.yyy.com`. Note that the host must be known to your local host, either through the `/etc/hosts` file or via a domain name server (DNS). Refer to the *Network Administration* guide for more information.
     – Select OK to validate the data and close the dialog box and return to the window titled Share Local Directory on *host name*. Note that all changes are deferred until you press OK in this window. When you press OK, the directories are made available for sharing.

To remove a share, you use the same utility as follows:

- Deleting hosts from the access list
- Modifying access to shared file systems by changing the read/write permissions or removing selected hosts from the access list
- Deleting 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:

1. Temporarily, where the mount will not persist across a reboot. A mount point will be created and the file system will be connected for the current session. If the system is shut down for any reason, the mount point will persist but the file-system connection will be lost and will not be reestablished when the system is booted.

2. Permanently, by specifying the shared NFS file systems in your local /etc/fstab file. For example, your /etc/fstab file may already 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.)

3. Automatically on request from a user, using the NFS automount utility. Refer to the *Network Administration* guide and the automount(8) reference page for information on using this option. Using automount will enable your local users to transparently mount any file systems that are shared with (exported to) your local system. You will not need to constantly respond to mount requests from users.

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.

Refer to the *Network Administration* guide for information on configuring the network and NFS. See Section 6.4.5 for a description of the process of sharing (exporting) file systems using the SysMan Menu options.

You can also manage shared file systems using the dxfileshare graphical interface, which can be launched 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. Refer to the dxfileshare(8) reference page for more information on invoking the interface.

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*, you will see a list of existing available NFS shared file systems listed in the /etc/fstab file, which provides you with the following information:

   a. Directory and Host – The name of the host, and the directory it is exporting to your local system.

   b. Mounted On – The local mount point on which the shared file system is mounted. This is a directory pathname, such as /tools/bin/imaging.

   c. Options – The access options for the directory, which can be:

- Read/Write – Allows users to both read data from and write data to the shared file system. Note that this may be dependent on access conditions set by the exporting host.
- Read-Only – Allows users only to read data from the shared file system.

d. Reboot – Indicates whether the mount will be reestablished if the system is shut down for any reason, and can be:

- true – Permanent; the entry is in the local /etc/fstab file and the mount will persist across reboots.
- false – Temporary; the entry is not in the local /etc/fstab file and the mount will not persist.

To add a file system to the list of NFS-shared directories, press the Add... button. A window titled Mount Network Directory: Add Network Directory will be displayed.

When you use this option, file systems will be mounted with the options hard (retries until a response is received) and bg (background mount) by default. Refer to the mount(8) reference page for more information on these options.

Follow these steps to add an NFS-shared file system:

1. Remote Host Name – Enter the name of the host sharing the file system. This can be the fully qualified name, such as ntsv.aaa.bbb.com or an alias listed in your /etc/hosts file.
2. Remote Directory Path – Enter the directory pathname of the share, such as /tools/toolbox/admin. You may need to verify this information from the /etc/exports file entries in the remote host.
3. Local Mount Point – Enter the pathname to the mount point that you want to use on the local host. This need not be the same as the remote pathname, but might be something that will indicate what is mounted, for example: /tools/remote_admin_tools.

   If the mount point does not exist, you will be given the option to create it.
4. Access Permission – Specify the user access to the file system as follows:
   - Read/Write – Allows users to both read data from and write data to the shared file system. Note that this may be dependent on access conditions set by the exporting host.
   - Read-Only – Allows users only to read data from the shared file system.
5. Mount on Reboot (put in /etc/fstab) – This checkbox determines whether the mount is permanent or temporary as follows:

- Checked – Permanent; the entry is in the local `/etc/fstab` file and the file system will be remounted when the system is rebooted.
- Unchecked – Temporary; the entry is not in the local `/etc/fstab` file and the file system will not be remounted when the system is rebooted.

Press the OK button to validate the share and return to the previous window. Press the Apply button to validate the share and continue adding more NFS-shared file systems. (Press Cancel to abort the operation and return to the previous window).

Permanent changes are deferred until you return to the Mount Network Directory on *hostname* and press OK. When you choose the OK option, the file systems will be mounted.

The Mount Network Directory (`/etc/fstab`) option is also used for the following tasks:

- Modify... – A window titled Mount Network Directory: Add Network Directory will be displayed, enabling you to change details of an existing share mount entry, such as changing the user access from the Read-only option to the Read/Write option.
- Delete – Select one of the listed share mounts and press this button to remove it from the list. Select OK to unmount the file system and remove it from the `/etc/fstab` file. Note that it may not always be possible for an unmount operation to complete. For example a user may be accessing the directory at the time the unmount command is issued. You should verify that the file system was unmounted and if necessary use the option described in Section 6.4.2.

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 Chapter 5.

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. Refer to the *Logical Storage Manager* 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. Refer to the `disktab`(4) reference page 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. Refer to the `newfs`(8) reference pages for a complete list of options. You can also 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. Partition or LSM Volume – Type the name of the disk partition or LSM volume that you selected to store the file system
2. Disk type – Type the name of the disk model, such as RWZ21.
3. Advanced newfs options – Enter any option flags, such as `-b 64` for a 64 kilobyte block size.

   Note that if you are unsure what options to use, clear all fields and press the Apply button. This will display a `newfs` information window, containing a list of flag options.

Press the OK button to create the file system and exit to the SysMan Menu or press the Apply button to create the file system and continue creating more file systems. To abort the operation, press 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

This section describes 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 *AdvFS Administration*, 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 also known as file system quotas. The file system quotas are also known as disk quotas because, when established, they limit the number of disk blocks that can be 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 will 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 also write the file to another file system, such as `/tmp`, 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`. Refer to the `fstab`(4) reference page for more information.
3. Use the `quotacheck` command to create a quota file where the quota subsystem stores current allocations and quota limits. Refer to the `quotacheck`(8) reference page 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 will affect each user or group. You can also use prototypes that allow you to quickly set up quotas for groups of users as described in Section 6.5.3.

5. Use the quotaon command to activate the quota system. Refer to the quotaon(8) reference page for more information.

6. To check 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. Refer to 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 equivalence 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 periodically verify your quota system. 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.

Refer to `quotacheck`(8), `quota`(8), 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 *AdvFS Administration*. 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 check UFS file systems. They also include some basic information on file system tuning. For a more detailed discussion of tuning, refer to the *System Configuration and Tuning* guide.

6.7.1 Checking UFS Consistency

The fsck program checks 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 also 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 checked. If you specify a file system, you should always use the raw device.

Refer to the fsck(8) reference page for information about command options.

Note

To check 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 regularly monitor the disk use of your configured file systems. You can do this in any of the following ways:

- Check available free space by using the df command
- Check 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 Checking Available Free Space

To ensure sufficient space for your configured file systems, you should regularly use the df command to check the amount of free disk space in all of 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* ...]

With no arguments or options, the df command displays the amount of free disk space on all of 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.

Refer to 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 Checking Disk Use

If you determine that a file system has insufficient space available, check 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. Refer to 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.

Refer to 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 *opt*] [*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 also reboot to apply the changes.

You can use command options to specify the dynamic parameters that affect the disk partition layout policies. Refer to tunefs(8) for more information on the command options and to sys_attrs_ufs(5) for information on UFS subsystem attributes.

6.8 Troubleshooting File Systems

The following tools can be used to react to 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* for an explanation of the file-system related administration you can accomplish with it. The option can be used 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. Information about the program is in fsx(8). Other programs in the directory, such as a tape exerciser (tapex) and a disk exerciser (diskx) might be useful when investigating file system problems.

- The dumpfs utility displays information on UFS file systems. Refer to the dumpfs(8) reference page.

- EVM (the Event Manager) can be used 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. Refer to Chapter 13 for information.
- The SysMan Station and Insight Manager provide graphical views of file systems and can be used to monitor and troubleshoot file system problems, such as lack of disk space. Refer to 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 will assign system resources to users. This chapter describes user account and group administration, organized into the following sections:

- Section 7.1 describes the different utilities that you can use to administer accounts and groups, and the user environments in which you can use these utilities. Any general constraints on use are also identified.
- Section 7.2 is a quick start section, providing brief information on the utilities. You can use the on-line help to guide you through a task.
- Section 7.3 provides information to help you understand general account and group concepts, and important data items such as the unique identifiers assigned to accounts and groups. This section also describes the contents of the data files for passwords and groups and setting the default characteristics of an account or group.
- Section 7.4 provides 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 7.5 provides specific instructions on using utilities to perform administrative tasks on user groups.
- Section 7.6 provides information on administering associated (synchronized) Windows NT domain and UNIX accounts.

7.1 Account Administration Options and Restrictions

Depending on how your local system is configured, the user environment, and your personal preferences, there are several methods of administering accounts and a number of different utilities that you can use. The following sections introduce and describe your 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. Refer to the on-line 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 UNIX or the Windows NT domain Administrator to use these utilities.

Table 7–1: Utilities for Administering Accounts and Groups

Utility	User Environment
SysMan Menu, Accounts options. Manage local users and groups. Manage NIS 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. It does not allow you to select the creation or deletion of associated (synchronized) accounts but does this automatically, depending on how the account defaults are configured (with `useradd` or `usermod`).
Account Manager, (`dxaccounts`).	Any X11-compliant user environment, such as CDE (the default UNIX environment). This is a graphical user interface that provides most user and group administrative options for both UNIX and Windows NT domain accounts.
`useradd`, `usermod`, `userdel`.	Character-cell terminal on UNIX. These are command line options that run on the UNIX system, providing you with access to all user account administrative tasks. You can use these commands to administer both UNIX and associated (synchronized) Windows NT domain accounts. You can also use these commands to configure the default account environment.
`groupadd`, `groupmod`, `groupdel`.	Character-cell terminal on UNIX. These are command line options that run on the UNIX system, providing you with access to all user group administrative tasks. You can use these commands to configure the default UNIX group environment.
Advanced Server for UNIX. User Manager for Domains.	A Microsoft Windows NT based application for a PC system. This utility 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.
Advanced Server for UNIX. `net` commands.	Commands that can be entered at a UNIX terminal or at the DOS prompt on a system running the Windows NT server. These commands replicate the behavior of the User Manager for Domains utility.

You must install and configure the Advanced Server for UNIX (ASU) software to use the Microsoft Windows-based utilities. ASU provides the following utilities that you can use to manage domain user accounts.

- User Manager for Domains, a Microsoft Windows application
- The `net` command line options which you can run in an MS-DOS window or UNIX terminal.

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. Refer to the ASU *Installation and Administration Guide* for more information on installing and using ASU.

7.1.2 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 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 also create SSO user groups using the same software and administrative tools.

7.1.2.1 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 contoller.
- The UNIX system cannot be running C2 level security. Refer to the *Security* guide for more information on security levels.
- You need the *Associated Products Volume 2* CD-ROM on which you will 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.

7.1.2.2 Installing the Single Sign-On Software

Install the software as follows:

1. Load the CD-ROM into the reader as described in the *Installation Guide*.
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*.

7.1.2.3 UNIX Requirements for Creating Single Sign-On Accounts

The following requirements for UNIX account characteristics apply to SSO accounts:

- You can only create SSO user accounts in the Windows 2000 user environment using a modified version of the standard Windows 2000 user management tools. You cannot create SSO accounts using UNIX tools such as `dxaccounts` or `useradd`.

  Note that you can upgrade existing Windows 2000 accounts to provide account holders with SSO privileges for UNIX resources.

- There are terminology differences between UNIX 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 UNIX. 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. Refer to

Section 7.3.2 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 account name 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 account 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 account 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 may vary depending on the security mechanisms in force. Refer to the *Security* guide for more information.

UID and *GID*

Each account requires a unique identification integer called a UID and each group has a GID. Refer to Section 7.3.1 and Section 7.3.3 for a description of these identifiers.

User Comment

This field enables you to enter a text description of the account 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`) . Refer to the `shells`(4) reference page and Section 7.2.7 for more information.

7.1.2.4 Creating Single Sign-On Accounts and Groups

Using the information prepared in Section 7.1.2.3, 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 display the Active Directory Users and Computers Window.
3. Open the `Users` folder and either select an existing user or open the Action menu and choose the New option then the User option.
4. Three dialog boxes are displayed 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 7.1.2.3, 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.

7.1.2.5 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 accidentally killed or stopped, 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.

Refer to the file headers and the *Windows 2000 Single Sign-On Installation and Administration Guide* for more information on these files.

7.1.3 Restrictions on Using the Utilities

The following restrictions apply when using account management utilities, or when certain system features are enabled:

- To configure the default UNIX account and group characteristics, you can only use the UNIX command utilities or `dxaccounts`. Refer to the ASU *Installation and Administration Guide* for more information on setting default values for PC accounts when ASU is in use.
- When enhanced security is enabled, it places restrictions on account creation and enables additional features such as enhanced passwords. Refer to the *Security* guide for more information.
- The Network Information Services (NIS) service enables users to log in to any system in the local network that is running NIS. User data such as account name and password is shared between all NIS systems and users will 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 and NIS users and groups. Features in the user account administration utilities that support NIS become enabled only when NIS is running. Refer to the *Network Administration* guide for information on setting up the NIS environment.
- 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. This condition is required to prevent corruption of the system files. When invoked, an account management utility creates a lock file, preventing other 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`.

7.2 Account Administration - Quick Start

This section provides 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 might 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 you can 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 automatically displayed to guide you through the options for configuring your system. The Account Manager icon included in System Setup enables you to configure initial accounts. This icon invokes the Account Manager (`dxaccounts`) graphical user interface (GUI). This is an X11–compliant GUI that can be used in CDE or other X-windowing environments. See Section 7.5.2 for information on using the Account Manager.

When the Advanced Server for UNIX (ASU) is installed and configured, you can also use this GUI to administer Windows NT domain accounts as described in Section 7.6.

Using `dxaccounts` to administer UNIX accounts is described in Section 7.4.2.

7.2.2 Using the SysMan Menu Accounts Option

The SysMan Menu Accounts options provide the same functions as `dxaccounts`, but with limited support for managing Windows NT domain accounts for PC clients. You can invoke these 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 a NIS (Network Information Service) environment. You can add local users to any system without adding them to the NIS environment. Refer to the *Network Administration* guide 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. Select Accounts to expand the menu options. The following options are displayed:
   - `Manage local users`
   - `Manage local groups`
   - `Manage NIS users`
   - `Manage NIS groups`
2. Move the pointer (or use the Tab key) to select an option. Press mouse button 1 or the Enter key to invoke the utility.
3. The first window (or screen) of the utility is displayed. Press the Add... button to begin creating an account and follow the online instructions.

Use of these utilities is described in Section 7.5.1, or in the online help.

7.2.3 Using the dxaccounts GUI

The X11–compliant graphical user interface (GUI) `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. Select Daily Administration, and click on the Account Manager icon.

The `dxaccounts` GUI also provides options for administering Windows NT domain users when ASU is installed. These options are grayed out on the window if ASU is not installed and configured.

You can also use the Account Manager to configure default options for user accounts, such as the shell and the parent directory. See Section 7.4.2.5 for information.

7.2.4 Using the Command-Line Utilities

The following command-line utilities are available for administering accounts and groups:

- `useradd`, `userdel`, and `usermod` – Use these commands to add, modify, and delete user accounts.
- `groupadd`, `groupdel`, and `groupmod` – Use these commands to add, modify, and delete groups.
- The `adduser` and `addgroup` 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 UNIX. 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 Guide* for information on installing the software subsets and the *Installation and Administration Guide* for information on configuring ASU for use.

7.2.6 Related Documentation

The following documentation contains information on administering accounts.

- Books
  - Refer to Chapter 6 for information on file systems and user file space.
  - The *Network Administration* guide provides information on NIS user accounts.
  - The *Security* guide 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 *Technical Overview* provides information on maximum system limits for numbers such as UIDs and GIDs.
  - The *Concepts and Planning Guide*, *Installation and Administration Guide*, and *Release Notes* provide information on ASU.
- 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).
  - The Account Manager is documented in `dxaccounts`(8)

– The system files are documented in passwd(4), group(4), shells(4), default(4),

– Individual commands are documented in passwd(1), vipw(1), grpck(8), and pwck(8).

- Online help – The SysMan Menu Accounts options and dxaccounts provide online help files 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.

7.2.7 System Files

The following system files may be updated when you perform account administration tasks and should be backed up regularly:

- 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 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:
.
.
.
.
```

- The /etc/passwd file consists of rows of one record (row) per user, containing seven fields of user data. See Section 7.3.2 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 /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.

- If enhanced security is in use, the following security files are relevant: /etc/auth/system/default, /tcb/files/auth.db, and /var/tcb/files/auth.db.
- If NIS (Network Information Services) is in use, the following NIS files are relevant: /var/yp/src/group, /var/yp/src/passwd, and /var/yp/src/prpasswd.
- 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.

7.2.8 Related Utilities

The resources in the following list are also useful when administering accounts. These commands may be useful in correcting system problems when the graphical user environments are unavailable, such as after a system crash, or if you only have access to a simple character-cell terminal.

- The vipw utility, documented in vipw(8), allows you to invoke a text editor to edit the password file manually. Note that you should avoid manually editing system files if possible, and 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 command allows you to edit the passwd file and at the same time locks the file to prevent others from modifying it. It also does consistency checks on the password entry for root and does not allow a corrupted root password to be entered into the passwd file.

  The vipw utility can be used to patch a corrupted passwd file when in standalone mode.
- A number of commands, such as who(1) and finger(1), provide information on user activities and account information.
- The csh, ksh, and sh commands invoke and interpret the C, Korn, and POSIX shells.
- The grpck and pwck utilities can be used to check the integrity of the group and passwd files.
- The quotaon command is used to turn quota information on and off.
- The passwd, chfn, and chsl commands provide the same functions as password options in command utilities, such as usermod and thedxaccounts Password option.

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) is for central account and group management. NIS allows participating systems to share a common set of password and group files. See the *Network Administration* manual for more information.

If enhanced 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* manual.

7.3.1 Understanding Identifiers – UIDs and GIDs

Each user is known to the system by a unique number called a user identifier (UID). The system also knows 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, note that some older utilities and applications will not support this number of UIDs and GIDs and you may need to take other precautions as follows:

- If you are running applications that have not recently been upgraded to the latest version, 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, you should probably upgrade to Kerberos Version 5.0. Similarly, If you use PATHWORKS, you should upgrade to ASU (Advancer Server for UNIX) 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 will return an `EINVAL` error. Users assigned a UID or GID greater than 65,535 will not be able to 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 changed when the maximum UID and GID range was increased. Check 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 will incorrectly copy 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 a UFS file system that contains quota files or other sparse files is backed up using the `vdump` utility and restored using the `vrestore` utility, the quota files or other sparse files will be restored as follows:
    - ☐ The first page of a file on disk will be restored as a fully populated page; that is, empty nonallocated disk blocks will be zero filled.
    - ☐ Any additional pages on disk will be restored sparse.

7.3.2 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 is an integer, the maximum value of which is defined in the *Technical Overview* (See also the `/usr/include/limits.h`) file. 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.

group_id — The GID for this account, which is an integer. Refer to 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) — 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. Refer the finger(1) and chfn(1) reference pages 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. Refer to the chown(1), mkdir(1), chmod(1), and chgrp(1) reference pages for additional information on creating a login directory.

login_shell — The absolute pathname of the program that starts after the user logs in. Normally, a shell starts. If you leave this field empty, the Bourne shell /bin/sh starts. Refer to the sh(1b) reference page for information on the Bourne shell. Users can change their login shell by using the chsh command. Refer to the chsh(1) reference page 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.

Note that you can only 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 7.4.2.5 for an explanation of how to do this with Account Manager (dxaccounts).

When the /etc/passwd file is very large, a performance degradation may occur. If the number of passwd entries exceeds 30,000, mkpasswd will sometimes fail 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 will be unable to read from or write to the password database correctly. This would cause the application to fail either by generating incorrect results or by possibly dumping core.

Any statically linked application would be affected if it directly or indirectly calls any of the libc ndbm routines documented in the ndbm(3) reference page and then accesses the password database. To remedy this situation, you must relink the application. If the mkpasswd -s option is avoided, you will not see this compatibility problem.

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. Refer to nis_manual_setup(7) for more information on adding users in a distributed environment.

7.3.3 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. Refer to 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 as 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 limitation on the number of groups that a user can be in, as documented in group(4). reference page. The maximum line length is LINE_MAX as defined in the limits.h file. User accounts should be divided into a number of manageable groups.

Note that you can also set defaults for certain GID values using the graphical or command-line utilities. See Section 7.4.2.5 for an explanation of how to do this with Account Manager (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 a NIS (Network Information Service) environment.

- Administer local and NIS users and associated Windows NT domain accounts using `dxaccounts`.

The process for using the `useradd` command-line utility is similar and is documented in the reference pages but does not support NIS accounts. Refer to the *Network Administration* for information on NIS. Note that the SysMan Menu Accounts options can also be used from a terminal.

Note

Avoid using `adduser` because it does not provide all the available options and is not sensitive to security settings. You should also avoid manual methods of adding user accounts to preserve the integrity of system files.

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
- Creating a new local or NIS user account
- Modifying an existing local or NIS account record
- Deleting a local or NIS user account

For information on how to 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 from the worksheet provided in Table 7–2. Note that if enhanced security is in use, you must make the data items comply with the minimum requirements (such as password length). Refer to the *Security* guide for more information. Items marked O in the table are optional data.

Refer to Section 7.3.2 for an explanation of the `passwd` file data items.

Table 7–2: Account Administration Worksheet

Data Item	O	Note	New Account
User Name			
Comments (gecos)	O	Full name	
Comments	O	Location	
Comments	O	Telephone	
Comments	O		
User ID (UID)	O		
Password		Use mixed case or alphanumeric	
Primary Group (GID)			
Secondary Group(s)	O		
Shell	O	Can be chosen	
Home Directory		Can be created automatically	
Lock Account	O		
Local User	O		
NIS User	O		
Windows User	O	Shares needed	

An example of typical user data is provided in Table 7–3.

Table 7–3: Account Administration Worksheet

Data Item	O	Note	New Account
User Name			carsonK
Comments (gecos)	O	Full name	Kit Carson
Comments	O	Location	Office 3T-34
Comments	O	Telephone	4-5132
Comments	O		Project Mars
User ID (UID)	O		Use next available
Password		Use mixed case or alphanumeric	Use site specific initial password
Primary Group (GID)			Users
Secondary Group(s)	O		marsx, 25
Shell	O	Can be chosen	ksh
Home Directory		Can be created automatically	/usr/marsx/carsonK
Lock Account	O		no
Local User	O		no
NIS User	O		yes
Windows User	O		yes, share \\maul\astools

7.4.1.2 Creating or Modifying Local Accounts

To create a new account, invoke the SysMan Menu and select the `Manage local users` option as described in Section 7.2.2. A table of local users is displayed, listing all the existing local user accounts. Use the following procedure to add a local user:

1. Select the Add... option to display the Manage Local Users: Add a User window.
2. Complete the data fields using the information from the worksheet described in Table 7–2.
3. If additional NIS options are required, select Options and enter the appropriate NIS values. Then press OK to return to the Add a User window.
4. Press OK to add the new user. You will be prompted to correct any errors, such as mistyped password confirmations.

5. You return to the Local Users window. A message confirming the successful addition is displayed. Press OK to return to the SysMan Menu.

To modify an existing account, invoke the SysMan Menu and select the Users option as described in Section 7.2.2. The Local Users table is displayed, listing all the existing local user accounts. Use the following procedure to modify a user entry:

1. Scroll through the list of users and select an entry.
2. Select the Modify... option to display the Account Manager: Modify a User window.
3. Change the contents of data fields as required.
4. If additional NIS changes are required, select Options and enter the appropriate NIS values. Then press OK to return to the Add a User window.

   To add or modify more than one account, click Apply instead of OK. All changes are deferred until you select OK to exit.
5. Press OK to confirm the changes. You will be prompted to correct any errors, such as mistyped password confirmations.
6. You return to the Local Users window. A message confirming the successful addition is displayed. Press OK to return to the SysMan Menu.

Online help provides explanations for the fields, and defines valid data.

7.4.1.3 Deleting Local Accounts

The following considerations may apply before deleting accounts:

- You can simply lock an account with the Modify... option, and later transfer the account to another new user using the Modify... option to change some account details.
- You may want to invoke `dxarchiver` before deleting the account, to create a compressed archive file of the user's directories and files. See the `dxarchiver`(8) reference page for more information.

To delete an account, select the Users option as described in Section 7.2.2. The Local Users table is displayed, listing all the existing accounts. Use the following process to delete a user:

1. Scroll through the list of users and select an entry.
2. Select the Delete... option to display the Account Manager: Delete a User window.

3. Optionally, choose Delete User’s Directory and Files if you want to remove the user’s resources and recover the disk space.
4. Press OK to delete the account. The list of local users is updated immediately.

7.4.1.4 Creating or Modifying NIS Accounts

To create a new NIS account, invoke the SysMan Menu and select the `Manage NIS Users` option as described in Section 7.2.2. The NIS Users table is displayed, listing all the existing local user accounts. Use the following procedure to create an account for a local user:

1. Select the Add... option to display the Manage NIS Users: Add a User window.
2. Complete the data fields using the information from the worksheet described in Table 7–2.
3. Press OK to add the new user. You will be prompted to correct any errors, such as mistyped password confirmations.
4. You return to the Manager NIS Users window. A message confirming the successful addition is displayed. Press OK to return to the SysMan Menu.

To modify an existing account, invoke the SysMan Menu and select the `Manage NIS Users` option as described in Section 7.2.2. The NIS Users table is displayed, listing all the existing local user accounts. Use the following procedure to modify a user entry:

1. Scroll through the list of NIS users and select an entry.
2. Select the Modify... option to display the Manage NIS Users: Modify a User window.
3. Change the contents of data fields as required.
4. Press OK to confirm the changes. You will be prompted to correct any errors, such as mistyped password confirmations.

   To add more than one account, click Apply instead of OK. All changes are deferred until you select OK to exit.
5. You return to the Local Users window. A message confirming the successful addition is displayed. Press OK to return to the SysMan Menu.

Online help provides explanations for the fields, and defines valid data.

7.4.1.5 Deleting NIS Accounts

To delete a NIS account, select the Manage NIS Users option as described in Section 7.2.2. The NIS Users table is displayed, listing all the existing accounts. Use the following process to delete a user:

1. Scroll through the list of users and select an entry.
2. Select the Delete... option to display the Manage NIS Users: Delete a User window.
3. Optionally, choose Delete User's Directory and Files if you want to remove the user's resources and recover the disk space.
4. Press OK to delete the account. The list of NIS users is updated immediately.

7.4.2 Using dxaccounts

Invoke `dxaccounts` as described in Section 7.2.3. The Account Manager on <host> window is displayed 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 `dxaccounts`. The processes are identical for administering NIS users, except that you must also be authorized to make changes to the NIS databases. Any options that do not affect the databases are available to all users, such as Find. Refer to the *Network Administration* guide for more information on NIS.

Note that If ASU is installed, additional options are displayed on the `dxaccounts` windows that enable you to administer accounts in Windows NT domains and create associated UNIX accounts simultaneously. Refer to the *Installation and Administration Guide* for more information on ASU.

7.4.2.1 Adding and Modifying Accounts

The same window is used to add or modify user accounts. If the account is new, you begin by clicking on the Add button. If the account is existing, you double-click on the user's icon. To add or modify accounts:

1. If the current view is not Local Users, pull down the View menu and choose the Local Users option.
2. Choose the Add button to display the Add/Modify Local User window and press the Add button.

   (To modify an existing account, double-click on the user's icon.)
3. Enter the new user name in the Username field.
4. You can opt to choose the next available GID, or enter a GID.

5. Use the pull-down menu to select the primary group, or clear the box and type a group name.

   If secondary groups are required, choose the Secondary Groups... button. 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. Press 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 optionally 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 box.

11. Press OK to create the account and return to the Account Manager main window. You will be prompted to correct any errors. The Current View is updated with an icon for the new user.

An alternative method of creating a new account is to clone it from an existing group as follows:

1. Click on an existing user icon to highlight it.

2. Choose the Copy button to copy the account.

3. Choose the Paste button to paste a new account version. The new icon label will have 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 User window. The Modify button is selected automatically.

5. Make the required modifications to the Account as follows:

   - Enter the new user name
   - Change the UID (or select the next available)
   - Change the password

6. Make any optional changes, such as Comments or Lock Account.

7. Press OK to add the modified account and return to the Account Manager on <host> window. This window is immediately updated with an icon for the new account.

7.4.2.2 Deleting Accounts

Invoke the `dxaccounts` utility as described in Section 7.2.3. The Account Manager on <host> window is displayed first.

1. Double-click on the required user's icon to highlight it.
2. Press the delete button. The Delete Local UNIX User window is displayed. You can opt to remove the user's files and directories at this time. (You may want to archive these. See the `dxarchiver` option.)
3. Press OK to confirm the deletion and return to the Account Manager on <host> window. This window is immediately updated, removing the deleted user.

7.4.2.3 Finding and Selecting Accounts

The `dxaccounts` utility provides a useful search feature to locate user accounts. You can also use this feature to select 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 7.2.3. The Account Manager on <host> window is displayed first.

1. Press the Find button.
2. Enter a search string in one of the fields (a text string) and press OK.

   The Find option will select and display all accounts where the data in the search field contains the search string. For example:

   - Enter the string `ad` in the Username field and press OK.
   - The Selected Users window is displayed, stating that the following users matched the search criteria.
   - The matched users include `adm`, `admin`, `adamK`, and `wadmanB`; these groups are highlighted in the Current View.

You can now select the modify (or delete) option to perform global operations on the selected users.

7.4.2.4 The Password Option

The dxaccounts utility provides an option to easily change or remove passwords for a single user or a group of users as follows:

1. Select the user or users (the Find option may be useful in selecting groups of users).
2. From the Edit menu, select Password.
3. In the New Password window, enter and confirm the new password. You can also opt for No password, although note the security implications of this option.
4. Press OK to confirm the change and return to the Account Manager main window.

7.4.2.5 Account Manager General Options

The Account Manager enables you to easily set defaults for newly created user accounts. Use the following procedure to add or modify defaults. Note that you can also set these through the command line, but not with the SysMan Menu options.

1. From the Options menu, select General..... The General Options window is displayed, enabling you to set the following defaults:

   a. Duplicates Policy – These options enable you to allow duplicate User Identifiers (UID) and Group Identifiers (GID).

   b. ID Ranges Policy – These options enable you to control the minimum, next, and maximum UID and GID.

   c. Default Primary Group – This option enables you to set the default primary group to a group other than users.

   d. Default Primary Group – This option enables you to set the default home directory to a location other than /usr/users.

   e. Default Shell for User – This option enables you to set the default login shell.

   f. Default Primary Group – This option enables you to set the default skeleton directory path to a location other than /usr/skel.

   g. Use Hashed Password Database – This option forces the creation of a hashed (encrypted) password database.

   h. Require Password For New Accounts – This option forces the entry of a password each time an account is created.

   i. Synchronize UNIX and Windows NT domain accounts – This option forces the automatic creation of an account when the UNIX account is created.

2. When you have made any required changes, press 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 with the SysMan Menu options. This method also allows you to add groups in a NIS (Network Information Service) environment.
- Administer groups using `dxaccounts`.

The processes for using the `groupadd`, `groupmod`, and `groupdel` commands are similar and are documented in the reference pages. Note that the SysMan Menu can also be used from a terminal.

Note

Avoid using `addgroup` as it does not provide all the available options and is not sensitive to security settings. Avoid using manual methods of adding user accounts to preserve system file integrity.

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 or NIS group
- Modifying an existing local or NIS group
- Deleting a local 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. Note that if enhanced security is in use, you must make the data items comply with the minimum requirements. Refer to the *Security* guide for more information.

Refer to Section 7.3.3 for an explanation of the `group` file data items. In the SysMan Menu options, you also have the option to specify values for

NIS groups. Refer to the *Network Administration* guide for information on configuring NIS.

Items marked O are optional during group creation.

Table 7–4: Group Administration Worksheet

Data Item	O	Note	New Account
Group Name			
Password		Not currently used.	*
Group Identifier (GID)	O	If unused, the next number will be assigned.	
User	O		

7.5.1.2 Creating or Modifying Groups

To create a new group, invoke the SysMan Menu and select the Manage local groups option as described in Section 7.2.2. The Local Groups table is displayed, listing all the existing local groups. The process for adding NIS groups is identical, except that you select the `Manage NIS Groups` option.

Use the following procedure to add a group:

1. Select the Add... option to display the Manage local groups: Add a Group window.
2. Complete the data fields using the information from the worksheet described in Table 7–4.
3. From the Members panel, highlight the names of users who will be the initial members of the new group. This action is optional.
4. Press OK to add the new user. You will be prompted to correct any errors.
5. You return to the Local Groups table window. A message confirming the successful addition is displayed. Press OK to return to the SysMan Menu.

To modify an existing group, invoke the SysMan Menu and select the Manage local groups option as described in Section 7.2.2. 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 select an entry.

2. Select the Modify... option to display 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. Press OK to confirm the changes.

   To add or modify more than one group, click Apply instead of OK. All changes are deferred until you select OK to exit.
5. You return to the Local Groups window. A message confirming the successful addition is displayed. Press OK to return to the SysMan Menu.

Online help provides explanations for the fields, and defines valid data.

7.5.2 Using the Account Manager

Invoke the `dxaccounts` utility as described in Section 7.2.3. The Account Manager on <host> window is displayed first. Using the data gathered in the Table 7–4 worksheet, 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.

7.5.2.1 Adding Groups

Add groups as follows:

1. Pull down the View menu, and choose the Local Groups option.
2. Choose the Add button to display the Add/Modify Local UNIX group window.
3. Enter the new group name in the Name field.
4. You can opt to choose the next available GID or enter a GID.
5. Double click on any user name to add that user to the group.
6. Press 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 creating a new group is to clone it from an existing group as follows:

1. Click on an existing group icon to highlight it.
2. Choose the Copy button to copy the group.

3. Choose the Paste button to paste a new group version. The new icon label will have 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. Click on the newly copied icon to highlight it.
5. Press the Add button to display the Add/Modify Local UNIX group window.
6. Make any required modifications to the group as follows. For example:
   - Rename the group
   - Change the GID
   - Add or delete members.
7. Press 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.

7.5.2.2 Modifying Groups

Invoke the `dxaccounts` utility as described in Section 7.2.3. The Account Manager on <host> window is displayed first. Use the following procedure to create a new group, using the data gathered in the Table 7–4 worksheet:

1. Double-click on the required group to display the Add/Modify Local UNIX group window.
2. Make any required modifications to the group as follows. For example:
   - Rename the group
   - Change the GID
   - Add or delete members.
3. Press OK to confirm the changes and return to the Account Manager on <host> window. This window is immediately updated with any name changes for the group.

7.5.2.3 Deleting Groups

Invoke the `dxaccounts` utility as described in Section 7.2.3. The Account Manager on <host> window is displayed first.

1. Double-click on the required group to highlight it.
2. Press the delete button. You will be prompted to ensure that you want to delete this group.

3. Press yes to confirm the deletion and return to the Account Manager on <host> window. This window is immediately updated, removing the deleted group.

7.5.2.4 Finding Groups

The Account Manager provides a useful search feature to locate groups and users who are members of groups.

Invoke the `dxaccounts` utility as described in Section 7.2.3. The Account Manager on <host> window is displayed first.

1. Press the Find button.
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 group identifier (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 `wallyB` and `cadwalZ`.

7.6 Administering Windows NT 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 NT domain accounts. For information on installing and configuring ASU, refer to the *Installation and Administration Guide*. Note that in such environments, 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 NT domain and the UNIX environment and are referred to as synchronized accounts in the UNIX utilities.

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 defaults using the `usermod` (or `useradd`) command at a terminal as shown in Example 7–1.

Note

When the Advanced Server for UNIX (ASU) is installed and configured, the creation of associated Windows NT domain and UNIX accounts is enabled by default. All account management utilities will have their PC support featuers enabled automatically. The value of `Synchronized UNIX/PC Accts` in the system default settings will be `=1` (on).

Example 7–1: Changing the Default Environment with 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
.
.
.
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 user account creation environment.

Example 7–1: Changing the Default Environment with usermod (cont.)

---

2 The output from the `usermod` command is this list of default values. 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 values for three of the defaults that apply to Windows NT domain accounts only.

4 This (truncated) list shows the new default values, which are as follows:

- pc_synchronize=1 – Create 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.

A similar command, `groupmod -D` is used to set the default environment values for creating new groups. You can also specify the environment options when you create a new account, to override the defaults. For more information, refer to the `useradd`(8), `usermod`(8), `userdel`(8) reference pages. At the command line prompt, you can also type `-h` after each command to display 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 values for newly created accounts.

It is not possible to use ASU account management utilities to perform operations on UNIX-only accounts, or to 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 will become 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 could easily 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 will inform you that the data files are locked.

When using the ASU utilities to add accounts, ASU is able to detect the presence of the lock file, but will be unable to create an associated UNIX account. It will only create a Windows NT domain account. No lock file error message will be displayed, and you will receive no confirmation that the associated account was not created. When using ASU tools, you should always verify the creation of an associated UNIX account by checking the contents of the `/etc/passwd` file.

7.6.1.1 Using SysMan Menu Accounts and Groups Options

The SysMan Menu Accounts utilities will not show any changes when ASU is running. If synchronized accounts are enabled, you will not see any differences in the windows and screens. However the following behavior should be noted:

- Add a user – You will be able to select 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 will be created automatically and you cannot override its creation.

- Delete a user – The associated Windows NT domain account will be deleted automatically. You do not have an option to override this deletion. If you want to retain the users' Windows NT domain account, do not use this option.

- Add/Modify a group – several `DOS----` groups will appear in the list, showing the default Windows NT domain accounts, such as `lanman` and `lmxadmin`.

The advantage of the SysMan Menu utilities is that you can use them in a number of different user environments; they present a consistent method of account administration no matter whether you are working in terminal, Microsoft Windows, or X windows. Refer to Chapter 1 for information on using the SysMan Menu.

7.6.1.2 Using the Account Manager

The Account Manager (dxaccounts) is an X11–compliant graphical user interface (GUI) and as such can only be displayed in an X-window user environment such as CDE. The Account Manager shows an option to create **PC** (Windows NT domain) accounts on the main window. This option is grayed out and unusable unless ASU is running. When ASU is running, the following features are available:

- When creating one type of account in Windows NT domain or UNIX, you can opt to create a synchronized account of the other type.
- You can opt not to create an associated PC (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, that enable you to display all Windows NT domain accounts and groups. When you select 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 will be 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 the Delete button, you are prompted to select the UNIX account, the PC account, or both.
- When using the View menu, Local Groups option, the PC `DOS` groups 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 using the processes described in Section 7.5.2 to perform administrative operations on PC accounts and groups.

The advantage of using the Account Manager is that it is a native X11 application and can use the features of the windowing environment such as iconic drag-and-drop or cut-and-paste to easily clone new user accounts and groups from existing entities. However, unlike the portable SysMan Menu Account utilities, it can run only under an X-window user environment.

7.6.1.3 Using Command-Line Utilities

The command-line utilities for administering user and group accounts are also used to configure the default account characteristics, as demonstrated 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, you can always override the default characteristics and specify customized characteristics 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 UNIX. 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 checking 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 will be 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

- pc_unix_username=*login_name*

  The synchronized UNIX account name. If no name is entered, it will be 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 is modifiable 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 will expire and logins will be prevented.

- `pc_logon_hours=`*`Dd0000-0000,Dd0000-0000....`*

  Specifies the days of the week and hours of the day during which logins will expire and logins will be 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 initially locked, disabling logins.

- `pc_passwd`*`0|1`*

  A text string that will be the initial account password. Note that you must precede this option with the `-x` flag and you will be prompted to enter a password, and then confirm the entry. The password will 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 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 default characteristics.

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.

Refer to the `useradd`(8) and `groupadd`(8) reference pages, 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 (`usrmgr.exe`) must be installed and can only be used from a system running Windows NT. It offers the same features as the `net` command line options.

Default characteristics for accounts, called **policies** in the context of this utility, can be set for all newly created accounts. However, you cannot set the default characteristics for synchronized UNIX accounts when using the User Manager for Domains.

Refer to the *Installation and Administration Guide*, and the User Manager for Domains online help for 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
```

Type `net help view` at the command line to display a list of `net` command options. See also the *Installation and Administration Guide* and the `net`(1) reference page for information on using `net` commands.

8

Administering the Print Services

This chapter describes how to set up and administer the files and utilities that make up 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. For example, you can wait until you have installed a printer and have gathered the information about its characteristics that you need to set it up.

During initial configuration of your system after a full installation, you will see a checklist titled *System Setup*. On this menu is an option for Printer Configuration, which runs the `printconfig` graphical user interface. Refer to Section 8.2.2 for information.

The following topics are discussed in this chapter:

- Section 8.1 provides an overview of the administrative tasks, describes the different configuration methods and the set up utilities that you can use. It provides pointers to other documentation that you may need to read. Also listed here are the various system files that are used by print operations and any utilities that are related to print operations.
- Section 8.2 describes how to use different utilities to set up local, remote, and networked printers. It describes the `printconfig` and `lprsetup` utilities. Also included is information on Advanced Server for UNIX (ASU) print facilities that support client PCs and the Advanced Printing Software.
- Section 8.3 describes routine print system maintenance, such as adding and removing printers or controlling print jobs.
- Section 8.4 provides reference information on advanced topics such as the structure of system files, spooling, daemons, error reporting, and and print filters.
- Section 8.5 describes current restrictions on the use of certain print filters.
- Section 8.6 provides information that enables you to test printers and resolve problems.

8.1 Print Administrative Tasks

To set up the print system, you perform the following tasks:

- Physically connect a printer to the system or ensure that you have access to it through a network
- Add information about a printer in the `/etc/printcap` file
- Create the required device files and spooling directories
- Start the print (`lpd`) daemon
- Manage printer accounting
- Verify printer installation and perform a test printing

After a printer is set up and running on your system, you need to:

- Manage the system and take care of routine changes such as adding new printers or changing the characteristics of existing printers
- Administer the print queues and files as your system needs change
- Control the daily operations and throughput of print jobs

The tools that you use to perform these operations are described in Section 8.1.5.

8.1.1 Advanced Printing Software

The Advanced Printing Software is an optional subset on the Associated Products CD-ROM. For information about this software, see the Advanced Printing Software *Installation Guide* and refer to the *Advanced Printing Software System Administration and Operation Guide* and the *User Guide*. Note that when the Advanced Printing Services software is installed, you must configure a gateway, as described in Section 8.2.2. to run Advanced Printing Services and the print daemon `lpd`on the same system. You can run Advanced Printing Services with `lpd` disabled, in which case it will receive all inbound remote print requests (on socket 515) instead of LPD. However, with this configuration, local `lpd` commands such as `lpr` will not work.

8.1.2 Printer Configuration Methods

Depending on your local system configuration, you have several options for installing and configuring printers, For example:

- Direct – Local installation of a single printer is the simplest installation. At the rear of the system are serial or parallel hardware ports to which you connect a printer with a cable. (The printer documentation describes the hardware installation.) This printer can be used by a single user, if connected directly to a single-user workstation, or it can be configured to be shared with other users if it is connected to a multiuser system or server. Note that connections using USB ports are not yet supported.

- Remote – Connection to a printer that is 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.
- Network – Connection to shared a network printer across a local area network (LAN) or a local area transport (LAT) connection or via TCP/IP. For example, a desktop workstation may be able to print to several different network printers.
- PC Network — Connection to Personal Computer (PC) print queues when using the Advanced Server for UNIX (ASU). This application is used to manage mixed IBM-compatible PC and UNIX printer environments. When ASU is installed, you have additional options to configure PC print queues and share printers between PC clients.

There are also several different ways to perform printer administration tasks, each of which provides certain advantages:

- The `printconfig` graphical user interface, which is recommended for first-time users and for quick setup.

  The `printconfig` utility is part of the standard set of system administrative tools. Refer to 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. When you use `printconfig`, it will automatically display a list of all the supported printers and enable you to configure them quickly.

  Note

  There are restrictions on using `printconfig` with older `/etc/printcap` files. Refer to the `printcap`(4) reference page for information.

- The `lprsetup` utility. This utility provides backwards compatability with previous releases.

  The `lprsetup` utility is a command-line utility that you run from a terminal. It performs the same tasks as `printconfig`, but does not support PC printers even if ASU is installed. Note that if you select the command-line option when invoking `printconfig`, the `lprsetup` utility will be invoked.

  Note

  Currently, `lprsetup` is the command-line utility for print configuration. However, `printconfig` is the preferred method for administering printers. 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.

- Manual edits to system files

  Experienced system administrators may want to manage printers by editing the system files. For example, if you are working on printer driver development, it may be necessary to create test entries in the printer configuration file `/etc/lprsetup.dat`. Alternatively, you may want to clone a particular printer configuration across a number of systems by editing the `/etc/printcap` file to include several configured printers, and copying that file to the destination systems. The reference information in this chapter will assist you in performing such tasks or in writing shell scripts. However, note that In this release, the `/var/spool/lpd` file is a special link to support clusters. 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 Related Documentation

The following documentation contains information on using printer configuration tools.

- Books:
  - Consult the printer 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 will provide information that you may need to provide to the configuration utility to use any special capabilities of the printer, such as tray selection.

    Usually, you will see your printer included in the list of supported devices when you use an installation utility (or if you look in `/etc/lprsetup.dat`). If it is not listed there, and the manufacturer does not provide information for using the printer under UNIX, you may be able to use it with generic settings provided in the configuration tools. However, you will probably not be able to use any special capabilities of the printer, and will most likely be limited to basic features.
  - The *Network Administration* guide provides information on configuring or connecting to networked printers.
  - The *Writing Software for the International Market* guide 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* provide information on setting up printer services in the Common Desktop Environment.

– The *Installation and Administration Guide* describes ASU features for managing PC print queues.

- Reference pages:

  – `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 that enables you to support local-language PostScript printing. The `pcfof`(8) reference page describes a generic print filter for ANSI, PCL 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.

  – `services`(4) – Describes the format of the `/etc/services` file where services defined for TCP/IP printing are defined.

- Online Help – The `printconfig` graphical user interface has an online help volume that explains how to use the interface and the `lprsetup` utility has command-line help.

8.1.4 System Files

The following system files contain printer configuration information. Note that some files, such as `/usr/spool`, are defaults or UNIX conventions. You can use your own preferred file names and locations.

- The `/etc/printcap` file contains the data on configured printers.

- The /etc/lprsetup.dat file contains the configuration data for all supported (known) printers. This information is transferred to the /etc/printcap file when a printer is installed and configured for use.
- The /usr/lbin directory contains the print filters for each known printer device. These filters are specified during printer installation, for example /usr/lbin/ln09of is the filter for the DEClaser 5100. Filter file names contain the following data:
  - The printer identifier (ID) string
  - The /etc/printcap code of
  - A string to determine any variant information, such as _isolatin1
- The /usr/spool 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 the /etc/printcap
- The /usr/spool/lpd directory is the CDSL in which lpd.lock is stored for each member of a cluster. On a cluster, spool directories must be shared, and are always located under /usr/spool.
- The /usr/adm/lpd*err files is the default directory where error log files for each installed printer. These are only created if error logging is enabled.
- 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.
- The /usr/lbin/lpd file is the line printer daemon. Configuration files are located in the /usr/spool/* (or /var/spool/*) directory.
- 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.

8.1.5 Related Utilities

The following utilities are also available for use in your printer environment:

- The lpc line printer control utility enables you to manage print queues and control access to printers. See the lpc(8) reference page for information.
- The pac utility formats the data from printer accounting log files and displays it or stores it in a text file. See the pac(8) reference page. See also the acct(8) reference page for information on accounting.
- CDE Application Manager — Desktop Apps contains the Print Manager graphical user interface, which enables you to perform the following

tasks. Refer to the online help for information on how to use these graphical interfaces.

– Manage print queues
– Control access to printers
– Customize your view of 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.

The Print Screen utility is also located in the Desktop Apps, if this utility has been installed.

- The printer icon on the CDE front panel allows you to select printers and manage print queues locally. You can also run `printconfig` from CDE Application Manager – Configuration, in addition to invoking it from SysMan Menu or the SysMan Station. The latter user environments you to use graphical tools remotely or from a different workstation such as a PC or another UNIX system. Consult the CDE documentation for information on setting environment variables such as `LPDEST` to assign system default printers in CDE environments.

8.2 Configuring Printers

The following sections describe the information you need in order to use the `printconfig` utility to connect a printer to your computer. Before proceeding, verify that the printer is physically connected to your system, accessible on the network (for remote printing), and has been tested 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. Once 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:

```
OSFPRINT500     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.

8.2.1 Gathering Information

Before adding a printer, you need to gather the information about the printer that you will need to enter when using the `lprsetup` or `printconfig` utilities. The information required will depend on whether the printer is remote, a direct connection, or a network connection via LAT or TCP/IP:

If your system is part of a network, you may need to consult your local network administrator or consult the *Network Administration* guide for information required when adding or accessing a network printer.

8.2.1.1 Direct and Network Printers

The following list identifies the information you need when installing a direct connection, or a network connection via LAT or TCP/IP.

- Available Printer Types (supported printers) — This can be determined from the `lprsetup.dat`(8) reference page or by viewing the `/etc/lprsetup.dat` file 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 will depend on how the printer is connected to your system. It can be:
  - 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* guide for more information.)
  - TCP – A networked print server device.
- Device Pathname — This depends on the Connection type:
  - If the connection type is direct, 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`.
  - If the connection type is LAT, `printconfig` will supply a default `/dev/lat` port or service name, LAT server node name, or LAT port name. (See the *Network Administration* guide for more information.)
  - If the connection type is TCP, you need the TCP address, which can be in one of two formats. The owner's manual of the printer will specify the assigned port number for the printer device (a numerical

string, such as 12345). The other part of the address is the host name or TCP/IP address:

- ☐ `@node/port` — The printer host (or node) name followed by either the port number such as 2501 for an LN17ps, 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/ln17port` maps to the entry `ln17port 2501/tcp` in `/etc/services`. The entry `@alfie.nic.ccc.com/2501` 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 2501 for an LN17ps) or the service name defined in `/etc/services`. For example `@123.321.123.321/2501`.

- 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 file `/etc/printcap` and the `printcap`(4) reference page contains a definitive list of supported symbols and values. When using `printconfig`, the online help provides a description of the symbols.

  Note that 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 — If you want to use printer accounting to track print consumables, you should choose the default (or enter your preference).

  – stty baud rate (hard-wired ports only) — If your printer specifies communication rate requirements you can enter it here. Note that the default rate can sometimes be increased to improve printer throughput.

  – Restrictions on use — If you want to control the volume and quantity of print jobs, set the restrictions here.

  – Default page layouts — If you want default vaues for certain page layout characteristics, set the characteristics here. Supported layouts may depend on printer restrictions and capabilities. Consult the owner's manual for the printer.

- Destination directories and files — If you want certain directory and file locations for print spooling or for error output, you can specify the locations here.

8.2.1.2 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 machine 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 only a small number of advanced options to be configured (or deconfigured, such as error logging) . See the information in Section 8.2.1.1. Note that most advanced options are not passed on to the remote system.

8.2.1.3 Printer Data Worksheet

You can photocopy the following table and use it as a worksheet.

Name / Format	Example/ Location
Printer Type (Alphanumeric string)	`ln09 (lprsetup.dat)`
Alias (Alphanumeric string)	alfie2 (User-supplied)
Connection type (software option)	TCP (Interface-supplied)
Device pathname, direct connection) (file name)	`/dev/tty00 (ports(8))`
Device pathname (Remote Connection) (port or service name)	`556 (lat reference page)`
Device pathname, network connection (host/service name)	`@alfie.nic.ccc.com/2501 (Printer Documentation, /etc/services file)`
Remote machine name (host name)	`alfabet.ccc.nic.com` (Network Administrator)
Remote printer name (printer name)	`lp0 (remote host printer configuration)`

8.2.2 Installing a Local Printer

This section describes how to install a printer using `printconfig` utility. The example given is a DEClaser 5100 printer installed locally using the graphical interface. It assumes that you have made all the physical connections and gathered the required information. You can also use printconfig to modify a printer configuration or remove a printer. These other tasks are described in Section 8.3.

Note

Do not use printconfig if you are modifying an /etc/printcap file on a system running Version 3.2 or older. There are incompatabilities 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 so that the basic installation and verification can be accomplished quickly. You can then use printconfig to modify the configuration later if required.

Invoke printconfig from CDE, or from the command line, as described in the printconfig(8) reference page.

You must have superuser privileges to run printconfig utility. Depending on the type of printer you are adding and the information you provide, the utility might 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 you have already gathered as described in Section 8.2.1:

- Select ln09
- 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 will be 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 will choose the Direct option.
- Finally, you need to specify the Device pathname, which will be 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 8.4.

After commiting 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. See Section 8.6.

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. This means that any print job that has no specified print queue will default to this printer. Generally, you would choose a high-speed low-cost draft printer as the default on a multiuser system, in case users choose to print large text files.

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 capabilites with appropriate files, such as PostScript or color graphics files. Check the printer utilities described in Section 8.1.5 to ensure that you can verify printer and queue status.

8.2.3 Setting Up Remote Printers

A remote printer refers to a printer that is already directly connected to a remote host. Remote printers may also be directly connected to the network, but 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 via the network. These jobs are then printed on the remote host. If you are setting up a remote printer from a remote machine, the local machine (the client) must be listed in the hosts.lpd file or hosts.equiv file of the remote machine (the host).

If your printer will be connected to a remote LAT terminal server, ensure that the LAT subsets are installed as described in the *Installation Guide*. To see if the LAT subsets are installed, enter:

```
# setld -i | grep OSFLAT
```

See the *Network Administration* guide for information on how to enable remote LAT terminal server printing.

Refer to section Section 8.2.1 for information on the data that you need to gather before performing this task, then invoke printconfig as described in Section 8.2.2 (the remote queue is the same queue as the example created in that section). The following example describes how to use printconfig to create some remote print configurations:

- Select remote
- Choose the Configure option

Because this is a remote printer, the next window is Printer Config: Remote Printer Settings. The next available printer name will be displayed (lp0 if this is the first printer that you are configuring on this system). Enter the following:

- The printer alias names, such as the name of the remote host and printer type, such as alphie_ln09.
- The remote machine name. For example, in Section 8.2.2 a queue was created on the system named alphie.ccc.nic.com.
- The remote printer name. For example in Section 8.2.2, a printer lp0 was added to the system.
- 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. Because the printer is remotely configured, you can only specify a small number of advanced options that affect local processing, such as the local error log file and spooling directory.

After commiting 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.

8.2.4 Setting Up TCP/IP Printing Using printconfig

TCP/IP printing allows you to submit print jobs to a remote printer that is directly connected to the network as a host device. Reference information on TCP/IP printing is located in Section 8.4.11. Gather the information as described in Section 8.2.

Invoke `printconfig` as described in the preceding sections. The first display is the main window titled Printer Configuration on *host name*. In this window you use the data you have gathered to select the printer type, for example:

- Select `ln17ps`
- 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 will be displayed, such as `lp4`. Complete the fields as follows:

- The printer alias names, such as the name of the local host or perhaps something to help users to identify the printer type and physical location, such as `ln17_office23`.

  This value can also be specified as `@`*`nodename`*`/`*`servicename`*, where *`servicename`* is defined in `/etc/services` and associated with the printer's TCP/IP port.
- The connection type, this will be TCP.
- Finally, you need to specify the device pathname as a port number or service, which in this case will be the network address of the printer, such as `@alfie.nic.ccc.com/2501` or `@123.321.123.321/2501`.
- 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 commiting 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.

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. This means that any print job that has no specified print queue will default to this printer. Generally, you would choose a high-speed low-cost draft printer as the default on a multiuser system, in case users choose to print large text files.

Select the required printer and choose Make Default. The current default printer is displayed in the field labeled Default Printer.

8.2.5 PC Print Queues Under ASU

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. Refer to the *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` from the CDE Application Manager — Configuration window, or from the command line, as described in the `printconfig`(8) reference page. 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.

Press OK to create the queue and return to the `printconfig` main window. The new queue will be displayed.

To test the status of the queue, use the following ASU command and check the output for the queue name as shown in the example:

```
# net share
.
.
.
Share name  Resource            Remark
----------  ------------------  ------------------------
psq1        lp0, lp2            Spooled
```

Note that you may need to perform other ASU tasks to make the queue available to PC systems. Consult the ASU doumentation for more information.

8.2.6 Using lprsetup to Install a Printer

This section describes how to install a printer locally (directly connected to your computer) using the lprsetup utility. You can also use lprsetup to modify a printer's configuration or remove a printer. These other tasks are described in Section 8.3.

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
Compaq Tru64 UNIX Printer Setup Program

Command < add modify delete exit view quit help >:
```

The lprsetup command options are described in Table 8–1.

Table 8–1: 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 can also enter a question mark (?) to get a description of the information you specify at the prompt.

Note

Some of the symbols displayed in the `lprsetup` script are not supported by the operating system. Refer to the `printcap`(4) reference page for information on the supported symbols.

The following example shows how to use the `lprsetup` command to set up an LN03R printer to be used by the local system. Note that some tables have been truncated to shorten the example:

```
# /usr/sbin/lprsetup
Compaq 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 [1] : Return

For more information on the specific printer types
enter 'printer?'

Enter the FULL name of one of the following printer types:

DEClaser1100 (ln07)   escp_a4_12cpi   hp6l_a4  la400    lg12      ln17_a4
DEClaser1150 (ln07r)  fx1050          hpIIID   la424    lg12plus  ln17ps
DEClaser1152_I18N     fx80            hpIIIP   la50     lg14plus  ln17ps_a4
DEClaser2100 (ln05)   generic_ansi    hpIIP    la600    lg31      ln20
DEClaser2150 (ln05r)  generic_ansi_a4 hpIV     la70     lgl04plus ln20_a4
DEClaser2200 (ln06)   generic_text    ibmpro   la75     lgl05plus ln40
DEClaser2250 (ln06r)  generic_text_a4 la100    la84     lgl08plus ln40_a4
DEClaser2300 (ln05ja) hp1120c         la120    la86     lgl09plus ln82r
DEClaser2400 (ln10ja) hp1120c_a4      la210    la88     lj250     lnc02
DEClaser3200 (ln08)   hp4000tn        la280    la88c    lj252     lnc02_a4
DEClaser3250 (ln08r)  hp4000tn_a4     la30n    la90     ln03      nec290
DEClaser3500 (ln14)   hp4mplus        la30n_a4 lf01r    ln03ja    ps_level1
DEClaser5100 (ln09)   hp4mplus_a4     la30w    lg02     ln03r     ps_level2
DEClaser5100_I18N     hp5simx         la30w_a4 lg04plus ln03s     remote
cp382d                hp5simx_a4      la324    lg05plus ln15      unknown
dl510ka               hp680c          la380    lg06     ln15_a4   wwpsof
ep1050+               hp680c_a4       la380cb  lg08plus ln17      xf
escp_a4_10cpi         hp6l            la380k   lg09plus

or press RETURN for [unknown] : ln03r
Enter printer synonym: tomf
Enter printer synonym: Return
Set device pathname 'lp' [] ? /dev/tty01
Do you want to capture print job accounting data ([y]|n)? y
Set accounting file 'af' [/usr/adm/lpd/lp1acct]? Return
Set spooler directory 'sd' [/usr/spool/lpd/lpd1] ? Return
Set printer error log file 'lf' [/usr/adm/lpd/lp1err] ? Return
Set printer connection type 'ct' [dev] ? Return
Set printer baud rate 'br' [4800] ? 9600
```

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. Refer to the printcap(4) reference page for more information on the various symbols.

```
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 of the possible symbols and defaults
The names of the printcap symbols are:

 af  br  cf  ct  df  dn  du  fc  ff  fo  fs  gf  ic  if  lf  lo
 lp  mc  mx  nc  nf  of  op  os  pl  pp  ps  pw  px  py  rf  rm
 rp  rs  rw  sb  sc  sd  sf  sh  st  tf  tr  ts  uv  vf  xc  xf
 xs  fo  ic  nc  ps  Da  Dl  It  Lf  Lu  Ml  Nu  Or  Ot  Ps Sd
 Si  Ss  Ul  Xf

Enter symbol name: q

        Printer #1
        ----------
Symbol  type  value
------  ----  -----
  af    STR    /usr/adm/lpd/lp1acct
  br    INT    9600
  ct    STR    dev
  fc    INT    0177777
  fs    INT    03
  if    STR    /usr/lbin/ln03rof
  lf    STR    /usr/adm/lpd/lp1err
  lp    STR    /dev/tty01
  mc    INT    20
  mx    INT    0
  of    STR    /usr/lbin/ln03rof
  pl    INT    66
  pw    INT    80
  rw    BOOL   on
  sd    STR    /usr/spool/lpd/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 at the number sign (#) prompt to exit. The comments will be insert 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
# Use this printer for draft-only
# Return

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
```

Refer to the lprsetup(8) reference page for more information on options.

8.2.6.1 Print Symbols for Advanced Printing Services

When setting up Advanced Printing Software, you should set the following print symbols:

- rm – Specify @dpa to indicate that jobs submitted to this printer should be 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.

Refer to the *Advanced Printing* administration documentation for more information.

8.3 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 already 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
- Modifying characteristics of existing printers
- Removing printers from the system
- Enabling printer accounting
- Controlling printer operations by using the CDE tools or the lpc command

Note that if you manually remove printers from the /etc/printcap file, you also have to manually remove spooling, accounting, and error directories and files.

8.3.1 Adding Printers

Once you have one printer set up, you can add other local, remote and networked printers at any time. Gather the information about each printer as described in Section 8.2

You can add printers by running `printconfig`, or you can add printers manually by performing the following steps:

1. If it does not already exist, create a printer spooling directory. Refer to Section 8.4.2.6.2.
2. Modify the `/etc/printcap` file and edit it to include a description of the printer using the configuration data from `/etc/lprsetup.dat`. Refer to Section 8.4.1.
3. Create an accounting file and a log file and enable printer accounting. Refer to Section 8.3.5. Protection and ownership of this file must be set 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.

8.3.2 Modifying Printers

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 manually modify a printer's configuration, edit the `/etc/printcap` file and modify the printer entry. Refer to Section 8.4.1 and to the `printcap`(4) reference page for information about the `/etc/printcap` file symbols.

8.3.3 Removing Printers

To remove a printer, run the `printconfig` utility and choose the printer that you want to delete, then choose Deconfigure. You are then prompted for confirmation that you want to delete the error log file and the accounting file because these files may be shared by more than one printer. If you do have shared files, do not delete them.

If you have included comments for the printer in the first line of its `/etc/printcap` file entry, note that the command line `lprsetup` utility does not delete the comments when you remove a printer. You can edit the `/etc/printcap` file and delete the comments.

To manually remove a printer, edit the `/etc/printcap` file and delete the entry that relates to the printer. You must also manually delete the accounting and log file and the spooling directory if no longer required.

You can also use `lpc` and the CDE print mangement tools to temporarily control access to printers and queues. See Section 8.3.4.

8.3.4 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 can also manage print jobs using the Print Manager located in the CDE Application Manager – Desktop Apps folder. Refer to the online help for information on how to use these graphical 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

Some `lpc` commands, for example, the `disable` command, require you to be superuser.

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 15 command arguments that you can specify with the `lpc` command. You can also use the `lpc` command interactively. If you enter the `lpc` command without any command arguments, the `lpc>` prompt is displayed. You can then 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.

You can specify the *argument* variables defined in Table 8–2.

Table 8–2: 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.
clean	Removes any temporary files, data files, and control files that cannot be printed (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. The message does not need to be quoted 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.
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 first kill the process and 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.

Table 8–2: lpc Command Arguments (cont.)

lpc Argument	Description
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>
```

Refer to the lpc(8) reference page for more information.

8.3.5 Enabling Printer Accounting

Printer accounting allows you to charge users for printing services and to determine the amount of printer usage.

There are two types of printer accounting: printer user accounting and printer summary accounting. Printer user accounting provides information about printer use according to the machine and user name that issues the print request. 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. If you use lprsetup to add a printer, it 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

The /var/adm/lpd directory should be owned by user adm and belong to group adm. The printer accounting files should have protection mode 644, be owned by user adm, and belong to group system.

In the printer's /etc/printcap entry, the af parameter specifies the name of the accounting file used to keep track of the number of pages printed by each user for each printer. The name of the accounting file should be unique for each printer on your system. Use the pac utility to summarize information stored in the printer accounting files. This file must be owned by user daemon and group daemon, which it will be if you use the printconfig utility to specify the printer accounting file.

The af parameter is not applicable for remote printer entries because the accounting policy for remote printers is employed at their (remote) systems.

Accounting is accomplished through programs called print filters. The printconfig utility will suggest a default print filter. Two print filter symbols, if and of, are needed for accounting. For example:

```
if=/usr/lbin/ln03rof
of=/usr/lbin/ln03rof
```

If you want to use separate accounting files for each printer on your system, the file names must be unique. However, an unlimited number of printers can share an accounting file. You cannot specify an accounting file for remote printers.

Accounting files must be owned by the print daemon. If you specify an accounting file, intermediate directories are automatically created as needed.

Note

Printer accounting does not work for PostScript files.

8.4 Reference Information

The following sections contain information about the data required for printer configuration, line printer daemon, lpd, and the system files that are required for print system operations. These files are created automatically if you use printconfig, as described in Section 8.2.2, or you can create and modify the files manually. Note that if you create files manually, you will also need to manually change the /etc/printcap file so the changes can take effect.

8.4.1 The /etc/printcap File

The lpd daemon uses the /etc/printcap printer database file to print requests. Each entry in the file describes a printer. Printer characteristics are specified by two-letter abbreviations called print symbols. The print symbols are described in this section and in the printcap(4) reference page. The lprsetup utility modifies the /etc/printcap file.

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:\
       :sb:\    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, but each field on the first line must be separated with 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 can be specified with 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.

Theprintcap(4) reference page lists the printcap symbol names, the type of values they accept, default values, and descriptions of the symbols.

8.4.2 Data in /etc/printcap

The following information is typically required for a printer entry in the /etc/printcap file.

8.4.2.1 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 Return 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 will be 1 and lp1. Jobs would then be queued to this printer using either of the the following commands:

```
# lpr -P1
# lpr -Plp1
```

You can also assign your own synonyms and use them to direct jobs to printers.

If you have only one printer or are entering the first of many printer names, the first name will have a printer number of 0. This is recognized as your system's default printer and will have an additional name of `lp`. This means if you use the `lpr` command without specifying a specific printer this is the printer that will always be used.

If this is the first printer connected to your system or a new printer added to an existing print system, create names that do not conflict with existing printer names. Ask your network administrator for the names of the remote printers on the network.

8.4.2.2 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 and only those supported by Compaq are listed. These printers have some default values already included in the setup utility.

The supported printer types are defined in `/etc/lprsetup.dat` and described in the `lprsetup.dat`(4) reference page.

You can set up unlisted printers by using `unknown` and then 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 `unknown`.

To install third-party printers, consult the documentation that came with the printer.

8.4.2.3 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 Return 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
```

8.4.2.4 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 directly connected 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/lp
```

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: `lp1`, `lp2`, `lp3`), 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) as shown in Table 8–3.

Table 8–3: 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

Note that if only one 9–pin serial connector is provided on a system, it may not be labeled as such. Some systems also use graphical icons instead of labels. Refer to the owners manual for the system for more information.

When you use `lprsetup`, the utility defaults to the next consecutive number when it sets up this file. For example, the default device pathname for the third serial line printer is `/dev/tty03`.

The default device special file is `/dev/lp`, which specifies a parallel printer.

For remote printers, specify a null argument with the `lp` symbol, or a node and port for TCP/IP, as in the following examples:

```
lp=
lp=@<node/port>
```

Note

If the port is used for logins, the `lprsetup` script turns off the terminal line established by the `getty` process so the terminal line can be used for a printer.

8.4.2.5 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 (must be specified in lowercase)
- `tcp` for TCP/IP devices (must be specified in lowercase)

8.4.2.6 Spooling Directories

In `/etc/printcap`, the `sd` parameter specifies the spooling directory where files are queued before they are printed. Each spooling directory should be 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 must be located on the machine 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 could be `/usr/spool/lpd2`. The default spooling directory for any printer is `/usr/spool/`.

Each printer entry in the `/etc/printcap` file should specify a spooling directory even if the printer is connected to another machine or is on another network. You specify a spooling directory with the `sd` symbol. For example:

```
sd=/usr/spool/lpd2/purple
```

Spooling directories must have the same parent directory name, which is normally /usr/spool/lpd. You can specify alternative locations if required when configuring printers.

8.4.2.6.1 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 /var/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/purple/lock.

The lock file prevents the lpd daemon from invoking another job on the printer while a file is printing. The lock file contains the process identification number of the daemon that is currently running. 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:

- The file .no_daemon is created when the queue has entries and no daemon is running and no files in the spooling directory have been removed or changed in the past ten seconds.
- The file .daemon_running is created by a running daemon before it has done its first check for jobs to print. As long as this file exists, the currently running daemon will do at least one more check for queued jobs before deciding there are no more jobs to print.

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 user files to be printed and contain printing instructions for the 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 the lpd(8) reference page.

Data files, whose names begin with df, are also located in the spooling directory. Data files contain text formated for printing. These files are identified by their print request identification numbers only.

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 a printer named milhaus has jobs currently waiting to be printed, the following command lists the files that are stored in the spooling directory:

```
# ls -l /var/spool/lpd/milhaus
-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
```

8.4.2.6.2 Creating a Spooling Directory

If you want to manually add a printer, use the mkdir command to create the spooling directories for each printer. The spooling directory permission mode must be set to 775. The directory's group and ownership must be set to the name daemon. For example:

```
# cd /var/spool/lpd
# mkdir lp1
# chmod 775 lp1
# chgrp daemon lp1
# chown daemon lp1
# ls -l lp1
drwxr-xr-x  2  daemon  daemon  24  Jan 12  1994  lp1
```

8.4.2.7 Baud Rate

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.

You specify a baud rate only for serial printers that are local (directly connected to your computer). Baud rates are not specified for printers connected to the console port or connected by a parallel port or LAT port.

8.4.3 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 to be printed 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 is usually started by the `/sbin/init.d/lpd` utility. You can start and stop the `lpd` daemon with the following commands:

/usr/lbin/lpd [-l]

/sbin/inetd/lpd [-start]

/sbin/inetd/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/sbin/lpd
```

8.4.4 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 ...
#
```

Log files should be monitored regularly for errors and should also be regularly deleted to prevent the files from filling up the available disk space. You can establish a regular clean up procedure using the `cron` utility. You can also control the volume of entries that are posted to `lpr.log` by specifying only the required priorities in the file /etc/syslog.conf. See the syslogd(8) reference page for instructions.

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 are usually located in the /var/adm directory. An error log file can be shared by all local printers, but you should specify the file in each /etc/printcap file printer entry.

8.4.5 Line Printer Daemon Filter Directory

The filters for the lpd daemon translate the data that you want to print into the format appropriate for your printer. You must specify the filter that matches each printer on your system. For example, to print files with the LN03R printer, you would use the ln03rof filter.

Input filters process job data being sent to the printer. Output filters process banner page data generated by the lpd. Many filters may be specified as either input or output filters, and will operate 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). For information on configuring printer accounting refer to Chapter 10.

Refer to the lpd(8) reference page for more information on using filter capabilities.

The lprsetup.dat(4) reference page lists the available print filters located in the /usr/lbin directory. For printers not listed in the reference page, consult the documentation or refer to the printer manufacturer for filter information.

8.4.5.1 General Purpose 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 will 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. Note that 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 will prevent potential file name conflicts. For example, copy file names before customizing as follows:

```
# cp ln17.pcf my_ln17.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.

8.4.6 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. Consult your printer manual 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. All bits should be cleared (using fc#0177777) before calling 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, example, `fc#0177777` clears all bits and the `fs` symbol is set to `0141` specifying the `OPOST`, `ONLRET`, and `OFILL` flag bits.

Table 8–4 lists each flag bit name, its octal value, and its description.

Table 8–4: 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	

Table 8–4: Flag Bits (cont.)

Flag	Octal Value	Description
FF1	0040000	
BSDLY	0100000	Backspace delay
BS0	0000000	
BS1	0100000	
OXTABS	1000000	Expand tabs to spaces

Refer to the `tty`(7) reference page for detailed information on flag bits.

8.4.7 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 of the mode bits, so you must either 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.

Refer to the `tty`(7) reference page for a detailed description of the status bits.

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 8–5 lists a description of each mode bit.

Table 8–5: Mode Bits

Mode	Octal Value	Description
ECHOKE	0000001	Echos 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

Table 8–5: Mode Bits (cont.)

Mode	Octal Value	Description
ICANON	0000400	Enable canonical input
ALTWERASE	0001000	Use alternate word erase algorithm
IEXTEN	0002000	Enable FLUSHO and LNEXT
XCASE	0040000	Canonical upper/lower presentation

8.4.8 Remote Printer Characteristics

If a printer will be used by users on remote machines, `/etc/printcap` files on the local machine attached to the printer and on remote machines that will use the printer must contain some network configuration information.

On the local machine attached to the printer, security is controlled by the entries in `/etc/hosts.lpd` or `/etc/hosts.equiv`.

You can optionally 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 machine 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. Refer to Section 8.4.1 for an example of an `/etc/printcap` file.

On the remote machine that will use the printer, you must specify the `rm`, `rp`, `lp`, and `sd` symbols.

The `rm` symbol specifies the name of the machine 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
```

8.4.9 Pagination and Imaging Parameters

Printer filters must know the size of an output page to perform proper page framing and line-feed and carriage returns (line folding).

For line printers, the pl and pw parameters specify the page length in number of lines (default is 66) and the column width in number of constant-width characters (default is 132), respectively. For example:

```
pl#55
pw#70
```

You should not specify a width of more than 80 characters for a letter-quality printer that uses 8 1/2-inch by 11-inch paper. If you specify a width that is greater than 80 characters on a printer, the page prints in landscape mode.

For high-resolution laser-type printers, the line length and page width parameters are py and px, which specify the number of pixels along the y- and x-coordinate planes of the printer output image area. Some printers can operate in either constant-width or imaging modes, so you must specify both sets of parameters. For example:

```
px#60
py#80
```

Refer to your printer's manual for its output characteristics.

8.4.10 Generic Print Filters

Tru64 UNIX provides print filters for generic and for local-language use.

The pcfof filter is a generic print filter that can be used with text and PostScript files. You can edit a template file /usr/lbin/pcf/template.pcf to customize the filter for use with additional printers. Refer to the pcfof(8) reference page for a list of options.

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 means that print jobs containing characters other than English do not have to be sent to printers where supporting fonts are resident. Refer to the wwpsof(8) reference page for a list of options. See the *Writing Software for the International Market* and the i18n_printing(8) reference page for additional information on local-language printer support.

8.4.11 TCP/IP Printing

TCP/IP printing allows you to submit print jobs to a remote printer that is directly connected to the network. To use this feature, your printer must contain a network interface card and must be registered with a TCP/IP node name and node address. TCP/IP can also be used to communicate with print server boxes or terminal servers which serve the same purpose as a built-in network interface card, but which make non-networked printer available on the network.

With TCP/IP printing, the local host manages print jobs in the same manner as it would manage print jobs for a local 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 printer 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 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`).

8.4.11.1 Setting Up TCP/IP Printing Manually

The following steps describe how to set up TCP/IP printing on a local host.

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 will listen 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 opt to create a service name, you will need the socket number in Step 2b when you edit the `/etc/services` file. Table 8–6 lists the socket numbers for three printers made by Compaq and one made by Hewlett Packard.

   Table 8–6: TCP/IP Socket Numbers

Printer	Socket Number
DEClaser 3500 (LN14)	10001
DEClaser 5100 (LN09)	10001
HP Laserjet 4m+	9100
LN17	2501

   To obtain the socket number for other printers, see your printer documentation. 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 and replace *servicename* with the name you will choose to enter in the `/etc/services` database in the next step. 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 can also remove the values for the `xs`, `xc`, `fs`, and `fc` control variables which establish settings that are relevant to the serial port driver. These 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 would add the following line to the `/etc/services` file:

```
declaser3500     10001/tcp
```

Note that the user-defined `declaser3500` string represents the service; it is the same string that you would have entered as the *servicename* in the `/etc/printcap` file in step 2a. Save the changes to the `/etc/services` file.

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` would be a useful fallback.

Once configured, 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 can also view the printer status and submit print jobs with the CDE print utilities.

8.5 Known Restrictions on the Use of Filters

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. Refer to 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 with which you could experience these 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 that would be supplied by the serial port driver is absent. Therefore, non-PostScript files that are not preformatted for the printer may not print out as they would 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, the printing problems may still be an issue for users who use serial-and-parallel-port to network-port converters, like the Compaq RapidPrint network interface box, which allow these printers to act like TCP/IP printers with built-in network support.
- The `hplaser4psof` PostScript filter works for PostScript files and for preformatted non-Postscript files (like PCL files), but it will likely produce unexpected results for files that have not been 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.

8.6 Testing and Troubleshooting Printers

This section provides a checklist for diagnosing printer problems. 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 can also 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. Refer to the `lpr`(1) reference page for more information on how to invoke these features.

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
```

Refer to the `lptest`(8) reference page for more information.

If a problem occurs on an existing printer or when adding a printer to a system, diagnose the problem as follows:

- Refer to the error log files specified in Section 8.4.4.
- Check the physical connections and if possible, swap the cable. Check all part numbers to ensure that cables and connectors are appropriate and suitable for the configuration. Note that 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. Refer to Section 8.4.1.
- Ensure that the printer daemon is present by using the following command:

  ```
  # ps agx | grep /usr/sbin/lpd
  ```

  Sometimes, the parent `lpd` process becomes hung, or a child process does. If the daemon is not running, it may have to be completely killed and restarted using the `kill -9` command on each process, or using the stop and start commands described in Section 8.4.3.

  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. consult the `lpd`(8) reference page for information.
- Check 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 (refer to Section 8.3.4). If print jobs are being created and queued, try configuring a different local or remote printer.
- Ensure that the appropriate spooling or device files have been created and that ownership and access are correct (refer to Section 8.4.2.6.1).

Note that for networked and remote printers, you also have to ensure that the systems are properly connected and authorized to transfer print jobs. Consult the *Network Administration* guide 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.

You should 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, as 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:

- Section 9.1 is an overview of the steps and options involved in creating a backup.
- Section 9.2 describes the main tasks involved in creating a backup.
- Section 9.3 describes how you set up a backup schedule.
- Section 9.4 describes the methods of creating a backup.
- Section 9.5 provides 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 9.6 describes the use of the `dump` command to perform a backup.
- Section 9.7 describes the use of the `restore` command to recover data from a backup.
- Section 9.8 describes the use of commands that enable you to archive individual files and directories, rather than complete file systems.
- Section 9.9 describes the use of `dxarchiver`, a graphical interface for archiving files and directories.
- Section 9.10 describes how you create a bootable tape. This is a bootable backup of the root file system and key system files that may be useful for disaster recovery.

9.1 Understanding Backup Tasks

This chapter describes basic backup operations for a system using the UFS file system. You may also 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, you should refer to the *AdvFS Administration* guide. 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 while it is being backed up. If a file is accessed while a backup is in process, the backup may not be able to record any 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) , you should refer to the *Logical Storage Manager* guide. Using features of LSM such as mirroring volumes, you might also 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. You can then break the mirror at any time to create an archive, with only a brief pause in system operations. (Note that 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 simply to recreate an environment on one or more systems.
- This chapter describes only those backup and archiving utilities that are included in the base operating system when installed.

  There are backup applications included on the *Associated Products* CD-ROM or supplied by third-party vendors. Refer to the *Installation Guide* for information on applications that might be included on the *Associated Products* CD-ROM. Refer to 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
  - Choosing a backup schedule
  - Creating small archives with pax, tar, and cpio 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
  - 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, the same type of dump utility and restore utility operations are supported as on as most other UNIX variants. See the dump(8) reference page for full details of all command options that are supported. Graphical and command-line tools for archiving and for creating a bootable tape of the standalone system (SAS) are also provided.

Prevention of data loss is an important part of any backup and recovery strategy. There are many tools for system monitoring that can be configured to help prevent situations that can 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 can be configured to monitor the system for specific high-priority events and report them to the administrator. See Chapter 13 for information on using EVM (event management) for information setting up the event-reporting strategy for your system and site. EVM can also be used 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, data files as well as system files, be protected from loss. Therefore, you should back up your entire system, including the system software. Most system files are static; that is, once they are installed they do not often change. Therefore, they do not need to be backed up as frequently as data files, which are dynamic, meaning they change constantly. Incremental backups are also possible, and should be considered if 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 are backed up regularly and static files are on file systems that are backed up occasionally. You may find that you have dynamic files on file systems that are backed 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 are backed up regularly. This allows you to back up those files without backing up an entire file system. You can also write shell scripts to automate these tasks and use the `cron` utility to automate the schedule. Refer to the `cron`(8) reference page for more information on scheduling tasks.

9.3 Choosing a Backup Schedule

When deciding how often to back up each file system, you should think about 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 will help 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 have changed since a 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 since the last level 0 dump, and a level 7 dump backs up only those files 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 a month you make a level 0 dump to tape of all the regularly backed up file systems. Then once a week, 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, very little time and tape storage are required for a backup.

9.4 Backup Methods

Depending on your needs and your local system configuration, there are several options for backing up data, as follows:

- The following command-line interfaces can be run 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 utility, `bttape`, is a SysMan Menu application that can be invoked from the command line, the SysMan Menu, or from CDE. Depending on how it is invoked, it will either run the command-line interface or a graphical interface that is appropriate to the windowing environment that you are using. (See Chapter 1 for more information.) The commands are `btcreate` and `btextract`.

  Use the bootable tape utility to create a bootable tape for recovery and to back up critical system data and customized system files. This utility 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 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 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.

Note that some tools provide you with additional options when you run them as superuser (root).

9.5 Preparing to Perform a Backup

This section contains 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 that should be performed.

Chapter 6 contains information on the UFS file system. Chapter 5 contains information on using disk and tape devices and on determining which disk and tape devices you want to back up. Also, refer to the information about the `cron` utility in Chapter 3 for information on scheduling regular backups. The following documentation contains other information that you may need to perform a backup:

- Books
  - The *AdvFS Administration* and *Logical Storage Manager* guides 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 will provide you with information on storage volume, media type, compression densities, and general operating instructions for a device.
- Reference pages
  - The `dump`(8), `rdump`(8), `vdump`(8), and the associated `restore`(8) and `rrestore`(8) reference pages provide information on the basic utilities for dumping file systems to tape and restoring them back to disk.
  - The `tar`(1), `pax`(1), and `cpio`(1) reference pages provide information on basic utilities for creating and manipulating archive files.
  - The `btcreate`(8), `btextract`(8), and `bttape`(8) reference pages provide information on the bootable tape interfaces.

- The cron(8) and crontab(1) reference pages provide information on creating cron entries for backup scripts that execute at specific dates and times.
- The mcutil(1) reference page describes the media changer manipulation utility.

- Both the Archiver and Bootable Tape graphical user interfaces provide online help that describes the different options available to the user, and defines what data can be entered into the data fields in each window.

9.5.1 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 restorecommands 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 utility 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 will be created on the bootable tape
  - /usr/lib/sabt/etc/fslist — A data file that specifies which file systems are backed up

9.5.2 Related Utilities

The following utilities are useful when performing backups:

- 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.
- 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 utility to check the size (in blocks)

of any directory, such as /usr/users. Command-line utilities du and df provide the same data.

- The 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.
- The Event Manager (EVM) provides a way of monitoring file system limits and will alert you of problems or can automatically start backups and cleanup of file systems.
- 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 also get information from the diskconfig tool, which has both command-line and graphical interfaces. This tool can be invoked from CDE Application Manager – Configuration folder, or from the SysMan Menu. This interface provides size information in megabytes, bytes, and blocks.

9.5.3 Prerequisite Tasks

The following prerequisite tasks apply to all the backup methods:

- Be familiar with the general principles of the interfaces.
- Ensure that all the required products or utilities are installed and configured (if necessary). The simplest way to do this is to refer to 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 refer to the hardware documentation for other test features. See also the hardware information tools listed in Section 9.5.2
- Check the size of the directories that will be backed 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 also use the graphical or command-line tools listed in Section 9.5.2.

- 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 floppy 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 this can often be done direct to the archive from the command line or by adding directories to existing archives). Refer to 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 will prompt 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 files:
  - One or more source files or directories. Note that in CDE, directories are identified as folders, and you can drag and drop them into the Archiver window from File View windows instead of typing 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`. (Note that you do not need to supply an extension or suffix for the archive file name. The utilities listed in Section 9.5.2 can assist you in finding the required device information, particularly if more than one tape drive is attached to a system.)
  - If you are restoring (unpacking) an archive, you need to supply the archive name, such as `/usr/archives/userfiles_990802.Z` or `/dev/tape/tape0_d0`, for a tape 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 /disk, /rdisk, /tape, or /ntape subdirectories. You can also use the graphical or command-line tools listed in Section 9.5.2 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. Refer to the tz(7) reference page 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 might corrupt the backup data. The dump command operates by checking the inodes of the files you want to back up. The inodes contain data such as table entries and other statistics. When 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, the backup may be corrupted.

  To shut down the system, unmount a file system, and check the integrity of a file system:

  1. Shut down the system 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'
     ```

     Refer to 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
     ```

     Note that the root file system remains mounted.

3. Use the `fsck` command to ensure the integrity of the file system.

   For example, to check a file system for an RZ57, unit 0, partition `c`, enter:

   ```
   # /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. Refer to *AdvFS Administration* 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 check 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

You should 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, which has the following command syntax:

dump *options* *filesystem*

The *options* parameter specifies a list of flags and their arguments and the *filesystem* parameter specifies the file system to be backed up. You should specify the file system with a full pathname. 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 `dump`(8) reference page describes the command options that you use to specify the characteristics of the tape device, such as block size, tape storage density, and tape length. 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. When *dump_file* is specified 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. The dump(8) reference page 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 command has the following command syntax:

dump -0u *filesystem*

The *filesystem* parameter specifies the name of a file system on your machine. The –0u 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. Note that 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, load a tape into your tape drive and enter:

   ```
   # dump -0u /
   ```

   After completing the backup, remove the tape from your tape drive.

2. To back up the /usr file system, load a new tape into your tape drive and enter:

```
# dump -0u /usr
```

After completing the backup, remove the tape from your tape drive.

3. To back up the /projects file system, load a new tape into your tape drive and enter:

```
# dump -0u /projects
```

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

You should set up a routine as part of your backup schedule to make it easier to remember which backup to do each day. This routine should 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, you should not keep this information on the computer system.

Once you have established 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 you 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 might 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. The rdump command has the following command syntax:

rdump -f *machine*: *device options filesystem*

The *machine* parameter specifies the name of the remote machine that has the backup device and *device* specifies the name of the backup device on that remote machine. The colon (:) between the *machine* and *device* parameters is necessary just as in other network file-addressing mechanisms.

The *options* parameter refers to the same list of flags available with the dump command.

The *filesystem* parameter refers to the local file system to be backed up.

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 should be transparent. Refer to the rmt(8) reference page for more information.

To back up the /projects file system from machine1 onto a tape drive on machine2 with the attached backup device /dev/rmt0h, enter the following command from machine1. The name of machine1 must be in the /.rhosts file of machine2 to allow access from machine1 to machine2.

```
# rdump -0uf machine2:/dev/tape/tape0_d0 /projects
```

The dump(8) reference page describes the options to the rdump command.

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 due to 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. When the shell procedures are finished, remove the backup tape and archive it.

Note that backup shell scripts are best used when the dump is small enough to fit on a single tape. You will need to specify the no-rewind device and the `–N` option to the `dump` command to inhibit the tape from automatically going off line when each dump is completed. When `dump` reaches the end of the tape, it will take the tape off line and someone will need to be available to replace the tape.

9.7 Restoring Data

Occasionally, you will have to retrieve files from your backup tapes, and you will likely 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 should be 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.

Once 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. Refer to *AdvFS Administration* for information on restoring the AdvFS file system. If you are using UFS, use the `restore` command to restore data from tapes made with the `dump` command. Because the `dump` command saves a single file system at a time, you must execute the `restore` command for each file system you want to restore. The `restore` command has the following command syntax:

restore *options*

The *`options`* parameter indicates the flags and arguments that you use to specify the characteristics of the tape device and special restore options. Refer to the `restore`(8) reference page for more information about these options. The following list describes the most commonly used options to the `restore` command:

`–i` The `i` (interactive) flag starts interactive restoration of files from the tape. After reading directory information from the tape, this option provides a shell-like interface that allows you to select the files you want to restore. The commands available in interactive mode are described in Section 9.7.3.

`-r`
: The `r` (restore) flag restores the entire contents of the file system on the backup tape into the current working directory. You should not do this except to restore an entire file system into an empty directory or to restore file system incremental dumps.

`-s`
: The `s` (skip) flag identifies which dump file on the media the `restore` command will use. This is useful when the dump media contains more than one dump image and not all of them will be restored. To effectively use this option, you must be consistent in the order in which you dump images to the tape. For example, if you dump multiple file systems to a single backup tape nightly, dump the file systems in the same order each night. This will assist you in locating a particular file or file system at restore time.

`-t` *names*
: The `t` (table of contents) flag creates a list of files and directories on the tape that matches the *names* argument. If you specify *names*, the `restore` command returns a list of the files and directories that are on the tape that matches the specified names. The *names* argument should be specified as `./`*filename* . For example, if the `.rhosts` file and the `staff` directory exist on the tape, the `restore -t ./.rhosts ./staff` command will list the file and the directory. If you do not specify *names*, the `restore` command returns a complete listing of the backed up files on the tape.

`-x` *names*
: The `x` (extract) flag restores from the tape the files and directories specified by the *names* argument. The *names* argument contains a list of files and directories to be restored from the tape. Specify names as `./`*filename* . For example, the `restore -x ./.rhosts ./staff` command will restore the `.rhosts` file and the `./staff` directory. If *names* specifies a directory name, then all the files in the directory are recursively restored.

–f *dump_file*	The f flag used with the *dump_file* argument restores the dump from the device specified by the *dump_file* argument instead of the default device, /dev/rmt0h.
–F *command_file*	The F flag used with the *command_file* argument specifies a file from which interactive restore commands are read. You should use this option in conjunction with the –i option.

If you are restoring a file system other than root or /usr, go to Section 9.7.1. If you are restoring the root and /usr file systems, go to Section 9.7.5. If the /var directory is on a separate file system than /usr, go to Section 9.7.5.

9.7.1 Restoring a File System

There may be times when you will need to restore a file system. This section describes a general procedure for restoring a file system. To restore individual files, go to Section 9.7.2.

When you restore a file system, you create a new file system and restore the files from the dump files by using the following command syntax. Refer to the *AdvFS Administration* guide for information on restoring an AdvFS file system.

newfs *raw_device*

mount *block_device* [*filesystem*]

cd *filesystem*

restore -Yrf *dump_file*

If the disk does not have a label, write the label by using the disklabel command before you create the new file system. Use the following command syntax to determine if the disk has a label:

disklabel -r *disk*

Writing a label with customized partition table settings may affect the entire disk. Use the following command syntax to write the default disk partition table:

disklabel -rw *disk disk_type*

The *disk* parameter specifies the disk that includes the device mnemonic and unit number. The *disk_type* parameter specifies the type of disk associated with the *disk* as described in the /etc/disktab file.

Invoke the editing option of the disklabel command to use the customized partition table settings. Refer to Chapter 6 or to disklabel(8) for more information. You can also use the diskconfig Disk Configuration interface.

The *raw_device* parameter specifies the full raw device pathname of the disk device on your system. The *block_device* parameter specifies the full block device pathname of the disk device on your system. The *filesystem* parameter specifies the full pathname of the file system you want to make available. The *dump_file* parameter specifies the full pathname of the file containing the dump data.

The following example shows the commands you use to restore a file system called /usr/projects on an RZ57 disk from a tape:

```
# disklabel -rw dsk1 rz57
# newfs /dev/rdisk/dsk1c
# mount /dev/rdisk/dsk1c /usr/projects
# cd /usr/projects
# restore -Yrf /dev/tape/tape0_d0
```

9.7.2 Restoring Files

When users lose files, they ask their system administrator to restore those files. Users may also ask you to restore an earlier version of a file. Whatever the reason for a file restoration, determine which tape contains the correct version of the file. If you are restoring a file on UFS, use the restore command to restore the file. If you are restoring a file on AdvFS, refer to the vrestore(8) reference page for information.

By asking 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. Use the following syntax:

restore -t ./*filename*

The –t option creates a list of files and directories on the tape that matches the *./filename* argument. 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 will overwrite the existing file. Restore the file by using the following form of the `restore` command:

restore -x ./*filename*

The file will be restored into your current working directory.

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 have been marked for reading begin with an asterisk (*).

`cd [ directory ]`	Changes the current directory to the directory specified by *directory*.
`pwd`	Lists the pathname of the current directory.
`add [ files ]`	Adds the files in the current directory or the files specified by *files* to the list of files recovered from the tape. Once they are specified to be read by the `add` command, files are marked with an asterisk (*) when they are listed with the `ls` command.
`delete [ files ]`	Deletes all the files in the current directory or the files specified by *files* 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 that have been 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 has been 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
```

Once in interactive mode, follow these steps to add the files to the list of files to be extracted:

1. Change to the `working` directory:

   ```
   restore > cd working
   ```

2. At the prompt, enter the file name as follows:

   ```
   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 (.) or current directory. For this example, reply with `n`.

   ```
   set owner/mode for '.'? [yn] n
   ```

6. Once the files are extracted, quit the interactive session as follows:

   ```
   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

There may be times when you need to restore files remotely. You can 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.

The rrestore command has the following syntax:

rrestore -f *machine*: *device* [*options*]

The *machine* argument specifies the name of the remote machine where the backup device is attached, and *device* specifies the name of the backup device on that remote machine. The colon (:) between *machine* and *device* is necessary just as in other network file-addressing mechanisms.

The *options* for the rrestore command are the same as for the restore command. See Section 9.7 for a description of the options.

To restore the ./working/file1 file onto the local directory on machine1 from a backup tape mounted on machine2 where the backup device /dev/rmt0h is attached, enter the following command from machine1. The name machine1 must be in the /.rhosts file of machine2 to allow access from machine1 to machine2.

```
# rrestore -xf machine2:/dev/tape/tape0_d0 ./working/file1
```

The rrestore command starts a remote server, /usr/sbin/rmt, on the remote machine to access the storage medium. This process should be transparent. Refer to the rmt(8) reference page for more information. See Section 9.7 for a description of the options to the rrestore command.

9.7.5 Restoring root and /usr File Systems

The root file system must be restored 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 /var.

The procedure in this section requires that you have access to the most recent dump files of your root and /usr file systems. You should use this procedure only when a catastrophic error occurs on the system disk, such as a disk crash or when an inadvertent deletion of either the root or /usr file systems renders the system inoperative.

The following steps show how you restore from a level 0 dump of files, using the text-based (or character cell) interface to perform the task:

1. Load the installation software. For removable media such as tape or CD–ROM, insert the media into the appropriate drive. For RIS installations, verify that the inoperative system has been registered on the RIS server. See *Sharing Software on a Local Area Network* for details. If the dump file is located on a remote system, include the host name of the inoperative system in the `/.rhosts` file of the remote system. For security reasons, be sure to delete the host name from the `/.rhosts` file after the restore operation is complete.
2. Boot the operating system as described for your processor and distribution media in the *Installation Guide*. If your system had a graphical interface, the `Installation Setup` screen would be displayed, rather than the following menu. However, in both cases you would select the `UNIX Shell` option.
3. Select the `UNIX Shell` option at the prompt.
4. If necessary, create the special files for the root file system device and dump file device. Refer to Chapter 5 for information on device special files and the `dsfmgr` and `MAKEDEV` commands

   After creating the system disk special file, configure the network by configuring the network interface and creating the host name database (`/etc/hosts`). Use the `ifconfig` command with the following syntax to configure the network interface:

   ifconfig *interface_id local_address mask*

   The *`interface_id`* parameter refers to the network device mnemonic. Refer to the `uerf`(8) reference page for information about obtaining an interface ID. The *`local_address`* parameter specifies the Internet address for the local host. The `netmask` *mask* parameter specifies how much of the address to reserve for subdividing networks into subnetworks. You can get the `netmask` value by entering the `ifconfig` command on a system within the immediate area. For example, to get the `netmask` value from the system `ln0`, enter:

   ```
   # ifconfig ln0
   ```

   Refer to the `hosts`(4) and `ifconfig`(8) reference pages for more information. Enter the following commands to configure the network for the system `localsystem`, with an Internet address of `120.105.5.1`, connected by an Ethernet interface to the remote system `remotesystem`, with an Internet address of `120.105.5.2`:

   ```
   # cd /etc
   # echo "127.0.0.1 localhost" >> hosts
   # echo "120.105.5.2 remotesystem" >> hosts
   ```

```
# ifconfig ln0 120.105.5.1 netmask 0xfffffc00
```

Some older systems broadcast all 0s instead of all 1s. In this situation, you must also specify the `broadcast address`.

5. Change to the root directory.

   ```
   # cd /
   ```

6. If the disk does not have a label, which could occur if the disk was physically damaged or replaced, write the default disk partition tables and bootstrap programs. The disk partitions and bootstrap programs should be operational. To determine if the disk has a valid label, use the `disklabel` command with the following syntax:

   disklabel -r *disk*

   Use the `disklabel` command with the following syntax to write the default disk partition table:

   disklabel -rw *disk* *disk_type*

   The *disk* parameter specifies the disk that includes the device mnemonic and unit number. The *disk_type* parameter specifies the type of disk associated with *disk* as described in the `/etc/disktab` file. For example, to write the default disk partition tables on an RZ57 disk, unit 0, enter the following command:

   ```
   # disklabel -rw /dev/disk/dsk0 rz57
   ```

   Note

   The options used with the `disklabel` command in this procedure cause the default disk partition tables to be written to the disk. Writing a label with customized partition table settings may affect the entire disk. If the disk you are restoring has customized partition table settings, invoke the editing option of the `disklabel` command. Refer to Chapter 6 or to the `disklabel`(8) reference page for more information.

7. Create a new root file system by using the following command syntax:

   newfs *raw_device*

   The *raw_device* parameter specifies the full raw device pathname of the disk device on your system. For example, to create a new file system on an RZ57, unit 0, enter:

   ```
   # newfs /dev/rdisk/dsk0a
   ```

8. Mount the file system by using the following command syntax:

   mount *block_device* [/mnt]

The *block_device* parameter specifies the full block device pathname of the disk device. For example, to mount the file system created in the previous step, enter:

```
# mount /dev/disk/dsk0a /mnt
```

9. Restore the file system:

   - If you are restoring dump files from a local file system, change to the /mnt directory, insert the medium containing the dump file, and enter the restore command with the following command syntax:

     restore [-Yrf] [*dumpfile*]

     The *dumpfile* parameter specifies the pathname of the file that contains the dump data. For a tape, you would 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 /mnt directory and use the rsh command with the following syntax:

     rsh [*remote_hostname*] ["dd if= *dumpfile* bs= *blocksize*" | restore -Yrf -]

     The *remote_hostname* parameter specifies the host name of the remote system that contains the dump file. The *dumpfile* parameter specifies the full pathname of the dump file on the remote system; and the *blocksize* parameter is necessary for reading from a tape.

     The dump file must be read with the same block size as was used when writing to 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 was written using the default block size, enter:

     ```
     # cd /mnt
     # rsh remotesystem  "dd if=/dev/tape/tape0_d0 bs=10k" \
     | restore -Yrf -
     ```

10. Change to the root directory and unmount the file system.

    ```
    # cd /
    # umount /mnt
    ```

11. Restore the /usr file system.

    - If the /usr file system is on the same device as root, the process is similar to steps 7 through 10. To restore the /usr file system on the g partition of the same device as the root file system from the same tape device, enter the following sequence of commands:

```
# newfs /dev/rdisk/dsk0g
# mount /dev/disk/dsk0g /mnt
# cd /mnt
# restore -Yrf /dev/tape0_d0
# cd /
# umount /mnt
```

- If the /usr file system is on a different device from root, the process is similar to steps 4 through 10.

12. Halt the system.

```
# halt
```

13. Boot the system as described for your processor and distribution media in the *Installation Guide.*

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. These utilities are often used to make a small archive of files that can be distributed 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 using the 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 create a tar archive to device /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 /usr/glenn except for file ./.netscape. See the tar(1) reference page 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 create a pax archive of the current directory to device /dev/tape/tape0_d0, enter:

```
# pax -w -f /dev/tape0
```

The following command reads the archive a.pax, extracting all files rooted in the directory /usr, relative to the current directory:

```
# pax -r -s ',^//*usr//*,,' -f a.pax
```

See the pax(1) reference page 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 create a cpio archive to tape device /dev/tape/tape12_d0, enter:

```
# cpio -ov < file-list -O/dev/tape12_d0
```

See the cpio(1) reference page for more information.

9.9 Using dxarchiver

The Archiver, dxarchiver, is a graphical interface 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 floppy disk
- Uncompress incoming archive files and compress newly created files
- Retrieve stored files from an archive file or device such as a tape or floppy disk

As dxarchiver 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 9.5.3, 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 /usr/bin/X11/dxarchiver 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. Refer to the reference pages for more information, but note that not all options may be available from the graphical interface.
3. Select any Archive Options. Note that you can only append to an existing archive, and you cannot further compress an existing archive if it 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 . or users/chan.)

When you are recovering files from an archive, you will only be able to write them to a temporary location if you specified a relative path during the archiving process. Otherwise, files will be restored to their original locations. (Potentially overwriting the existing version unless you rename it.)

4. Specify the source, the files and directories that will be archived. 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, use the OK button to add them to the container.

5. When all required files are specified, press Archive... and the Archiver: Archive window will be 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. Note that you do not need to enter a file name extension; the Archiver will add an identifier such as .Z.

   When you press OK, the destination is displayed under the Destination Container box.

7. Press Create Archive. A window titled Archiver working will be displayed, flashing a green button to indicate that the archive is being written. The files being archived are displayed in the Destination Container.

8. When the archive is complete, you can optionally print a copy of the files list to keep as a record with the tape.

9. Press Cancel to return to the Archiver main window. You can optionally 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 will be read and the contents displayed in the Show Contents Window. This step is recommended before you proceed to delete any files from the disk that was backed up.

To extract an archive, you need to specify a destination on a target device such as a disk. Note that if you are not recovering a damaged file system on a complete disk partition, you may want to consider using a temporary location rather than overwriting existing directories. You can then restore individual files and directories as needed. You can also opt to restore selected files from the archive. The process is as follows:

1. Invoke `/usr/bin/X11/dxarchiver` 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. To select individual files and directories, press Show Contents... and the tape or archive file will be read and the contents displayed in the Archiver Show Contents window. Select individual files or directories as follows:
   - In the Archiver Show Contents window, click on a file or directory to highlight it.
   - Move to another file or directory, hold down the Ctrl key and click to select it.
   - When all required files are selected, press OK in the Archiver Show Contents window. The files will be 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. Press the Extract... button 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. Note that this must be an existing directory, or you must open a terminal and create it with `mkdir` (or create a folder with the New Folder option in CDE File Manager). The destination is displayed under the Destination Container box.
5. Press Extract Contents to begin the extraction. A window titled Archiver Working will be displayed, flashing a green button to indicate that the archive is being extracted. The files being recovered are displayed in the Destination Container.
6. When the archive is complete, you can optionally print a copy of the files list, to keep as a record.
7. Press cancel to return to the Archiver main window. Before exiting, you should 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. This step is strongly 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 also add additional files systems to the tape for data or programs that you may need.

To create the SAS kernel, you must use the `bttape` interface or the `btcreate` command-line utility. Once you have created the kernel, you can restore the customized image using the `btextract` utility. The following sections provide an overview of the `bttape` interfaces, both for the CDE graphical interface and the `btcreate` and `btextract` utilities. For information on syntax and examples, see the reference page for each utility. Note that this interface is part of the SysMan Menu and can be invoked and used from a terminal, and various windowing environments.

9.10.1 Restrictions and Requirements

The following sections describe the restrictions and requirements for building an SAS kernel on a system.

9.10.1.1 Tape Device Requirements

When using 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 will not work. Tapes with a variable block size, such as the QIC-320 and QIC-525, will work with bootable tape. Using an improperly configured QIC tape drive to create a bootable tape will result 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` utility 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 the tape must be changed. If the tape is not changed promptly, warning messages repeat periodically until the tape is changed. When you change the tape, the warning messages stop.

The behavior of the open call to a tape device has changed. You can no longer use write mode to open a write-protected tape. An attempt to open the tape will fail, returning the following message:

```
EACCES (permission denied)
```

If an application is written so that it attempts to open the tape device with O_RDWR when the intention is only to read the tape, the open attempt will fail. Applications should be changed to open the device with O_RDONLY. For applications that cannot be changed, use the following command to obtain the previous behavior of the open call:

```
# sysconfig -r cam_tape open_behaviour=0
```

9.10.1.2 Supported Software and Devices

For this release, bootable tape will not work with the LSM product. The following processor platforms are supported:

- DEC 3000-500
- DEC 3000-400
- DEC 3000-600
- DEC 3000-300
- DEC 3000-300X
- DEC 3000-900
- DEC 2100
- AlphaStation 600
- AlphaStation 200
- AlphaServer 1000A
- AlphaServer 2100
- AlphaServer 4100

For tape drives, you must ensure that the kernel was built with the tape drive connected to your system. If the drive was not connected when the kernel was built, you will see dump errors and the system will not be able to boot from the tape drive. The following tape devices are supported:

- TLZ06, 4mm, 2.0GB/4.0G
- TLZ07, 4mm, 4-8G
- TZK10, QIC tape, 320MB-525 MB
- TZK11, QIC tape, 2.0G
- TZ86, 5-1/4-inch cartridge

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` utility provides both a noninteractive and interactive user interface. Both require that you have superuser (root) privileges.

9.10.2.1 Gathering Information

To prepare for a `btcreate` session, you must have the following information available:

- 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 30720 blocks (512 bytes per block). This disk partition should not be mounted 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 shows valid UFS and AdvFS entries:

  UFS:

  ```
  /dev/dsk1a   /     ufs
  /dev/dsk1g   /usr  ufs
  ```

  AdvFS:

  ```
  root_domain#root   /     advfs
  usr_domain#usr     /usr  advfs
  ```

Note

Do not select swap partitions for file system backups.

- An *addlist_file*, which lists the files or directories you want to include on the miniroot file system.
- An *fslist_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. The file must be written in the Bourne shell language (`sh1`) as it is the only shell provided

on the miniroot file system. The btcreate utility copies the custom_install.sh file onto tape and places it in the sbin directory on the miniroot file system. The btextract utility 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 the btcreate command 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 will be required.
- You can label disks using your own custom disklabel script. Your customized disklabel script 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 will label the disks in the usual manner. Refer to the disklabel(8) reference page for more information.

9.10.2.2 Creating the SAS Kernel

To create the SAS kernel, the btcreate utility 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  30720
```

These modifications indicate that a memory file system of 30720 is being configured. 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 utility executes the doconfig command and moves the bootable kernel to the /usr/sys/bin directory. For information on syntax format and flags, see the btcreate reference page.

9.10.3 Using the btextract Utility

The btextract utility is a shell script that restores file systems from tapes that contain a SAS kernel created using the btcreate utility. You have the option of performing a default restoration or an advanced restoration of the system.

Performing a default restoration, you can duplicate the customized system on more than one machine of the same hardware platform type; however, you cannot specify which disk partitions to use for the restore operation. Instead, the btextract utility restores file systems using the disk partition information gathered during the btcreate session; all existing information is overwritten.

Performing an advanced restoration, you can specify which disk partition to use, but the customized system can only be duplicated on a machine of the same hardware platform type.

To use the btextract utility, 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 is completed, the shell invokes the btextract utility. If you created a /usr/lib/sabt/sbin/custom_install.sh script during the btcreate session, the btextract utility invokes the custom_install.sh script before exiting. See the btcreate reference page for more information.

After the btextract utility completes its task, you must shut down the system, then reboot the system from the restored disk as follows:

```
# shutdown -h now
>>> boot DKA100
```

In this example, DKA100 is the BOOTDEV.

For more information and examples, see the btextract reference page.

9.10.4 Using bttape with the Graphical Interface

The following steps describe the basic process for creating a bootable tape and assumes that you have already gathered the necessary device data as described in Section 9.10.2.1, and the tape device is ready to save. (See Chapter 1 for information on the various interface options when invoking bttape):

1. Invoke the bttape interface. For example, from the CDE Application manager, choose System_Admin, then Storage_Management and click on the Bootable Tape icon. A window titled Bootable Tape Creation on <hostname> will be displayed and you should complete the fields or select 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. Click on the apropriate button for either option.
  - The option to specify a disk partition name such as `dsk0b` with the Specify Disk Partition/mfs... button. (This button dispays 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_d0`. This is the name of the device on which the SAS kernel will be saved, but you can opt to 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. Note that you cannot exceed 360 KB of data in the mfs. This list of files is stored in `/usr/lib/sabt/etc/addlist` by default but you can opt to create and specify your own location.

  To create a new append file, or modify an existing append file:

  a. Press the Create/Modify Miniroot Append File button to display the Create/Modify window.
  b. Press Add to display 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 will be located, such as `/sbin`. Press OK to return to the Create/Modify window.
  c. The files to be appended will be listed in the Contents of file: box. Press OK to return to the main window for Bootable Tape Creation.
- 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. Press the Create/Modify File Systems Backup File... button to display the Create/Modify window.

b. Press Add to display 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`. Press OK to return to the Create/Modify window.

c. The file systems to be backed up will be listed in the Contents of file: box. Press OK to return to the main window for Bootable Tape Creation.

2. After completing the required fields, you are ready to create the tape. Press OK to proceed. A message window is displayed to indicate that the task has started. The creation of the tape may take twenty or more minutes, depending on the speed of the devices used.

   If the task cannot be completed, a further message is displayed, informing you that the error log is located in `/var/adm/btcreate.log`.

3. When the tape has been successfully written, a window is displayed that confirms the success and the location of the log file, `/var/adm/btcreate.log`.

   Print the `btextract`(8) reference page and store it with the tape for future reference.

4. Use the instructions in Section 9.10.3 and the `btextract`(8) reference page to restore the bootable SAS kernel.

10

Administering the System Accounting Services

This chapter describes how to set up and use the system accounting services. 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. You can also manually invoke accounting shell scripts and commands to obtain accounting information when you need it. The following information is included in this chapter:

- Section 10.1 provides an overview of system accounting, describing what can be recorded, the scripts and commands used, and the system files and logs.
- Section 10.2 describes how to set up system accounting.
- Section 10.3 describes how to start or stop accounting.
- Section 10.4 describes how connections to the system are recorded, the log files and explains the associated commands.
- Section 10.5 describes how user jobs running on the system are recorded, the log files, and explains the associated commands.
- Section 10.6 describes how the use of disk storage is recorded and explains the associated commands for retrieving data.
- Section 10.7 describes how the use of system administration services are recorded and explains the associated commands for retrieving data.
- Section 10.8 describes how the use of printer services are recorded and explains the associated commands for retrieving data.
- Section 10.9 describes the reporting features available for all accounting operations and explains the associated commands.

10.1 Accounting Overview

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	Checks 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 check 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. Refer to 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. Refer to 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 to Table 10–5 list the accounting files.

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 queueing (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: 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: 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.

Table 10–4: Summary Files in the /var/adm/acct/sum Directory (cont.)

Name	Type	Description
`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: 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
OSFACCT500   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 sections that follow describe these steps in detail.

1. Enable accounting in the /etc/rc.config file.
2. Verify the qacct, pacct and fee files.
3. Edit the /usr/sbin/acct/holidays file to specify prime time, nonprime time, and holidays.
4. To enable automatic accounting, modify the files in the /usr/spool/cron/crontabs directory to invoke accounting shell scripts and commands.

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. Refer to Section 10.3 for more information.

10.2.2 Verifying the qacct, pacct, and fee Files

The qacct queueing 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 accidentally destroyed) you must recreate them as CDSLs. See Chapter 6 and the cdslinvchk(8) and mkcdsl(8) reference pages 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.

Note that the files must also be owned by the adm user and group and have permissions of 644. Use the chown and chgrp commands to reset these values if required.

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 checks 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. You can also add this command to the `runacct` shell script.

`pac`
: This command displays printer accounting records. You can direct the output to a file. To enable printer accounting, refer to Section 10.8.

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.

Refer to Chapter 3 and to the `crontab`(1) reference page 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 will not work, and the following message will be 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, refer to the `wtmpconvert`(8) reference page 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.
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 checks 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. Refer to the fwtmp(8) reference page 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 re-create 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. Refer to 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...]

Refer to the ac(8) reference page 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. Refer to the acctcon1(8) reference page 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 this 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 Jan  13 17:20:12 1994 EDT
to   Fri Jan  14 09:56:42 1994 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*]

Refer to the prctmp(8) reference page 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
- tty 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 Jan 21 10:56   still logged in
root    ttyp1     raven     Fri Jan 21 08:59 - 09:00  (00:00)
```

```
root    ttyp0     raven     Thu Jan 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 will grow in size. The `ckpacct` command checks 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.

Table 10–7: The tacct File Format (cont.)

Column	Heading	Description
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.
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/chargefee` shell script.
16	PROCESSES	Specifies the total number of processes spawned by the user during the accounting period.

Table 10–7: The tacct File Format (cont.)

Column	Heading	Description
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 will be 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. Refer to 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 checks 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
- tty name
- Process start time
- Process end time
- Real seconds
- CPU seconds
- Mean memory size (in kilobytes)

Refer to the acctcom(8) reference page 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 Jan 17 02:00:00 1994
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. Refer to the `sa`(8) reference page 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 also 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
```

Refer to the acctcms(8) reference page 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 will consist 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 also 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 also invoke the command manually.

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. Refer to the fstab(4) reference page for more information.

Note

If you have a swap space specified in the /etc/fstab file, the dodisk shell script will 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*]

Refer to the diskusg(8) reference page 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 checks 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*]

Refer to the acctdusg(8) reference page 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 usually merged 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. Refer to 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*...]

Refer to the `pac`(8) reference page 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.
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 will 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 will be merged with other records to create a daily report.
PROCESS	Produces process accounting report files.
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 been successfully completed. The runacct shell script logs messages to this file. You can use this file to determine which tasks have been successfully completed. The following is an example of an active file:

```
Fri Feb 4 11:02:56 EST 1994
-rw-r--r-- 1 adm  adm      0 Jan 31 03:00 /var/adm/acct/nite/dacct
-rw-rw-r-- 1 root system 924 Jan 05 10:45 /var/adm/wtmp
-rw-rw-r-- 1 adm  adm      0 Jan 08 13:46 fee
-rw-rw-r-- 1 adm  adm      0 Jan 07 02:00 pacct
-rw-rw-r-- 1 adm  adm   8904 Jan 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. Note that the runacct shell script may not have successfully completed this state.
- 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 will 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/last-date 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. Check 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.

`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. Check 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. File records are usually merged according to the user identification number or the user login name.

The default command output is in binary format, but you can also 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 also 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.

Refer to the acctmerg(8) reference page 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 re-create the `/var/adm/sum/tacct`*mmdd* file, enter:

```
# acctmerg –i < tacct_temp > tacct.0617
```

6. To re-create 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*

Refer to the `prtacct`(8) reference page 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]

Refer to 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. Refer to 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

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. You 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 might 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 quickly resolve problems and bring the system back online.

The topic of monitoring is closely related 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 in getting your system up and running quickly after a fault in the operating system or hardware. Therefore, it is recommended that you at least follow the monitoring guidelines in Section 11.1.

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. Refer to your Owner's guide for information on hardware test commands.

The following topics are covered in this chapter:

- Section 11.1 contains basic monitoring guidelines and provides an overview of the utilities. It also provides pointers to related topics.
- Section 11.2 describes some of the monitoring utilities in greater detail.
- Section 11.3 describes 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. Not all systems support this feature.
- Section 11.4 describes how you use the system component the test utilities. Note that your system hardware also provides test routines. Refer to the Owner's Manual for more information. If you need to obtain

detailed information on the characteristics of system devices (such as disks and tapes) see the `hwmgr` command, documented in Chapter 5.

11.1 Overview of Monitoring and Testing

This section provides some general guidelines for monitoring your system, and 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 you will use to monitor your system on a daily basis.

   Review the overview of monitoring utilities provided in this section. Based on the system configuration, select utilities that meet the requirements of the configuration and your monitoring needs. For example, if you have a graphics head terminal and you want to monitor several distributed systems you might want to set up the SysMan Station. If you want to monitor a single local server the `dxsysinfo` window might adequate.

   If applicable, set any attributes that trigger warnings and messages. For example, you might want to set a limit of 85% full on all file systems to prevent loss of data due to a full device.

   Note

   Many optional subsystems provide their own monitoring utilities. You should familiarize yourself with these interfaces and decide whether they are more appropriate than the generic utilities.

2. Establish a baseline

   Run the `sys_check -all` utility to:

   - Establish a no-load baseline.
   - Determine whether any system attributes need to be tuned.

   If necessary, use the information from `sys_check` to tune system attributes. Refer to the *System Configuration and Tuning* guide for information on Tuning your system. Store the baseline data where it can be easily accessed later, such as on another system. You might also want to 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 that you want to monitor, such as -perf. This might have a small impact on system performance, so you might not want to run it during peak end-user demand.

   Analyze the output from the sys_checkutility and perform any additional recommended changes that meet with your operational requirements. This might involve further tuning of system attributes or configuration changes such as the reallocation of system resources using a utility such as the Class Scheduler. See Section 11.2.2 for information on using the sys_checkutility.

4. Set up Event Management (EVM)

   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 EVM.

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, 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 might also be required when diagnosing system problems. See Chapter 14 for information on configuring the crash dump environment.

   - Install and configure any optional performance utilities, such as the Performance Manager. If supported by the target system, you should also 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 interfaces that filter and track system data in real time and display it on a graphics head terminal.

Choose monitoring utilities that best fit your local environment and monitoring needs and consider the following:

- Using monitoring utilities canimpact system performance.
  - To help diagnose problems in performance, such as I/O bottlenecks, a simple command such as `iostat` might be adequate.
  - To provide a quick visual check of resources on a single-user system, the X11 System Information interface (`dxsysinfo`) might 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 check whether they are functioning correctly), the web-based Insight Manager utility can be used for both UNIX servers and client PC systems.
- You might need to provide output from a monitoring utility to your technical support site during problem diagnosis. It will greatly reduce your system downtime 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
```

Refer to the `vmstat`(1) reference page 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
```

Refer to the `iostat`(1) reference page for more information.

`who`

The `who` 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 `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
```

Refer to the `who`(1) reference page for more information. See also the `users`(1) reference page.

`uptime`

The `uptime` command reports how long the system has been running. Refer to the `uptime`(1) reference page for more information.

Refer also to the `netstat` command and the *Network Administration* guide 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. Refer to 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 Insight Manager [imconfig]

Invokes the interface that enables you to configure Insight Manager and start the Insight Manager daemon. Refer to Chapter 1 for information on configuring Insight Manager.

View Virtual Memory (VM) Statistics [vmstat]

This is a SysMan Menu interface to the `vmstat` command, described previously in this section.

View Input/Output (I/O) Statistics [iostat]

This is a SysMan Menu interface to the `iostat` command, described previously in this section.

View Uptime Statistics [uptime]

This is a SysMan Menu interface to the `uptime` command, described previously in this section.

In addition, the following options are provided under the Support and Services menu item:

Create escalation Report [escalation]

Invokes the escalation report feature of the `sys_check` utility. The escalation report is used only in conjunction with diagnostic services, and will be requested by your technical support organization. Refer to Section 11.2.2 for more information on using the escalation options in `sys_check`.

Create configuration Report [config_report]

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. Note that using this option creates a full default report which can take many minutes to complete and can impact system performance. Refer to Section 11.2.2 for more information on using the `sys_check`utility.

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. Refer to Chapter 1 for information on using the SysMan Station.

11.1.2.3 X11-Compliant Graphical Interfaces

The operating system provides System Management folders containing several graphical interfaces that are typically used under the default Common Desktop Environment (CDE) windowing environment. You can invoke these interfaces from the CDE Front Panel by clicking on the Application Manager icon to display the Application Manager folder. From this folder, select the System Admin icon, and then the MonitoringTuning icon. This folder provides icons that invoke the following SysMan Menu items:

Configuration Report

This icon invokes a graphical interface to the system configuration report feature of the sys_check utility.

Escalation Report

This icon invokes a graphical interface to the escalation report feature of the sys_check utility.

Insight Manager

This icon nvokes the interface that enables you to configure Insight Manager and start the Insight Manager daemon.

The remaining applications in this folder relate to system tuning. Refer to the *System Configuration and Tuning* guide for information on tuning using the Process Tuner (a graphical interface to the nice command) and the Kernel Tuner (dxkerneltuner).

The Tools folder provides graphical interfaces to the commands such as vmstat. Invoke these interfaces from the CDE Front Panel by clicking on 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 interface to the iostat command, described previously in this section.

Network Statistics

This is a graphical interface to the netstat command. Refer to the *Network Administration* guide for information on monitoring your network.

System Messages

This is a graphical interface to the /var/adm/messages log file, which is used to store 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, refer to Chapter 12 and Chapter 13.

Virtual Memory Statistics

This is a graphical interface to the `vmstat` command, described previously in this section.

Who?

This is a graphical interface to the `who` command, described previously in this section.

The remaining X11–compliant monitoring application is located in the Application Manager – DailyAdmin folder. Click on 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 also be obtained using commands such as `vmstat`.
- Two warning buttons for files and swap. These buttons are filled with 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. Refer to Chapter 6 and Chapter 9 for information on increasing the available file system space.

11.1.2.4 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 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.

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.

Performance Manager

Performance Manager is an SNMP-based, user-extensible, real-time performance monitoring and management utility. It enables you to detect and correct performance problems on a single system (or a cluster). Performance Manager has a graphical user interface (GUI), and a limited command-line interface using commands such as the getone command to read and display lines of data. The GUI can be configured to display tables and graphs, showing many different system parameters and values, such as CPU performance, physical memory usage, and disk transfers.

Performance Manager comprises two primary components: Performance Manager GUI (pmgr) and Performance Manager daemon (pmgrd). Additional daemons are used in monitoring TruCluster clusters (clstrmond) and the Advanced File System (advsfd), supplied in the AdvFS Utilities subset.

The Performance Manager software subsets are included on the Associated Products, Volume 2 CD-ROM. No license is required to install and use the software. For an overview of features refer to the release notes. The PostScript file is PMGR***_RELNOTES.ps and the text file is PMGR***_RELNOTES.txt. The *Performance Manager* guide is provided in the Software Documentation CD-ROM.

11.1.3 Related Documentation

The following topics are closely related to system monitoring and testing:

- Refer to Chapter 10 for information on administering the system accounting services, which enables you to monitor and record access to resources such as printers.
- Refer to Chapter 12 for instructions on configuring and using basic system event logging using the basic binlogd and syslogd event

channels. This chapter also describes how you access system log files, where events and errors are recorded.

- Refer to 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.
- Refer to the *Network Administration* guide for information on monitoring the system's networking components.
- Refer to the *System Configuration and Tuning* for information tuning your system in response to information gathered during monitoring and testing.

11.2 Configuring and Using Monitoring Utilities

This section introduces some of the monitoring utilities and iscusses their setup and use. Refer to the documentation and reference pages supplied with each application for more information. Refer to 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. Refer to Chapter 12 and Chapter 13 for information on these topics.

11.2.1 Using collect 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 fully defined in the `collect`(8) reference page. 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 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 using the `-H` (history) option. Compressing data files for economical storage.
- Specifying specific users, groups, and processes for which data is to be sampled.
- Using the `-p`, you can specify multiple data files and use the `collect` utility to play them back as one stream. Using the `-f` option you can combine multiple binary input files into one binary output file.

The `collect` utility locks itself into memory using the page locking function `plock()`, and cannot be swapped out by the system. It also raises its priority using the priority function `nice()`. If required, page locking can be disabled using the `-ol` command option and the priority setting can be disabled using the `-on` command option. However, using `collect` should have minimal impact 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 update this report regularly or as your system configuration changes.
- The opportunity to perform automated checking of many system attributes (such as tuning parameters) and receive feedback on settings that might be more appropriate to the current use of the system.

  The `sys_check` utility also checks 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 it produces an HTML document on standard output. Used with the -escalate flag, the script produces /var/tmp/escalate* output files by default. 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 -h
```

The output produced by the sys_checkutility typically varies between 0.5MB and 3MB in size and it can take from 30 minutes to an hour to complete the check. Refer to the sys_check(8) reference page for more details of the various command options. You can greatly reduce the run time 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 also 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, refer to 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 Insight Manager and can be read from the Insight Manager Configuration Report option. See Chapter 1 for information on using 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 on the reliability and availability of the operating system and its hardware environment such as crash data files. The information is automatically copied to your systems vendor by internet mail or DSN link, if available. Using this data, performance analysis reports are created and distributed to development and support groups. This information is only used internally 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 10 minutes of your time. MPH will not impact or degrade your system's performance because it runs as a background task, using negligible CPU resource. 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 (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 2:00am, it extracts any new events records from the default event log `/var/adm/binary.errlog`.
- Every day at 3:00am it transfers the event and timestamp data and any new `crashdc` data files in `/var/adm/crash` to the system vendor. The average transfer is 150 blocks of data.

For more information, see http://availability.ayo.dec.com/cars or contact mph@Compaq.Com if you have specific questions.

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 should be the root user and 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
   ```

   Where ******* is the version number, such as 025.
3. Enter the 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 will have to restart the MPH script as described in the MPH documentation. See the `mph`(1) reference page for more information.

11.3 Environmental Monitoring

On any system, thermal levels can increase because of poor ventilation, overheating conditions, or fan failure. Without detection, an unscheduled shutdown could ensue, causing the system's loss of data or damage to the system itself. By using Environmental Monitoring, the thermal state of AlphaServer systems can be detected and users can be alerted in time enough to recover or perform an orderly shutdown of the system.

The Environmental Monitoring framework consists of four components:

- The loadable kernel module and its associated APIs.
- The Server System MIB subagent daemon.
- The `envmond` daemon.
- The `envconfig` utility.

tThese 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 current temperature of the system. 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 has not been 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 will differ when the hardware support is provided for this feature.
`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 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 using the `HIGH_TEMP_THRESH` function type, which reads the platform-specific upper threshold operational temperature.

11.3.1.3 Server System MIB Subagent

The Server System MIB Agent, (which is an eSNMP subagent) is used to export 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.

For each Server System MIB variable listed in Table 11–3, code is provided in the subagent daemon, which accesses the appropriate parameter from the kernel module through the CFG interface.

11.3.2 Monitoring Environmental Thresholds

To monitor the system environment, the `envmond` daemon is used. You can customize the daemon by using the `envconfig` utility. The following sections discuss the daemon and utility. For more information, see the `envmond` and `envconfig` reference pages.

11.3.2.1 Environmental Monitoring Daemon

By using the Environmental Monitoring daemon, `envmond`, threshold levels can be checked and corrective action can ensue before damage occurs to your system. Then the `envmond` daemon performs the following tasks:

- Queries the system for threshold levels.
- When the cooling fan on an AlphaServer 1000A fails, the kernel logs the error, synchronizes the disks, then powers down the system. On all other fan failures, a hard shutdown ensues.
- Notifies users when a high temperature threshold condition has been resolved.
- 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 `envmond` daemon must be started during the system boot, but after the eSNMP and Server System MIB agents have been started. You can customize the `envmond` daemon using the `envconfig` utility.

11.3.2.2 Customizing the envmond Daemon

You can use the `envconfig` utility to customize how the environment is queried by the `envmond` daemon. 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:

- Turn environmental monitoring on or off during the system boot.
- Start or stop the `envmond` daemon after the system boot.
- Specify the frequency between queries of the system by the `envmond` daemon.

- Set 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.
- Specify the grace period allotted to save data if a shutdown message has been broadcasted.
- Display the values of the Environmental Monitoring variables.

11.3.3 User-Definable Messages

Messages broadcasted or logged by the Environmental Monitoring utility can be modified. The messages are located in the following file:

/usr/share/sysman/envmon/EnvMon_UserDefinable_Msg.tcl

You must be root to edit this file and 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}
```

Note that you should 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 disk drives by using the diskx command (Section 11.4.6)

- Exercising tape drives by using the `tapex` command (Section 11.4.7)
- Exercising communications systems by using the `cmx` command (Section 11.4.8)

In addition to the exercisers documented in this chapter, your system might also support the DEC Verifier and Exerciser Tool (VET), which provides a similar set of exercisers. Refer to 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 `/usr/field` must be your current directory.

The commands that invoke the system exercisers provide an option for specifying a file where diagnostic output is saved 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 and can cancel them at any time by pressing 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. For exercisers such as `fsx` that need to write to a file system, 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 pressing 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 system error log file, /var/adm/binary.errlog. 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, after reading a log file, you should delete 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 has the difficulty by looking at the /var/adm/binary.errlog file, using either the dia command (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 the intro(2) and sigvec(2) reference pages.

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. For more information, see the fsx(8) reference page.

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*]

You can specify one or more of the following options:

–f*path* Specifies the pathname of the file system directory you want to test. For example, –f/usr or –f/mnt. The default is /usr/field.

–h Displays the command's help message.

–o*file* Saves the output diagnostics in *file*.

–p*num* Specifies the number of fsxr processes you want fsx to initiate. The maximum number of processes is 250. The default is 20.

–t*min* Specifies how many minutes you want the fsx command to exercise the file system. If you do not specify the –t option, the fsx command runs until you terminate it by pressing Ctrl/c in the foreground.

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 4095. The memx command runs 1s and 0s, 0s and 1s, 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

For more information, see the memx(8) reference page

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 of the swap space will be used if all 20 initiated processes (the default) run simultaneously. This would prevent 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*]

You can specify one or more of the following options:

-s Disables the automatic invocation of the shared memory exerciser, shmx.

-h Displays the command's help message.

-m*size* Specifies the amount of memory in bytes for each process you want to test. The default is the total amount of memory divided by 20, with a minimum size of 4095 bytes.

-o*file* Saves the output diagnostics in *file*.

-p*num* Specifies the number of memxr processes to initiate. The maximum number is 20, which is also the default.

-t*min* Specifies how many minutes you want the memx command to exercise the memory. If you do not specify the -t option, the memx command runs until you terminate it by pressing Ctrl/c in the foreground.

The following example of the memx command initiates five memxr processes that test 4095 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 by the –t option is exhausted.

You automatically invoke the shmx exerciser 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*]

The shmx command options are as follows:

–h Prints the command's help message.

–o*file* Saves diagnostic output in *file*.

–v Uses the fork system call instead of the vfork system call to spawn the shmxb process.

-t*time* Specifies *time* as the run time in minutes. The default is to run until the process is killed.

-m*size* Specifies *size* as the memory segment size, in bytes, to be tested by the processes. The *size* value must be greater than zero. The default is the value of the SHMMAX and SHMSEG system parameters, which are set in the /sys/include/sys/param.h file.

-s*n* Specifies *n* as the number of memory segments. The default (and maximum) number of segments is 3.

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 a Disk Drive

Use the diskx command to exercise the disk drives. The main areas that are tested include the following:

- Reads, writes, and seeks
- Performance
- Disktab entry verification

Caution

Some of the tests involve writing to the disk; for this reason, use the exerciser cautiously on disks that contain useful data that the exerciser could overwrite. Tests that write to the disk first check for the existence of file systems on the test partitions and partitions that overlap the test partitions. If a file system is found on these partitions, you are prompted to determine if testing should continue.

You can use the diskx command options to specify the tests that you want performed and to specify the parameters for the tests. For more information, see the diskx(8) reference page.

The diskx command has the following syntax:

diskx [*options*] [*parameters*] -f *devname*

The -f *devname* option specifies the device special file on which to perform testing. The *devname* variable specifies the name of the block or character special file that represents the disk to be tested, such as /dev/disk/dsk1h. The last character of the file name can specify the disk partition to test.

If a partition is not specified, all partitions are tested. For example, if the *devname* variable is /dev/disk/dsk0, all partitions are tested. If the *devname* variable is /dev/disk/dsk0a, the a partition is tested. This parameter must be specified and can be used with all test options.

The following options specify the tests to be run on disk:

–d
: Tests the disk's disktab file entry. The disktab entry is obtained by using the getdiskbyname library routine. This test only works if the specified disk is a character special file. See the disktab(4) reference page for more information.

–h
: Displays a help message describing test options and parameters.

–p
: Specifies a performance test. Read and write transfers are timed to measure device throughput. Data validation is not performed as part of this test. Testing uses a range of transfer sizes if the –F option is not specified.

  The range of transfer sizes is divided by the number specified with the perf_splits parameter to obtain a transfer size increment. For example, if the perf_splits parameter is set to 10, tests are run starting with the minimum transfer size and increasing the transfer size by 1/10th of the range of values for each test repetition. The last transfer size is set to the specified maximum transfer size.

  If you do not specify a number of transfers, the transfer count is set to allow the entire partition to be read or written. In this case, the transfer count varies, depending on the transfer size and the partition size.

  The performance test runs until completed or until interrupted; the time is not limited by the –minutes parameter. This test can take a long time to complete, depending on the test parameters.

  To achieve maximum throughput, specify the –S option to cause sequential transfers. If the –S option is not specified, transfers are done to random locations. This may slow down the observed throughput because of associated head seeks on the device.

–r

Specifies a read-only test. This test reads from the specified partitions. Specify the –n option to run this test on the block special file.

This test is useful for generating system I/O activity. Because it is a read-only test, you can run more than one instance of the exerciser on the same disk.

–w

Specifies a write test. This test verifies that data can be written to the disk and can be read back to verify the data. Seeks are also done as part of this test. This test provides the most comprehensive coverage of disk transfer functions because it uses reads, writes, and seeks. This test also combines sequential and random access patterns.

This test performs the following operations using a range of transfer sizes; a single transfer size is used if the –F option is specified:

- Sequentially writes the entire test partition, unless the number of transfers has been specified using the –num_xfer parameter
- Sequentially reads the test partition

The data read from the disk is examined to verify it. Then, if random transfer testing has not been disabled (using the –S option), writes are issued to random locations on the partition. After the random writes are completed, reads are issued to random locations on the partition. The data read from random locations is examined to verify it.

The following options modify the behavior of the test:

–F

Performs fixed size transfers. If this option is not specified, transfers are done using random sizes. This option can be used with the –p, –r, and –w test options.

–i

Specifies interactive mode. In this mode, you are prompted for various test parameters. Typical parameters include the transfer size and the number of transfers. The following scaling factors are allowed:

- k or K (for kilobyte (1024 * n))
- b or B (block (512 * n))
- m or M (megabyte (1024 * 1024 * n))

For example 10 K would specify 10,240 bytes.

–Q Suppresses performance analysis of read transfers. This option only performs write performance testing. To perform only read testing and to skip the write performance tests, specify the –R option. The –Q option can be used with the –p test option.

–R Opens the disk in read-only mode. This option can be used with all test options.

–S Performs transfers to sequential disk locations. If this option is not specified, transfers are done to random disk locations. This option can be used with the –p, –r, and –w test options.

–T Directs output to the terminal. This option is useful if output is directed to a log file by using the –o option. If you specify the –T option after the –o option, output is directed to both the terminal and the log file. The –T option can be used with all test options.

–Y Does not prompt you to confirm that you want to continue the test if file systems are found when the disk is examined; testing proceeds.

In addition to the options, you can also specify test parameters. You can specify test parameters on the diskx command line or interactively with the –i option. If you do not specify test parameters, default values are used.

To use a parameter, specify the parameter name, a space, and the numeric value. For example, you could specify the following parameter:

```
–perf_min 512
```

You can use the following scaling factors:

- k or K (for kilobyte (1024 * n))
- b or B (for block (512 * n))
- m or M (for megabyte (1024 * 1024 * n))

For example, –perf_min 10K causes transfers to be done in sizes of 10,240 bytes.

You can specify one or more of the following parameters:

–debug Specifies the level of diagnostic output to be produced. The greater the number specified, the more output is produced describing the exerciser operations. This parameter can be used with all test options.

–err_lines	Specifies the maximum number of error messages that are produced as a result of an individual test. A limit on error output prevents a large number of diagnostic messages if persistent errors occur. This parameter can be used with all test options.
–minutes	Specifies the number of minutes to test. This parameter can be used with the –r and –w test options.
–max_xfer	Specifies the maximum transfer size to be performed. If transfers are done using random sizes, the sizes are within the range specified by the –max_xfer and –min_xfer parameters. If fixed size transfers are specified (see the –F option), transfers are done in a size specified by the –min_xfer parameter. Specify transfer sizes to the character special file in multiples of 512 bytes. If the specified transfer size is not an even multiple, the value is rounded down to the nearest 512 bytes. This parameter can be used with the –r and –w test options.
–min_xfer	Specifies the minimum transfer size to be performed. This parameter can be used with the –r and –w test options.
–num_xfer	Specifies the number of transfers to perform before changing the partition that is currently being tested. This parameter is only useful if more than one partition is being tested. If this parameter is not specified, the number of transfers is set to a number that completely covers a partition. This parameter can be used with the –r and –w test options.
–o*filename*	Sends output to the specified file name. The default is to display output on the terminal screen. This parameter can be used with all test options.
–perf_max	Specifies the maximum transfer size to be performed. If transfers are done using random sizes, the sizes

are within the range specified by the –perf_min and –perf_max parameters. If fixed size transfers are specified (see the –F option), transfers are done in a size specified by the –perf_min parameter. This parameter can be used with the –p test option.

–perf_min
: Specifies the minimum transfer size to be performed. This parameter can be used with the –p test option.

–perf_splits
: Specifies how the transfer size will change if you test a range of transfer sizes. The range of transfer sizes is divided by the number specified with the –perf_splits parameter to obtain a transfer size increment. For example, if the –perf_splits parameter is set to 10, tests are run starting with the minimum transfer size and increasing the transfer size by 1/10th of the range of values for each test repetition. The last transfer size is set to the specified maximum transfer size. This parameter can be used with the –p test option.

–perf_xfers
: Specifies the number of transfers to be performed in performance analysis. If this value is not specified, the number of transfers is set equal to the number that is required to read the entire partition. This parameter can be used with the –p test option.

The following example performs read-only testing on the character device special file that /dev/rdisk/dsk0 represents. Because a partition is not specified, the test reads from all partitions. The default range of transfer sizes is used. Output from the exerciser program is displayed on the terminal screen:

```
# diskx -f /dev/rdisk/dsk0 -r
```

The following example runs on the a partition of /dev/disk/dsk0, and program output is logged to the diskx.out file. The program output level is set to 10 and causes additional output to be generated:

```
# diskx -f /dev/disk/dsk0a -o diskx.out -d -debug 10
```

The following example shows that performance tests are run on the a partition of /dev/disk/dsk0, and program output is logged to the diskx.out file. The –S option causes sequential transfers for the best test results. Testing is done over the default range of transfer sizes:

```
# diskx -f /dev/disk/dsk0 -o diskx.out -p -S
```

The following command runs the read test on all partitions of the specified disks. The disk exerciser is invoked as three separate processes, which generate extensive system I/O activity. The command shown in this example can be used to test system stress:

```
# diskx -f /dev/rdisk/dsk0 -r &; diskx -f /dev/rdisk/dsk1
-r &; diskx -f /dev/rdisk/dsk2 -r &
```

11.4.7 Exercising a Tape Drive

Use the `tapex` command to exercise a tape drive. The `tapex` command writes, reads, and validates random data on a tape device from the beginning-of-tape (BOT) to the end-of-tape (EOT). The `tapex` command also performs positioning tests for records and files, and tape transportability tests. For more information, refer to the `tapex`(8) reference page.

Some `tapex` options perform specific tests (for example, an end-of-media (EOM) test). Other options modify the tests, for example, by enabling caching.

The `tapex` command has the following syntax:

tapex [*options*] [*parameters*]

You can specify one or more of the options described in Table 11–4. In addition to options, you can also specify test parameters. You specify parameters on the `tapex` command line or interactively with the `–i` option. If you do not specify test parameters, default values are used.

To use a test parameter, specify the parameter name, a space, and the number value. For example, you could specify the following parameter:

```
–min_rs 512
```

You can use the following scaling factors:

- k or K (for kilobyte (1024 * n))
- b or B (for block (512 * n))
- m or M (for megabyte (1024 * 1024 * n))

For example, 10 K would specify 10,240 bytes.

The following parameters can be used with all tests:

`–err_lines` Specifies the error printout limit.

`–fixed` *bs* Specifies a fixed block device. Record sizes for most devices default to multiples of the blocking factor of the fixed block device as specified by the *bs* argument.

The following parameters can be used with the –a option, which measures performance:

–perf_num	Specifies the number of records to write and read.
–perf_rs	Specifies the size of records.

Other parameters are restricted for use with specific tapex options. Option-specific parameters are documented in Table 11–4.

Table 11–4: The tapex Options and Option Parameters

tapex Flag	Flag and Parameter Descriptions
–a	Specifies the performance measurement test, which calculates the tape transfer bandwidth for writes and reads to the tape by timing data transfers. The following parameters can be used with the –a flag: –perf_num: Specifies the number of records to write and read. –perf_rs: Specifies the size of records.
–b	Causes the write/read tests to run continuously until the process is killed. This flag can be used with the –r and –g flags.
–c	Enables caching on the device, if supported. This flag does not specifically test caching; it enables the use of caching on a tape device while other tests are running.
–C	Disables caching on TMSCP tape devices. If the tape device is a TMSCP unit, then caching is the default mode of test operation. This flag causes the tests to run in noncaching mode.

Table 11–4: The tapex Options and Option Parameters (cont.)

tapex Flag	Flag and Parameter Descriptions
–d	Tests the ability to append records to the media. First, the test writes records to the tape. Then, it repositions itself back one record and appends additional records. Finally, the test does a read verification. This test simulates the behavior of the `tar –r` command. The following parameters can be used with the `–d` flag: `–no_overwrite` — Prevents the append-to-media test from being performed on tape devices that do not support this test. Usually, you use this parameter with the `–E` flag. `–tar_num` — Specifies the number of additional and appended records. `–tar_size` — Specifies the record size for all records written in this test.
–e	Specifies EOM test. First, this test writes data to fill a tape; this action can take some time for long tapes. It then performs reads and writes past the EOM; these actions should fail. Finally, it enables writing past the EOM, writes to the tape, and reads back the records for validation purposes. The following parameters can be used with the `–e` flag: `–end_num` — Specifies the number or records to be written past EOM. (Note that specifying too much data to be written past EOM can cause a reel-to-reel tape to go off line.) `–end_rs` — Specifies the record size.
–E	Runs an extensive series of tests in sequential order. Depending on tape type and CPU type, this series of tests can take up to 10 hours to complete.
–f*device*	Specifies the name of the device special file that corresponds to the tape unit being tested. Refer to Chapter 6 for information on device names. `/dev/tape/tape0_d0` is the default device.

Table 11–4: The tapex Options and Option Parameters (cont.)

tapex Flag	Flag and Parameter Descriptions	
–F	Specifies the file-positioning tests. First, files are written to the tape and verified. Next, every other file on the tape is read. Then, the previously unread files are read by traversing the tape backwards. Finally, random numbers are generated, the tape is positioned to those locations, and the data is verified. Each file uses a different record size. The following parameters can be used with the –F flag:	
	–num_fi	Specifies the number of files.
	–pos_ra	Specifies the number of random repositions.
	–pos_rs	Specifies the record size.
	–rec_fi	Specifies the number of records per file.
–G	Specifies the file-positioning tests on a tape containing data. This flag can be used with the –F flag to run the file position tests on a tape that has been written to by a previous invocation of the –F test. To perform this test, you must use the same test parameters (for example, record size and number of files) that you used when you invoked the –F test to write to the tape. No other data should have been written to the tape since the previous –F test.	
–g	Specifies random record size tests. This test writes records of random sizes. It reads in the tape, specifying a large read size; however, only the amount of data in the randomly sized record should be returned. This test only checks return values; it does not validate record contents. The following parameter is used with the –g flag:	
	–rand_num	Specifies the number of records to write and read.
–h	Displays a help message describing the tape exerciser.	

Table 11–4: The tapex Options and Option Parameters (cont.)

tapex Flag	Flag and Parameter Descriptions
–i	Specifies interactive mode. In this mode, you are prompted for various test parameters. Typical parameters include the record size and the number of records to write. The following scaling factors are allowed: • k or K (for kilobyte (1024 * n)) • b or B (for block (512 * n)) • m or M (for megabyte (1024 * 1024 * n)) For example, 10 K would specify 10,240 bytes.
–j	Specifies the write phase of the tape-transportability tests. This test writes a number of files to the tape and then verifies the tape. After the tape has been successfully verified, it is brought off line, moved to another tape unit, and read in with the –k flag. This test proves that you can write to a tape on one drive and read from a tape on another drive. The –j flag is used with the –k flag. Note the –j flag and the –k flag must use the same parameters. The following parameters can be used with the –j and –k flags: –tran_file: Specifies the number of files to write or read. –tran_rec: Specifies the number of records contained in each file. –tran_rs: Specifies the size of each record.
–k	Specifies the read phase of the tape-transportability tests. This test reads a tape that was written by the –j test and verifies that the expected data is read from the tape. This test proves that you can write to a tape on one drive and read from a tape on another drive. As stated in the description of the –j flag, any parameters specified with the –j flag must be specified with the –k flag. (See the description of the –j flag for information on the parameters that apply to the –j and –k flags.)
–L	Specifies the media loader test. For sequential stack loaders, the media is loaded, written to, and verified. Then, the media is unloaded, and the test is run on the next piece of media. This verifies that all of the media in the input deck can be written to. To run this test in read-only mode, also specify the –w flag.
–l	Specifies the EOF test. This test verifies that a zero byte count is returned when a tape mark is read and that an additional read fetches the first record of the next tape file.

Table 11–4: The tapex Options and Option Parameters (cont.)

tapex Flag	Flag and Parameter Descriptions	
–m	Displays tape contents. This is not a test. This flag reads the tape sequentially and prints out the number of files on the tape, the number of records in each file, and the size of the records within the file. The contents of the tape records are not examined.	
–o filename	Sends output to the specified file name. The default sends output to the terminal screen.	
–p	Runs both the record-positioning and file-positioning tests. For more information, refer to descriptions of the –R and –F flags.	
–q	Specifies the command timeout test. This test verifies that the driver allows enough time for completion of long operations. This test writes files to fill the tape. It then performs a rewind, followed by a forward skip to the last file. This test is successful if the forward skip operation is completed without error.	
–r	Specifies the record size test. A number of records are written to the tape and then verified. This process is repeated over a range of record sizes. The following parameters can be used with the –r flag:	
	–inc	Specifies the record increment factor.
	–max_rs	Specifies the maximum record size.
	–min_rs	Specifies the minimum record size.
	–num_rec	Specifies the number of records.
	–t	Specifies a time limit (in minutes). The default is to run the test until it is complete.

Table 11–4: The tapex Options and Option Parameters (cont.)

tapex Flag	Flag and Parameter Descriptions	
–R	Specifies the record-positioning test. First, records are written to the tape and verified. Next, every other record on the tape is read. Then, the other records are read by traversing the tape backwards. Finally, random numbers are generated; the tape is positioned to those locations, and the data is verified. The following parameters can be used with the -R flag:	
	–pos_num	Specifies the number of records.
	–pos_ra	Specifies the number of random repositions.
	–pos_rs	Specifies the record size.
–s	Specifies the record size behavior test. Verifies that a record that is read returns one record (at most) or the read size, whichever is less. The following parameters can be used with the –s flag:	
	–num_rec	Specifies the number of records.
	–size_rec	Specifies the record size.
–S	Specifies single record size test. This test modifies the record size test (the –r flag) to use a single record size. The following parameters can be used with the -S flag:	
	–inc	Specifies the record increment factor.
	–max_rs	Specifies the maximum record size.
	–min_rs	Specifies the minimum record size.
	–num_rec	Specifies the number of records.
–T	Displays output to the terminal screen. This flag is useful if you want to log output to a file with the –o flag and also have the output displayed on your terminal screen. This flag must be specified after the –o flag in the command line.	
–v	Specifies verbose mode. This flag causes detailed information to be output. For example, it lists the operations the exerciser is performing (such as record counts), and detailed error information. Information provided by this flag can be useful for debugging purposes.	

Table 11–4: The tapex Options and Option Parameters (cont.)

tapex Flag	Flag and Parameter Descriptions
–V	Specifies enhanced verbose mode. This flag causes output of more detailed information than the –v flag. The additional output consists of status information on exerciser operations. Information provided by this flag can be useful for debugging purposes.
–w	Opens the tape as read-only. This mode is useful only for tests that do not write to the media. For example, it allows the –m test to be run on a write-protected media.
–Z	Initializes the read buffer to the nonzero value 0130. This can be useful for debugging purposes. If the –Z flag is not specified, all elements of the read buffer are initialized to zero. Many of the tests first initialize their read buffer and then perform the read operation. After reading a record from the tape, some tests validate that the unused portions of the read buffer remain at the value to which they were initialized. For debugging purposes, you can set this initialized value to a number other than zero. In this case, you can use the arbitrary value 0130.

The following example runs an extensive series of tests on tape device `/dev/tape/tape0_d0` and sends all output to the `tapex.out` file:

```
# tapex –f /dev/tape/tape0_d0 -E -o tapex.out
```

The following example performs random record size tests and outputs information in verbose mode. This test runs on the default tape device `/dev/tape/tape0_d0`, and the output is sent to the terminal screen.

```
# tapex -g -v
```

The following example performs read and write record testing using record sizes in the range 10 K to 20 K. This test runs on the default tape device `/dev/tape/tape0_d0`, and the output is sent to the terminal screen.

```
# tapex -r -min_rs 10k -max_rs 20k
```

The following example performs a series of tests on tape device `/dev/tape/tape0_d0`, which is treated as fixed block device in which record sizes for tests are multiples of the blocking factor 512 KB. The append-to-media test is not performed.

```
# tapex -f /dev/tape/tape0_d0 -fixed 512 -no_overwrite
```

11.4.8 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 press Ctrl/c. For more information, refer to the cmx(8) reference page.

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*]

The cmx command options are as follows:

–h
: Prints the command's help message.

–o *file*
: Saves output diagnostics in *file*.

–t *min*
: Specifies how many minutes you want the cmx command to exercise the communications system. If you do not specify the –t option, the cmx command runs until you terminate it by pressing Ctrl/c in the foreground.

–l *line*
: Specifies the line or lines you want to test. The possible values for *line* are found in the /dev directory and are the last two characters of the tty device name. For example, if you want to test the communications system for devices named tty02, tty03, and tty14, specify 02, 03, and 14, separated by spaces, for the *line* variable. In addition, the line variable can specify a range of lines to test. For example, 00-08.

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 press 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 channnels. Information on managing log files is also included.

The following topics are discussed in this chapter:

- Section 12.1 explains your options for monitoring system events.
- Section 12.2 describes how to set up event monitoring.
- Section 12.3 describes how to recover and read event logs after the system has crashed.
- Section 12.4 explains your options for managing log files.

The Event Manager (EVM) provides an integrated approach to administering system events and errors. Refer to the `EVM`(5) reference page for an introduction to EVM, and see Chapter 13 for information on configuring and using EVM.

12.1 Understanding the Basic Event-Logging Facilities

The operating system uses two basic mechanisms to log system events:

- The system event-logging facility
- The binary event-logging facility

The events detected and recorded by these facilities can be reviewed using the Event Manager (EVM), DECevent, or the error report formatter, `uerf`.

EVM is the recommended method of administering system events. Refer to Chapter 13 for information on configuring EVM. The EVM viewer, `evmviewer`, provides a graphical interface for selecting, filtering, and displaying system events. See the `EVM`(5) and `evmviewer`(8) reference pages for more information.

Refer to the DECevent documentation, and the `uerf`(8) reference page for more information on these utilities.

Note

The uerf command utility will be retired in a future release. You should migrate your event management procedures to EVM as soon as possible.

The log files that the system and binary event-logging facilities create 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. Refer also to the syslog.conf(4) reference page.

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 syslog messages are forwarded from the syslog daemon to EVM, in the form of EVM events. Those syslog messages are posted to the EVM daemon, evmd, by syslogd if the syslogd forwarding function is turned on with the -e option. Events are posted with the EVM name of sys.unix.syslog.*facility*.

Refer to the syslog.auth(4) and syslog_evm.conf(4) reference pages 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 also logged by the syslog function in a less detailed summary 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 the records to a remote system, as specified in the /etc/binlog.conf default configuration file, which is created when you install your system.

The event management utility of choice is EVM in place of DECevent (dia) or the uerf error log reporting facility. You can still examine the binary event log files using the dia command or the uerf command. Both commands translate the records from binary format to ASCII format.

The DECevent utility is an event translation utility that you can use to produce ASCII reports from entries in the system's binary event log files. The DECevent utility is invoked by entering the dia command at the command line. Entering the command without any options will cause DECevent to immediately access and translate the contents of the event log files, displaying the events as shown in Example 12–1. Events will scroll up the terminal screen until you press Ctrl/C.

Example 12–1: Sample Translated Event

```
*************************** ENTRY   48 ********************* [1]
Logging OS                        2. <OS> [2]
System Architecture               2. Alpha
Event sequence number             2.
Timestamp of occurrence              04-MAY-1999 09:50:49 [3]
Host name                            <host name>

System type register      x0000001B  AlphaServer 800 or 1000A
Number of CPUs (mpnum)    x00000001
CPU logging event (mperr) x00000000

Event validity                    1. O/S claims event is valid [4]
Event severity                    5. Low Priority
Entry type                     250. Generic ASCII Info Message Type

SWI Minor class                   9. ASCII Message
SWI Minor sub class               4. Informational

ASCII Message                  Test for EVM connection of binlogd [5]
```

[1] The number of the event in the translated log. Note that the number might be based on the selection or filtering of events.

[2] Identification of the operating system (<OS>), and system architecture.

3 The timestamp (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, a low priority informational message that the EVM utility was testing for a connection to the `binlog` event logger.

5 The actual message logged by the event, which might also have been displayed to a terminal or console at the time the event occurred.

For information about administering the DECevent utility, see the following documentation:

- *DECevent Translation and Reporting Guide*
- `dia`(8)

Section 12.2.1.3 discusses the `/etc/binlog.conf` file.

12.2 Configuring Event Logging

The best method of monitoring system events is to use Event Manager (EVM), as described in Chapter 13. EVM uses the default system and binary event-logging configuration to filter events. You can change this 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 opt to concentrate on certain subsystems, such as mail or print services, and control how and where event messages are logged.

To enable system and binary event-logging, the special files must exist and the event-logging daemons must be running. Refer to Section 12.2.3 and Section 12.2.4 for more information.

As for many other system files, `/var/adm/syslog.dated` and other files in `/var/adm` 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. Refer to 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, edit the `/etc/syslog.conf` or `/etc/binlog.conf` configuration file to specify how the system should log events. In the files, you specify 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; and the destination for messages.

The following sections describe how to edit the configuration files.

12.2.1.1 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 with the syslogd −f *config_file* command.

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 generating 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.

The syslogd 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 /etc/syslog.conf file or to disable an entry.

The facility and severity level are separated from the destination by one or more tabs.

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.

You can specify the following facilities:

Facility	Description
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	Reserved for local use, along with local1 to local7.
mark	Receives a message of priority info every 20 minutes, unless a different interval is specified with the syslogd −m option.
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.

You can specify the following severity levels, which are listed in order of highest to lowest severity:

Severity Level	Description
emerg or panic	A panic condition. You can broadcast these messages to all users.
alert	A condition that you should immediately correct, such as a corrupted system database.
crit	A critical condition, such as a hard device error.
err	Error messages.
warning or warn	Warning messages.
notice	Conditions that are not error conditions, but are handled as special cases.
info	Informational messages.

Severity Level	Description
debug	Messages containing information that is used to debug a program.
none	Disables a specific facility's messages.

You can specify the following message destinations:

Destination	Description
Full pathname	Appends messages to the specified file. You should 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.
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.

You can specify in the /etc/syslog.conf file that the syslogd daemon create daily log files. To create daily log files, use 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*

Where:

- 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 previously 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. The current directory is a link to the latest *date* directory. To get the latest logs, you only need to reference the /var/adm/syslog.dated/current 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 specify a line item similar to the following:

```
mail.info  /var/adm/syslog.dated/mail.log
```

If you specify the previous line item in the /etc/syslog.conf, the syslogd daemon could create the following daily directory and file:

```
/var/adm/syslog.dated/11-Jan-12:10/mail.log
```

12.2.1.2 Configuring syslog to use EVM

By default, syslogd is configured to forward events to EVM with the -e option. (See Section 12.2.4). You can select which syslog events are forwarded to EVM 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 EVM.

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 subscribes to all emergency events from syslog [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 (a part of the system) 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 EVM.

You can select 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 subscription 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 should be 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 EVM name of `sys.unix.syslog.`*facility*.

For more information, refer to `syslog_evm.conf`(4) and Chapter 13.

12.2.1.3 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. Note that EVM subscribes to `binlog` events by default, and any configuration options you select will affect what events are available to EVM. You can filter and select `binlog` events using EVM utilities, as described in Chapter 13.

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 tabs.

You can specify the following event class codes:

Class Code	General
*	All event classes.
dumpfile	Specifies the recovery of the kernel binary event log buffer from a crash dump. A severity level cannot be specified.

Class Code	Hardware-Detected Events
100	CPU machine checks and exceptions
101	Memory
102	Disks
103	Tapes
104	Device controller
105	Adapters
106	Buses
107	Stray interrupts
108	Console events
109	Stack dumps
199	SCSI CAM events

Class Code	Software-Detected Events
201	CI port-to-port-driver events
202	System communications services events

Class Code	Informational ASCII Messages
250	Generic

Class Code	Operational Events
300	Startup ASCII messages
301	Shutdown ASCII messages
302	Panic messages
310	Time stamp
350	Diagnostic status messages
351	Repair and maintenance messages

You can specify the following severity levels:

Severity Level	Description
*	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

You can specify the following destinations:

Destination	Description
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 @) 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.

12.2.2 syslog Security and Remote Messages

Unless the domain host name of a remote host is entered in the local file, /etc/syslog.auth, the local system will not log any syslog messages from that remote host. If you intend to use the secure version of syslogd on a 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 file /etc/syslog.auth using a text editor. This file must be owned by root and have permissions set to 0600.
3. Add the names of any remote hosts that are allowed to forward syslog messages to the local system. Host names must meet the following criteria:

- Each remote host name should appear in a separate line in `/etc/syslog.auth`. (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 `/sys/include/sys/param.h`, although each line in the `/etc/syslog.auth` file is limited to 512 characters.

Note that the `syslogd` daemon's forwarding option (`-e`) is used by default, so remote messages logged to `syslog` are forwarded to EVM.

A plus character (+) by itself allows event forwarding from all hosts. A host name can also be preceded by a minus character (-) to expressly prohibit that host from forwarding events. If the file `/etc/syslog.auth` is not present on the system, then forwarding from all hosts is enabled.

Refer to the `syslog.auth`(4) and `syslogd`(8) reference pages for additional 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

Refer to the `MAKEDEV`(8) reference page for more information.

12.2.4 Starting and Stopping the Event-Logging Daemons

The `syslogd` and `binlogd` daemons are automatically started by the `init` program during system startup. However, you must ensure that the daemons are started. You can also specify options with the command that starts the daemons. Refer to the `init`(8) reference page for more information.

12.2.4.1 The syslogd Daemon

You must ensure that the syslogd daemon is started by the init program. If the syslogd daemon is not started or if you want to specify options with the command that starts the syslogd daemon, you must edit the /sbin/init.d/syslog file and either include or modify the syslogd command line. Note that you can also invoke the command manually.

The command that starts the syslogd daemon has the following syntax:

/usr/sbin/syslogd [-d] [-e] [-s] [-m *mark_interval*] [-p *path*] [-f *config_file*] [-b *rcvbufsz*]

By default the initialization of the daemon uses only one command option, the -e option, which configures the daemon to automatically forward events to EVM. 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
```

Refer to the syslogd(8) reference page for information on the command options.

Note

You must ensure that the /var/adm directory is mounted, or the syslogd daemon will not work correctly.

The syslogd daemon reads messages from the following:

- The domain socket /dev/log file, which is automatically created by the syslogd daemon
- An Internet domain socket, which is specified in the /etc/services file
- The special file /dev/klog, 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 any changes to the /etc/syslog.conf configuration file without shutting down the system:

```
# 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, you must edit the /sbin/init.d/binlog file and either include or modify the binlogd command line. Note that you can also invoke the command manually.

The command that starts the binlogd daemon has the following syntax:

/usr/sbin/binlogd [-d] [-f *config_file*]

Refer to the binlogd(8) reference page 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
- 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`
```

Using the following command, apply any changes to the /etc/binlog.conf configuration file without shutting down the system:

```
# 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 meet 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 kilobytes (8192 bytes) and 1MB (1048576 bytes). Small system configurations, such as workstations, can use a small buffer. Large server systems that use many disks might 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;`

After you modify the /sys/data/binlog_data.c file, you must rebuild and boot the new kernel.

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 is as 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 is as 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, checks 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, checks 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 Maintaining Log Files

On a well maintained system, the size of the various log files should not become a problem as you should carefully select the events that you want to log, monitoring the logs for error conditions that result in many postings, and regularly archiving your important event logs. The /var/spool/cron/crontabs/root file contains some default entries for managing some log files. as follows:

```
0 2 * * 0 /usr/lbin/logclean /var/adm/cron/log > /dev/null
0 2 * * 0 /usr/lbin/logclean /var/adm/wtmp > /dev/null
0 2 * * 0 /usr/lbin/logclean /var/adm/messages > /dev/null
```

You can use the cron daemon to specify that other log files be deleted. However, you should also 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 to clean up the older logs in /var/adm/syslog.dated

```
5 1 * * * find /var/adm/syslog.dated -type d -mtime +5 -exec rm -rf '{}' \;
```

This file entry causes all directories under /var/adm/syslog.dated that were modified more than 5 days ago to be deleted, along with their contents, every day at 1:05. Refer to the crontab(1) reference page for more information.

12.4.1 Startup Log Messages in /var/adm/messages

The size of the message buffer used to store boot-log messages is controlled by the msgbuf_size kernel attribute. The minimum default value for this attribute is 8K, 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 = 32,768, which is equivalent to 32KB.

For large systems with many adapters and devices, the default value might 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 might 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.
3. Set the new Boot Time Value of the msgbuf_size subsystem.
4. Press the Apply button to implement the change, and exit from the dxkerneltuner utility.

You can also use the sysconfig and sysconfigdb commands to implement this change, as described in Chapter 4.

Note

In a cluster, the clu_create command and clu_add_member script invokes the sysconfigdb command to set msgbuf_size to 16K, overriding the automatic scaling algorithm used in standalone systems. Consult the *Cluster Administration* guide for an appropriate cluster setting for `msgbuf_size`.

13

Event Management Using EVM

The Event Manager, EVM, is a comprehensive event management system. In addition to providing traditional event handling facilities, EVM unifies its own events and events from other channels to provide a single source of information, simplifying the task of monitoring system activity. EVM 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:

- Section 13.1 provides an overview of the Event Manager, EVM.
- Section 13.2 describes how to set up and customize EVM.
- Section 13.3 describes how to use EVM to assist in the administration of your system.

13.1 EVM Overview

A critical part of a UNIX system administrator's job is to monitor the state of the system, and to be aware and ready to take action when certain unusual conditions occur, such as when a disk fills or a processor begins reporting 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 described as system **events**.

An event is only 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 might be interesting to the administrator or to some other class of system user. If it is a system event it 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 either be part of the local system or of a remote system.

When a system component has something interesting to report, it makes the information available through an event **channel**. An event channel is any facility used to publish or retrieve event information, and can be a simple log file, an event management system, or a program that can be 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.

UNIX supports a number of channels through which system components can report event and status information, and it is the administrator's responsibility to regularly check the information available at each channel to be sure that the system is operating normally. The UNIX 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 a number of other logs in various locations around the system, each of which might require some level of monitoring. The Event Manager, EVM, 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. EVM's 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 allow events to be filtered, sorted, and formatted in a variety of ways. EVM can also be configured to perform automatic notification of selected conditions.

Rather than replacing the familiar event channels, such as `syslog` and `binlog`, EVM encapsulates them. These channels remain in place, and continue to handle the same set of events as they always did. However, EVM makes the existing channels much more accessible.

EVM provides you with the following features:

- Facilities for users and applications to post and monitor events
- Support for all existing event channels, including `syslog` and `binlog`
- Support for encapsulation of custom event channels
- Integration with DECevent and Compaq Analyze for translation of binary error log events
- Integration of a graphical event viewer with the SysMan application suite
- Choice of summary-line or detailed view of events, including online explanations

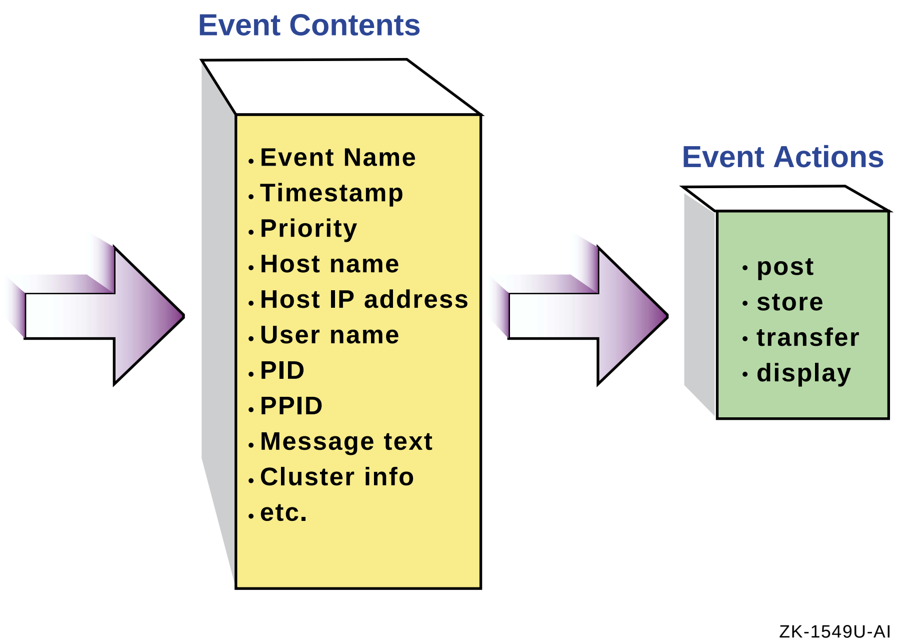
- Full set of command-line utilities for posting and handling events from shell scripts and from the command line
- Configurable event logger that allows full control over which events are logged and optimizes storage space used by identical events
- Configurable event forwarding that allows automatic notification of selected events
- Automatic log file management that performs daily archiving and purging tasks
- Support for the application programming interface (API) library
- Centralized event information
- Configurable authorization for posting or accessing events

13.1.1 Understanding EVM Events

An EVM event is a binary package of data that can contain any or all of a set of standard data items, including a name, a timestamp and information about the poster, and variable data, which is named and supplied by the poster. For example, an event reporting the failure of a device might hold variables containing the pathname and type of the device. Events are typically created and posted by an EVM posting client, and distributed to other clients by the EVM daemon. A process might also create an event and store it in a file, or send it directly to another process, rather than post it. Other processes can then read or receive the event and extract the information it contains.

Although the EVM logger captures posted events and stores them in a system log file, you can easily capture your own set of events using the logger or the `evmwatch` monitoring utility and store them in your own file for later analysis. Figure 13–1 shows a graphical representation of an event.

Figure 13–1: Event Model



In Figure 13–1, the Event Contents box shows some items that might be included in the event, such as the process identifier (PID) and the name of the host system on which the event was generated. The Event Actions box shows some of the actions that can be performed on any event.

Because an event is a package of binary data, you cannot view it directly with a text viewer. However, EVM includes command-line utilities which understand the format of the event, and which can be used to perform basic operations at the UNIX command prompt or in shell scripts. The EVM commands can be used to retrieve events from storage, sort them into any useful order, format them for display, watch for new events being posted, or post new events.

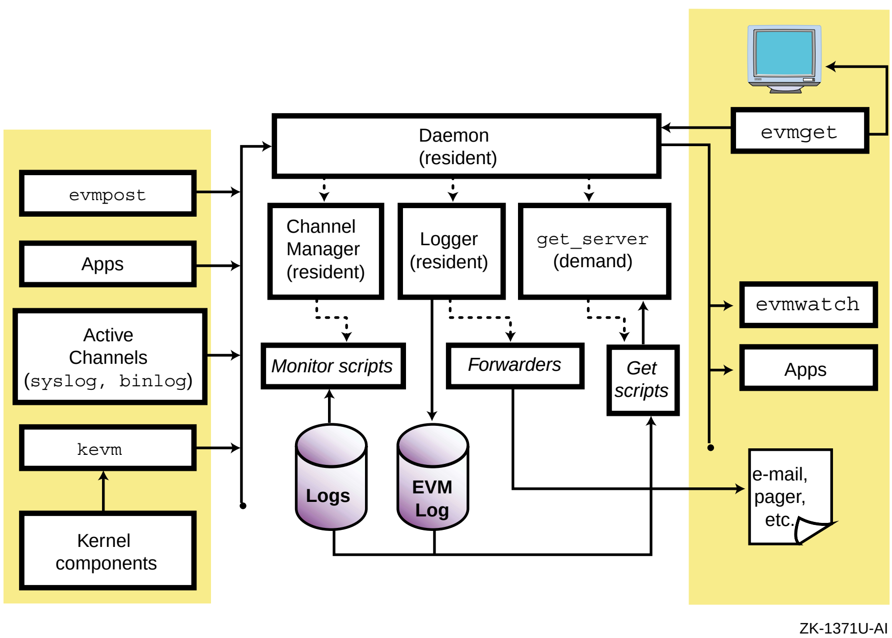
The EVM utilities are designed to be used together in pipelines. For example, you might pipe a set of events from a file into the `evmsort` utility, and pipe the output into the formatting utility, `evmshow`, then pipe the output of that command into `more`, or redirect it to a file.

Once the event file has been converted to text form, you can use other standard utilities to analyze it. For example, you might display just the event names, and then pipe the display into `sort -u` and `wc -l` to determine how many different types of events are in the file.

13.1.2 EVM Components

This section describes how the different parts of EVM interact. It also describes the system files used to run EVM and any files created by EVM during normal operations. A model of the system is shown in Figure 13–2.

Figure 13–2: EVM Component Model



In Figure 13–2, client components involved in posting are shown at the left, EVM system components in the center, and client components involved in subscribing and retrieval are at the right. Active event channels are those that post directly to EVM. 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 EVM is the `evmd` daemon, which is initialized when the system is booted to run level two. For event management to function during system startup, the initialization of the daemon and its child processes is synchronized as follows:

- When the system is booted, some kernel components post events as part of their initialization sequences. Because the EVM daemon is not yet running, these events are queued in kernel memory until the daemon is ready to accept them.
- The EVM daemon is started early in the run level two initialization sequence of system startup. (Refer to 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

- Once the logger has established its listening connection, and is ready to log events, the daemon begins accepting posted events from kernel and user-level posters.

The EVM 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
- Use configurable **forwarding commands** to send mail to the system administrator when high-priority events occur

The resident **channel manager** process, `evmchmgr`, is configured to run periodic channel-monitoring scripts, which might post events. It 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. It is run by the daemon 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. As it receives events from the posters, the daemon merges them with corresponding **event templates** from its template database, and distributes them to its listening clients.

On the right side of the model, `evmwatch` and other application programs that need to receive event information as it happens create **listening 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 can be executed by the logger in response to the occurrence of certain events.

13.1.2.1 EVM Command-Line Utilities

EVM provides a number of command-line utilities both for administering the EVM 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.

Table 13–1: EVM 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 EVM daemon for distribution
evmshow	Accepts one or more EVM 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 administrative commands which are normally invoked during system initialization. The individual command reference pages discuss other conditions under which the command should be used.

Table 13–2: EVM Administrative Utilities

Command	Description
evmchmgr	The EVM channel manager is started automatically by the EVM daemon and should not be run manually. It executes the periodic functions defined for any channel.
evmd	The EVM daemon receives events from posting clients and distributes them to subscribing clients that have indicated that they are interested in receiving them. The daemon is a critical system facility which is started automatically at system startup time and should never be terminated.
evmlogger	The EVM logger is started automatically by the EVM daemon. This command receives events from the daemon and writes them to each of the logs whose filter string they match. This command also serves as an event forwarding agent that can be used to take action when required.
evmreload	This command posts control events, which instruct EVM components to reload their configuration files. When you modify the configuration file you must use this command to load the new configuration into EVM.

Table 13–2: EVM Administrative Utilities (cont.)

Command	Description
`evmstart`	This command starts the EVM daemon. It is intended for use by the system startup scripts, but it also can be used to restart EVM if it has been terminated for any reason.
`evmstop`	This command stops the EVM daemon, preventing events from being posted or subscribed to. It is intended for use by the system shutdown scripts. Because EVM is required to be running for many system functions to operate correctly, this command should not be used under normal circumstances.

13.1.2.2 EVM Application Programming Interface

The EVM API library, `libevm.so`, contains an extensive range of event management functions. This library enables programmers to design programs that interface with EVM. The API functions enable programs to post events, send requests and notifications to the daemon, or receive responses and information from the daemon. Use of these interfaces is described in the *Programmer's Guide*, and the `EVM`(5) reference page lists all the available API functions.

13.1.2.3 EVM System Files

The following system files are created or used by EVM:

- Executable files for EVM administrative commands are located in `/usr/sbin`.
- Command executable files are located in `/usr/bin`.
- Initialization files are located in `/sbin/init.d`.
- Configuration files are located in `/etc` as follows:
  - The `/etc/evmdaemon.conf` file is a configuration file read by the channel manager, `evmchmgr`, and other EVM facilities. Refer to Section 13.2.1.1 and the `evmdaemon.conf`(4) reference page for a complete description of this file.
  - The `/etc/evmchannel.conf` file is the configuration file for the channel manager, `evmchmgr`. It describes all the configured channels from which events can be retrieved. Refer to Section 13.2.1.2 and the `evmchannel.conf`(4) reference page for a complete description of this file.
  - The `/etc/evmlogger.conf` file is the configuration file for the logger, `evmlogger`. It contains commands used to direct the display, forwarding, or storage of events. Refer to Section 13.2.1.3 and the

evmlogger.conf(4) reference page for a complete description of this file.

- The /etc/evm.auth file is used to control access to events and event services. Refer to Section 13.2.2.2 and the evm.auth(4) reference page for a complete description of this file.
- Log files, working files and local installation files are located in the following subdirectories of /var/evm:
  - The /var/evm/sockets CDSL directory contains a domain socket node, evmd, and a related lock file, evmd.lck. Local clients use this socket for connection.
  - The /var/evm/evmlog CDSL directory contains the event logs created by the default EVM 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 is started 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 the first event is received after midnight (00:00 hours).

    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 13.2.3 for more information on managing log files.
  - The /var/evm/adm/logfiles CDSL directory contains output message logs created by the resident components of EVM: the daemon, logger and channel manager. New files are created each time EVM is started, and the old files are renamed by appending a suffix, .old, to their names, overwriting any previous old files. These message logs are encapsulated by EVM's misclog event channel, so their contents are visible through evmget and the event viewer.
  - The /var/evm/shared directory is a work directory used to hold temporary files required for client authentication.
  - The /var/evm/adm/templates 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.
  - The /var/evm/adm/channels directory is provided for installation of local and third-party event channel scripts.

- ☐ The /var/evm/adm/filters directory is provided for installation of local and third-party event filter files.

– The /var/run/evmd.pid file contains the daemon process identifier (PID), saved by evmd for future actions such as stopping EVM.

– The /var/run/evmlogger.info file contains the logger's process identifier, and information about the log files being managed. The evmlog channel retrieval and daily cleanup functions use this information.

– System-supplied definition files for templates, channels, and filters are located in the following subdirectories of /usr/share/evm. These files should not be modified:

  - ☐ The /usr/share/evm/channels 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.
  - ☐ The /usr/share/evm/filters directory contains system filter files.
  - ☐ The /usr/share/evm/templates directory contains system event template files.

13.1.3 Related Utilities

The following existing 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 syslogd forwards selected events to EVM for further storage and distribution. EVM stores syslog events in the evmlog files to reduce the overhead of retrieval from potentially very large text files. Refer to the syslogd(8) reference page 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. In addition to storing events in its own log files and distributing them to its own clients, the binlogd daemon forwards all events to EVM for further distribution. EVM retrieves binary error log events from storage through the binlog event channel functions. Refer to the binlogd(8) reference page 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. EVM uses DECevent's translation facility, `dia`, to translate binary error log events into human-readable form. Refer to the `dia`(8) reference page for more information. Compaq Analyze performs a similar role on some EV6 series processors. Refer to the Compaq Analyze documentation and `ca`(8) reference page for more information.

13.2 Administering EVM

The role of the administrator in running EVM involves the following principal activities:

- Configuring EVM, described in Section 13.2.1
- Controlling who is allowed to post or access events, described in Section 13.2.2
- Managing log files, described in Section 13.2.3
- Providing event reporting facilities for other system users, described in Section 13.2.4
- Installing new products that use EVM capabilities, described in Section 13.2.5

For information on using EVM, see Section 13.3.

13.2.1 Configuring EVM

Configuring EVM means establishing and maintaining its configurable resident components, which are:

- The EVM 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, EVM is automatically configured to run with default configuration options that should be suitable for most installations. However, you should 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, or an alternate logging directory is established.

EVM 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 should be reestablished by running the evmreload command. See the evmreload(8) reference page for information on this command.

Configuration files are described in the following sections and in the corresponding reference pages.

13.2.1.1 EVM Daemon Configuration

The EVM daemon reads its configuration file, /etc/evmdaemon.conf, at system startup and whenever you issue a reload request using the evmreload command. For a complete description of the contents and syntax of the configuration file, see the evmdaemon.conf(4) reference page. Example 13–1 shows some sample entries in a daemon configuration file.

Example 13–1: Sample EVM Daemon Configuration File Entries

```
# Event template directory:
sourcedir "/usr/share/evm/templates" [1]

# Start the EVM Logger                      [2]
start_sync "/usr/sbin/evmlogger -o /var/run/evmlogger.info \
                        -l /var/evm/adm/logfiles/evmlogger.log"
# Start the EVM 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
   }

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 will have completed

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 will be used to make operational details available to the evmlog event channel functions.

3 These statements define the event_get event retrieval service, which is used by the evmget command to retrieve events.

4 These statements define an activity monitor. If 500 or more events are received during any ten minute period, the daemon posts a high-priority event to alert the system administrator. Monitoring (counting) of events is 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 EVM clients. Refer to the evmdaemon.conf(4) reference page and to Section 13.2.2 for information about the security implications of changing this value.

If you make any changes to the configuration file you must run evmreload(8) to make the changes known to the EVM daemon.

13.2.1.2 EVM 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, by the evmshowcommand, and by the event retrieval process. For a complete description of the contents and syntax of the channel configuration file, see the evmchannel.conf(4) reference page. Example 13–2 shows sample channel configuration entries.

Example 13–2: Sample EVM 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        evmlog4
    path        /usr/share/evm/channels/evmlog
    events      *5
    fn_get      "evmlog_get"
    fn_details  "evmlog_details"
```

Example 13–2: Sample EVM Channel Configuration File (cont.)

```
    fn_explain  "evmlog_explain"
    fn_monitor  "evmlog_mon"
    fn_cleanup  "evmlog_cleanup 7 31"6
    mon_period  15:00  # Monitor every 15 minutes7
   }
```

1 This line declares the default path for all channel functions. In this case it is the /usr/share/evm/channels directory. 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 2:00am 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 In this line, the asterisk (*) indicates that the channel provides default event handling, meaning that its functions will be invoked to provide details and explanations for any events whose names do not match the events value of any other channel.

6 This line defines the arguments to this function that specify when the cleanup occurs. In this example, log files older than seven days are compressed and log files older than thirty-one days are deleted.

7 This line sets the monitoring period, causing the /usr/share/evm/channels/evmlog/evmlog_mon function to be invoked every 15 minutes.

13.2.1.3 EVM Logger Configuration

The EVM logger handles storage and forwarding of events, according to entries in its configuration file, /etc/evmlogger.conf. For a complete description of the contents and syntax of this file, see the evmlogger.conf(4) reference page. Example 13–3 shows sample entries in a logger configuration file. An example of possible customization of the logger would be to direct output to a terminal in addition to a log file.

Example 13–3: Sample EVM 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
# Don't forward mail events through mail
   filter  "[prio >= 600] & ![name @SYS_VP@.syslog.mail]"  11

   suppress                                               12
{  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"                                                13
   }
```

1 This line specifies the configuration group that defines an event log.

2 This line specifies that the `evmlog` event channel uses the `name` to get configuration information about the log.

3 This line specifies that the log files are stored in the directory `/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 EVM 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 will close it and begin a new log file, with a sequentially numbered suffix (for example, _2) appended to the file name.

6 If this line is uncommented and the sample path is replaced by the pathname 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. Refer to `EvmFilter`(5) for details of EVM filter syntax.

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. Refer to the `evmlogger.conf`(4) reference page 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 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, mail events are explicitly excluded from this forwarder.

12 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 due to the same event being posted repeatedly. In the example, a particular event is only forwarded once every two hours.

13 This line defines the command that execute 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.

13.2.2 Security Considerations

Security is an important consideration when dealing with events, for the following reasons:

- Uncontrolled access to certain event information might provide an unauthorized user with sensitive information about system operation.
- Posting certain events might cause critical system actions to occur. For example, application failover or system shut down.

Traditionally, security has been maintained over event information by restricting read access to log files and limiting certain posting operations to the `root` user. Because the EVM 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 also be careful to avoid compromising security when writing executable functions to be used in the EVM environment. Refer to the *Programmer's Guide* for more information about protecting channel functions.

13.2.2.1 User Authentication

The EVM daemon authenticates the identities of all local system users before accepting any connection request. There is currently no authentication of remote users. See Section 13.2.2.3 for information about remote connections.

13.2.2.2 User Authorization

Access to events is controlled by the EVM authorization file, `/etc/evm.auth`.

Users can be authorized individually and by group 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. An indicator (+ or -) that is not followed by a user list implicitly grants or 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 the evm.auth(4) reference page for additional details.

Example 13–4: Sample EVM 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 The class of events having names beginning sys.unix.evm.control can be posted only by root, but accessed by all users. The entry @SYS_VP@ is a macro which will be replaced with sys.unix when the file is read.

2 The class of events having names beginning sys.unix.evm.msg.admin can only be posted by root, but might be accessed by root or anybody in the admin group.

3 The class of events having names beginning sys.unix.evm.msg.user can be posted or accessed by all users.

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 changes known to the EVM daemon.

13.2.2.3 Remote Access

Although EVM provides facilities that allow clients to connect to an EVM daemon running on a remote system, it does not currently provide a means of authenticating these clients, and consequently cannot make restricted event information or services available to them. By default, remote access is disabled in the daemon's configuration file, /etc/evmdaemon.conf, and you should only enable it if your system is running in a fully secure environment.

If remote access is enabled, remote clients can connect without authentication, although they are only permitted to take actions that are completely unrestricted in the remote system.

13.2.3 Managing Log Files

The EVM channel manager, evmchmgr, provides log management capability through the channel fn_cleanup function, which can be defined for any channel through the channel configuration file, evmchannel.conf. See Section 13.2.1.2 for additional information on this file.

By default, channel cleanup functions are run when EVM is started, and then at 2:00 A.M. every 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 as follows:

- 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. See Section 13.2.1.2 for a description of this file. Setting either of these values to zero disables the corresponding function. Logs are compressed for archive using the gzip command. See the gzip(1) reference page.

The default channel configuration also provides a similar cleanup function for the SysMan Station message log files, through the misclog event channel. The syslog and binary error log channels can be managed through crontab entries. Because the binary error log file is typically not managed on a daily basis, the channel's cleanup function posts a daily EVM event reporting the size of the log. If the log is growing significantly you should review the log entries and, if necessary, use the cleanup options in `binlogd` to initiate a cleanup. See the `binlogd`(8) reference page for more information.

Note that `evmget` does not retrieve `evmlog` events stored in archived (zipped) logs. To retrieve events from archived logs you must first uncompress them with the `gunzip` command. See the `gunzip`(1) reference page for information on unzipping archive files.

13.2.4 Event Templates

An event template is a centrally held description of an event. The template is used for the following purposes:

- To register the event with the EVM daemon, so that it will allow the event to be 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`. Place any installation-specific or third-party event templates in the local directory `/var/evm/adm/templates`, which is linked into the system directory. Each time an event is posted, the EVM 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.

Template files require specific ownership and permissions to be recognized by EVM. See the `evmtemplate`(4) reference page for details. Refer to the *Programmer's Guide* for details of installation of new event template files.

13.2.5 Installing New EVM 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 might be necessary to modify EVM configuration and authorization files, and to add new templates. See Section 13.2.1 for a discussion of the various configuration files. See Section 13.2.2.2 for information on changing the authorization for new clients.

To add new event templates, create new template files as described in Section 13.2.4 and add them to the `/var/evm/adm/templates` directory or to a subdirectory. Refer to the evmtemplate(4) reference page for details of the required ownership and permissions of a template file.

Refer to the *Programmer's Guide* for additional information about developing EVM client applications.

13.2.6 Configuring binlog Event Translation Utilities

DECevent is a rules-based translation and reporting utility that provides event translation for binary error log events. EVM uses DECevent's translation facility, dia, to translate binary error log events into human-readable form. Some newer processors do not support DECevent and can only support 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 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.

Although the EVM infrastructure directly recognizes events only its own EVM format, an event posted through another channel, such as binlogd, can be passed to EVM by creating a wrapper EVM event and inserting the lower-level event into the EVM event as variable data. The whole package can then be passed to EVM without EVM 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 EVM. When binlogd receives an event from the operating system it first stores the event in its own log file and distributes it to its own clients, in its traditional manner. It then creates an EVM event named sys.unix.binlog, and adds a variable called binlog_event, containing the binlogd event data. Finally, it posts the package to the EVM daemon for further distribution. The EVM daemon deals with the package as it would any EVM 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 the Details... button in the event viewer's event summary window, EVM 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 can be useful if you are 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 ====================
EVM 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 =====================
Logging OS                         2. Digital UNIX
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:
===============================================================
```

EVM 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 for any reason, 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 will 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 might want to use it. 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 wish to use a remote Compaq Analyze server to do translations, you will need to edit the local channel configuration file, as described below.
- Recent processors produce binlogd events with a new header format that differs from the format producted 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 will use DECevent for binlogd event translation, or will use a Compaq Analyze server running on the local system, you do not need to change the standard configuration. If you plan to use a Compaq Analyze server running on a remote system, you will 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 EVM to use DECevent to provide translations if it is available; otherwise EVM will attempt to connect to a Compaq Analyze server running on the local host. If neither of these options is successful, 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 opt to append further -ca items. In the following example, EVM will try 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 has been 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 file you should run the `evmreload -c` command to make the EVM channel manager aware that the file has been updated.

Note that EVM does not start the Compaq Analyze server; it must already be running on the selected system for the translation to succeed. The server is usually started automatically when the system is initialized. For more information, log onto a system that has Compaq Analyze installed and refer to the `desta`(8) reference page.

See Section 13.3.5 for procedures that enable you to test whether either translation utility is available on your system.

13.3 Using EVM in System Administration

The ability of EVM 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 `root` when events with a priority of 600 (alert) or greater are posted, but you should take care to check the full event log on a daily basis, using the event viewer or command-line utilities. You can also configure the logger to take other actions, such as sending a pager message, according to any criteria you choose, and you can monitor events at your terminal as they occur, using the `evmwatch` command.

13.3.1 Monitoring Events Using Commands or Scripts

The `evmwatch` command is a subscribing EVM client you can use to monitor events from the command line or from within a shell script. Example 13–6 shows its use from the command line while Example 13–8 shows its use from within a script.

See the `evmwatch`(1) reference page for a complete description of this command. See the `EvmFilter`(5) reference page for a description of the filter syntax used with the `evmwatch` command. Both `evmwatch` and `evmshow` have numerous options that you will find useful in monitoring your system. Refer to the `evmwatch`(1) and `evmshow`(1) reference pages for details of the command options.

Example 13–6: Monitoring Events from the Command Line Using evmwatch

```
# evmwatch -f '[priority >= 300]' | 1 evmshow \
-t "@timestamp [@priority] @@"2
```

1 This command subscribes for all events with a priority of at least 300, and as each one arrives it is written to stdout and piped to the next command.

2 This command formats the events into human-readable form. In this example the timestamp, priority, and formatted message text is displayed for each event.

You can use the evmwatch -i command option to retrieve a list of all registered events, and pipe the output using the evmshow command to display the event templates in any desired format. For example:

```
# evmwatch -i | evmshow -t "@name [@priority] @format" -x
```

The -i option asks EVM for a list of all event templates; that is, the events that are registered and could be posted. Templates are returned as if they are posted events, and can be piped into the evmshow command for display. The evmshow command 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.

These events have not actually occurred, so the command sequence requested the event's message format, rather than an expanded message. As a result, the summary lines display the messages with names of variables, rather than their values. For example you might see the following summary line:

```
AdvFS: Balance error on AdvFS domain $domain
```

In this example, $domain is a variable that is replaced with the name of the domain when the event is posted.

If you display an event with the evmshow command using @@ in your show template, it provides you with a formatted list of explanations, where available. You can customize the output in many ways as shown in Example 13–7.

Example 13–7: Customizing Formatted Output

```
# evmwatch -i -f "[name sys.unix.fs.advfs]" | \1
   evmsort -s "@priority-" | \2
    evmshow -x -t "@name [@priority] @format"3
```

1 Ask only for AdvFS event templates

2 Sort into priority order (highest first)

3 Display the output using a filter so that only certain events are returned.

Example 13–8 shows how the commands are used within a shell script. The example monitors for a custom event from a remote application, ensuring that the application stays active.

Example 13–8: Monitoring Events from a Shell Script

```
#! /bin/sh
# Monitor a remote application for a heartbeat pulse event

# Pulse due every 5 minutes, so timeout if nothing in 6 minutes
timeout=360

heartbeat_event=myco.myprod.myapp.remote.pulse                       1
evmwatch -f "[ name $heartbeat_event ]" -w $timeout >/dev/null       2
if [ $? -ne 0 ]                                                      3
then
    echo "event { name myco.myprod.myapp.remote.nopulse }" |
         evmpost                                                     4
    exit 1
fi
```

1 This line establishes the name of the pulse event. The event is posted only if its name has been specified in an event template file

2 In this line, the `-f` option identifies the filter used to select events. The `-w` option causes the `evmwatch` command to reset its internal timer when the event is received.

3 This line is an `if` condition that tests the return code from `evmwatch`, taking the `then` action only if the `evmwatch` command should return a non-zero exit value. See the `evmwatch`(1) reference page for more information

4 This line specifies the event to be posted if an error occurs. The event name must have been declared in an event template file. All of the values in the event are taken from the template.

During the monitoring of system operations, users or system administrators might need to post additional events to explain some occurrence, to mark some time point, or for any other reason. The `evmpost` command is used to post events. See the `evmpost`(1) reference page for a description of this command.

13.3.2 Reviewing Event Activity

To assist in the diagnostic processes, EVM provides a graphical event viewer, `evmviewer`, reporting utilities, and a set of command-line utilities. These features are described in the following sections.

13.3.2.1 Viewing Events

Launch the `evmviewer` utility ifrom the View Events option on the SysMan Menu. Application controls available with the viewer provide the following options:

- Selection of the host where event logs are stored.
- Filtering to select or reject events based on one or more of the following:
  - Event name
  - Time span within which events occurred
  - Event priority
  - Posting host
- Advanced filtering capability. See the `EvmFilter`(5) reference page for the available filter options. This capability includes storage and retrieval of custom filter strings.
- Customizing of the event summary display, which is limited to a selection of the event items recognized by the viewer. This might not include all of the items in an event.
- Printing of event summaries and event details.
- Storage of event summaries and event details in a file for later analysis.

See the `sysman`(8) and `evmviewer`(8) reference pages for more information on using these applications. See the online help associated with the event viewer for information on using the viewer options.

13.3.2.2 Generating Reports

Using EVM command-line utilities in conjunction with other commands, you can generate printed reports or create formatted displays to use for problem analysis and resolution, system accounting, or other purposes. You can also create programs using the API functions cited in Section 13.1.2.2 in order to use the EVM capabilities for analysis and accounting purposes.

You can create reports that are based on the event logs archived on any host, providing that security requirements for data access are met. The `evmget` command retrieves the events from various logs and the `evmshow` command formats the event data into readable form. You can also pipe the output of the evmshow command to any other available formatting commands, or store the output in files for additional processing.

Using the evmsort command, you can sort event data using an event data item as a sort key. By default, the evmsort command sorts events in in ascending order by timestamp, that is, the oldest events first. The sort sequence is controlled by the sort specification command-line option. The specification is a character string of the following form:

```
@key_item[+|-] [:@key_item[+|-]
```

The *key_item* is the name of any EVM standard data item. The optional trailing character indicates ascending (+) or descending (-) sort. Multiple items can be specified in the string by separating them with colon characters (:). The left-most key_item in the specification is the most significant key. See the evmshow(1) reference page for more information on this command.

The primary tool for formatting information in a report produced by the evmshow command is the show template. A show template is a string that contains event data-item **specifiers** of the form *@item_name[%width]*, where *item_name* is the name of any standard event data item, such as timestamp or priority and *[%width]* indicates the display field width as a number of characters. If a show template is supplied, evmshow replaces each specifier in the template with the value of the corresponding data item from the event, producing an output string containing only the required items. Any characters in the show template that are not part of an item specifier are displayed unchanged in the output.

The special specifier is represented by two at symbols (@@) and is replaced with the formatted event message text. If an item specified in the show template is not present in the event, a minus sign (-) is output in its place. An at symbol (@) not associated with a valid item name is unchanged in the output. If it is necessary to have an at symbol (@) in the output, use a backslash before the at symbol (\) as an escape sequence. For example; \@compaq.com

You can set the environment variable EVM_SHOW_TEMPLATE to any string to be used as a show template for each iteration of the evmshow command.

For example (assuming ksh syntax):

```
EVM_SHOW_TEMPLATE="@timestamp @@"
```

The show template variable is explained in the evmshow(1) reference page. If you set the EVM_SHOW_TEMPLATEvariable as shown in the preceding example, the default format for any event output from the evmshow command will be the event's timestamp (@timestamp) followed by the formatted event message (@@). Events generally contain a format data item, which is the raw event message, and you can use @format instead of @@. However, in most cases the format includes references to variable values that are also contained in the event; when you specify @@ in the template the evmshow command replaces those references with the values of the variables and a complete message is returned. If you do not specify a template, a default of @@ (with no timestamp) is used.

Once you have the show template defined, you can monitor events from your login terminal in real time, using the following command:

```
# evmwatch | evmshow &
```

You can customize this further by changing the template and by adding a filter to avoid returning events that you do not want to see. For example the following value of the EVM_SHOW_TEMPLATE variable also displays the event priority:

```
EVM_SHOW_TEMPLATE="@timestamp [@priority] @@"
```

You can further control the selected events at the command line. For example, the following command shows events of a priority greater than or equal to 300:

```
# evmwatch -f "[priority >= 300]" | evmshow
```

Using the the -t option with the evmshow command, you can override the default show template and specify any valid template file.

Example 13–9 and Example 13–10 provide samples of of the EVM command-line utilities used with the pipe command (|) to produce reports. These commands normally appear on a single line after the command prompt. In the example, line breaks are indicated by a backslash (\).

Example 13–9: Using EVM Commands to Select Events and Display a Report

```
# evmget -f '[priority >= 300]'   | \          1
   evmsort -s "priority-:timestamp" | \        2
    evmshow -t "@priority  @name  @@" | \      3
      more                                        4
```

1. This command retrieves only those events with a priority of 300 or higher.

2. This command sorts the events with the highest priority event first. Events with the same priority are sorted oldest first.

3. This command formats each event for display, showing the priority, the name, and the formatted message.

4. This command specifies that the events are displayed on the screen. Additional formatting or analysis statements could replace this line.

Example 13–10: Using EVM Commands to Format Events and Print a Report

```
# evmget                               | \      1
   evmsort -s "@name:@timestamp-"    | \    2
    evmshow -t "@name%40  @timestamp" | \ 3
      lpr                                     4
```

1 This command retrieves all events.

2 This command sorts the events by name. Events with the same name are sorted with the most recent events first.

3 This command formats the events for display, showing the name in a column with a minimum width of 40 characters, and the timestamp.

4 This command prints the events on the default printer.

13.3.3 Responding to Event Activity

The response to event activity is a procedural action determined by your site-specific needs and conditions. This response can range from activating alarms or paging responsible personnel, through more mundane reactions such as making a log entry or ignoring an expected occurrence of a regular activity. It is also possible to configure the event processing sequence to perform a series of dependent tasks, using an event output by one task as the trigger to activate the next process. EVM provides an interface to the response activity through the logging facility. The available options are event storage and event forwarding.

The EVM logger, `evmlogger`, started automatically by the EVM daemon, is responsible for the following:

- Storing selected events in one or more log files
- Displaying selected events on the system console or other device
- Forwarding selected events to interested parties in some other form

The logger is an ordinary EVM client that is controlled through a configuration file. The default is `/etc/evmlogger.conf`. See the `evmlogger.conf`(4) reference page for additional information on this file. See the `evmlogger`(8) reference page for additional information on the command.

13.3.3.1 Displaying Detailed Event Information

While many events are posted as part of normal activity, and are simply informational in nature, others will require specific action. In many cases you can get additional information about the event by displaying a detailed view, although note that many event types simply carry text information from another source, and the explanation will be generic.

You can display a detailed view of an event through the event viewer by selecting the event of interest and selecting the “Details...” button. To display details from the command line you will need to isolate the event or events of interest, and then pipe them into the `evmshow -d`command (to show full details of the event) or the `evmshow -x` command (to show just the explanation text). The simplest way to isolate an event is to use its `event-id`; however if the system has been restarted recently the `event-id` may not be unique, and might need to be qualified with the timestamp. Example 13–11 illustrates the commands and the output generated by the commands.

Example 13–11: Displaying an Event Explanation from the Command Line

```
# evmget | evmshow -t "@timestamp @event_id%4 @@"  [1]

...
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
...

# evmget -f '[id = 3] & [time 1999:4:15:*:14:19:06]' |  \ [2]
  evmshow -t "@timestamp @@" -x                         [3]

15-Apr-1999 14:19:06 EVM: Mark event - initial
    The EVM "mark initialization" event is posted by the
    evmlog event channel's "fn_monitor" function when it
    is first started by the EVM channel monitor. This
    event indicates the point at which EVM began to post
    mark events; the monitor function should run
    periodically thereafter to post normal "mark" events.
```

[1] This command retrieves and displays all events, in a format that includes the `event_id` for each event in a column that is four characters wide.

[2] This command retrieves only the event that has an `event_id` of 3, and that was posted at 14:19:06 on April 15th, 1999.

The timestamp information is necessary in order to exclude any other events that might have the same event_id.

3 When run with the -x option, evmshow displays the explanation for the event. Use of the -t option in conjunction with -x displays formatted information before the explanation, which is useful confirmation that you have selected the expected event.

13.3.3.2 Handling Events Automatically

If you want to automate the handling of selected events, you can configure the EVM 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 13.2.1.3 and the evmlogger.conf(4) reference page for more information.

When configuring the logger to forward an event, note that:

- 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 can be converted to text form.
- The logger executes the forwarding command synchronously, meaning that the logger itself cannot continue executing until the command is finished. Forwarding commands should therefore be short-duration commands.
- Event text might 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 should exclude mail events.

13.3.3.3 Logging Events

All events meeting the specifications of an eventlog statement in the configuration file are written to the event log as binary events. See Section 13.1.2.3 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 the evmlogger.conf(4) reference page for the explanation of this criterion.

13.3.3.4 Displaying Events on a Terminal

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 13.2.1.3 for a discussion of the configuration file.)

You can also use the evmshow command to display events retrieved from log files as described in Section 13.3.2.

13.3.3.5 Forwarding Events

Events can be forwarded as specified in the configuration file. (See Section 13.2.1.3 for a discussion of the configuration file.)

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 can be 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 or mailx command, or another command-line mail processor, to send a mail message to a responsible person or paging service
- Invocation of additional software that causes emergency shutdown procedures to commence
- Invocation of some dependent process that is waiting for the event to occur

13.3.4 Posting Quick Messages

The evmpost command provides two quick message formats, one for use by system administrators and one for use by all users. For system administrators the format of the command is:

```
# evmpost [-a msg] [-p priority]
```

For general system users the format is:

```
# evmpost[-u msg] [-p priority]
```

The use of these options results in the posting of an event with the event name `sys.unix.evm.msg.admin` or `sys.unix.evm.msg.user`.

13.3.5 Troubleshooting EVM

If you suspect that EVM is not operating correctly, the first step is to check the message files in `/var/evm/adm/logfiles`. Messages in these files are also displayed through the EVM viewer and `evmget`, as part of the `misclog` event channel.

The following list describes some common problems and the initial steps you should take in trying to resolve such problems:

Kernel events are not being posted.

Check the EVM daemon log file for errors using the following command:

```
# more /var/evm/adm/logfiles/evmdaemon.log
```

Check for the presence of the kernel interface pseudodevice using the following command:

```
# ls -l /dev/kevm
```

If this pseudodevice is not present, create it 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 checking the authorization file using the following command:

```
# more /etc/evm.auth
```

Verify that the event is registered using the following command:

```
# evmwatch -i -f '[name event_name]' | \
  evmshow -t "@name"
```

If the events are still not shown, run `evmreload` and check again. If they are still not visible, check that the template files are correctly installed.

Verify that the subscriber is authorized to access these events, using the following command:

```
#  more /etc/evm.auth
```

Verify that the expected events are actually being posted using the following command:

```
# evmwatch | evmshow -t "@name @@"
```

Run the program that posts the event, and check that the preceding `evmwatch` command displays them correctly.

A posting program is unable to post events.

Verify that the EVM daemon is running, using the following command:

```
# ps -aef | grep evmd
```

Verify that the poster is authorized to post these events by checking the authorization file using the following command:

```
# more /etc/evm.auth
```

Verify that the event is registered 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 check again. If they are still not visible, check that the template files are correctly installed.

Expected `syslog` or `binlog` events are not visible through EVM.

You must either be logged in as root or belong to the `adm` group in order to access `syslog` and `binlog` events.

Check that the `binlogd` and `syslogd` daemons are running, using the `ps` command.

Check 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.

Check the sizes of all log files, particularly the `evmlog` files (`/var/evm/evmlog`), the binary error log (`/var/adm/binary.errlog`) and the SMS daemon log files (`/var/adm/sysman/sysman_station/logs`).

Be sure to use the `-L` option of the `ls` command when looking at file sizes, to make sure that you see the file itself and not a symbolic link or CDSL.

See `binlogd`(8) for details of binary log size management, but note that EVM retrieves events from the archive log file, so starting a new log might not immediately reduce the number of events available to EVM. You can use `cron` to perform a regular `binlog` archive. You can reduce the sizes of the `evmlog` files by changing configuration values in the logger and channel configuration files (`/etc/evmlogger.conf` and `/etc/evmchannel.conf`).

Expected events are not being logged.

Check the event priority. Only events with a priority of 200 or higher are logged by the EVM logger.

Cannot post or subscribe to events through a remote daemon.

Check that remote access is configured and if it is not:

- Set remote_connection to True in the daemon configuration file.
- Run the following command:

```
# evmreload -d
```

(Be sure to consider the security implications of enabling a remote connection.)

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 will tell you if the subset OSFDIABASE*** DECEvent Base Kit (Translation/Analysis) is installed or not. If it is not installed, mount the installation media and use the setld command to install the subset. Refer to the setld(8) reference page.

Check for 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 you do not see the running desta daemon, the Compaq Analyze utility might still be installed but not running or properly configured. To test whether if the utility is installed on your system, look for the binaries 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 using the `setld` command.

13.3.6 Updating Compaq Analyze

You can download the latest version of Compaq Analyze (and other WEBES tools) from the following location: http://www1.service.digital.com/svctools/webes/index.html. Download the kit from the website, saving it to `/var/tmp/webes`. Unpack the kit using a command similar to the following:

```
# tar -xvf <tar file name>
```

Use the following command to install the Compaq Web Based Enterprise Service Suite:

```
# setld -l /var/temp/webes/kit
```

During the installation, you can safely select the default options. However, you might not want to install all the optional WEBES tools. Only Compaq Analyze is used by EVM. Refer to the separate Compaq Analyze documentation and `ca`(8) reference page for more information. Note that recent Alpha EV6 processors are supported only by Compaq Analyze and not DECevent

14
Administering Crash Dumps

This chapter describes how you configure the system crash dump environment and how you save and store crash dumps and their associated data. Crash dumps are a snapshot of the running kernel, taken automatically when the system shuts down unexpectedly. Crash dumps are most often referenced 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 might be familiar with techniques of crash dump analysis and you might want to take and analyze your own dump files.

The following topics are discussed in this chapter:

- Section 14.1 provides an overview of crash dumps.
- Section 14.2 describes how you create a crash dump.
- Section 14.3 describes how you choose the content and method of a crash dump.
- Section 14.4 describes how you take a crash dump manually.
- Section 14.5 describes how you store and archive crash dumps.

14.1 Overview of Crash Dumps

When a system shuts down unexpectedly, it writes all or part of the data in physical memory to swap space on disk (the virtual memory space). Such shut down events are often referred to as system crashes or panics. The stored data and status information is called a crash dump. Crash dumps are different from application-related core dumps, after which the system usually keeps running. After a crash dump, the system is shut down to the console prompt (>>>) and must be rebooted when the problem is identified and resolved.

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 might 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. You must also 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 on-disk virtual memory swap space. Administering swap space is described in Chapter 3. System event managment is described Chapter 12 which describes the `binlogd` and `syslogd` event management channels.

The following documentation contains information on crash dumps or related topics:

- Books:
  - *Installation Guide* – Information on the initial swap space and dump settings configured during installation.
  - *Kernel Debugging* – Information on analyzing crash dumps. Note that you may need to install software development subsets and appropriate licenses to use this feature.
- Reference pages:
  - `savecore`(8) – The program that copies a core dump from swap partitions to a file.
  - `expand_dump`(8) – Produces a noncompressed kernel crash dump file.
  - `dumpsys`(8) – Copies a snapshot of memory to a dump file, without halting the system. This feature 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 also use the graphical interface `/usr/bin/X11/dxkerneltuner` to modify kernel attributes. See the `dxkerneltuner`(8) reference page for information. Online help is also available for this interface. The `dxkerneltuner` interface can also 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 utility if you need to add additional temporary or permanent swap space to produce full dumps.
  - `dbx`(1) – The source level debugger.

14.1.2 Files Used During Crash Dumps

By default, the savecore command copies a crash dump file into /var/adm/crash, although you can redirect crash dumps to any file system that you designate. As for many other system files, /var/adm/crash 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 – 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 noncompressed (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 and /etc/binlog.conf – The logging configuration files

14.2 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 /sbin/savecore script invokes the savecore command. This command moves crash dump information from the swap partitions 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. For information about the command syntax, see the savecore(8) reference page.

14.2.1 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.)

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.) The savecore command stores the kernel image in a file named vmunix.*n*, and by default it stores the (compressed) contents of physical memory in a file named vmzcore.*n*.

The *n* variable specifies the number of the crash. The number of the crash 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.

14.2.2 Crash Dump Logging

Once 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 due to 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 might 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. For more information about how system logging is performed, see the syslogd(8) reference page.

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 was due to 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. For more information about how binary error logging is performed, see the binlogd(8) reference page.

14.2.3 Swap Space

When the system creates a crash dump, 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. Once the system crashes, the information is useless and can be safely overwritten.

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 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 will be 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 all the available swap partitions, the system writes the crash dump to the swap partitions until those partitions are full. It then writes the remaining crash dump information at the end of the primary swap partition, possibly filling that partition.
- 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.

The way that a crash dump is taken can be controlled by several kernel attributes, as follows:

- dump_sp_threshold – Controls the partitions to which the crash dump is written. The default value of 4096 causes the primary swap partition to be used exclusively for crash dumps that are small enough to fit the partition.
- partial_dump – Specifies whether a partial crash dump or a full crash dump is preserved. This attribute is on by default.
- compressed_dump – Specifies whether a dump is compressed to save space. This attribute is on by default.
- 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.
- dump_user_pte_pages – Specifies if you want the system to include user page tables in partial crash dumps. Off by default.

In most cases, compressed dumps will fit on the primary swap partition and you will not find it necessary to modify these variables. 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 14.3.

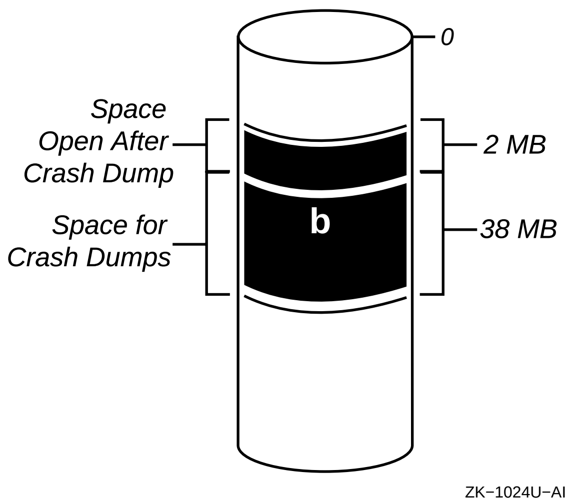
Controlling the Use of Swap Partitions

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 4096 blocks, meaning that the system attempts to leave 2 MB of disk space open in the primary swap partition after the dump is written.

Figure 14–1 shows the default setting of the dump_sp_threshold attribute for a 40 MB swap partition. (Note that 40 MB is atypical of a real swap partition size on most systems, but the principle is the same.)

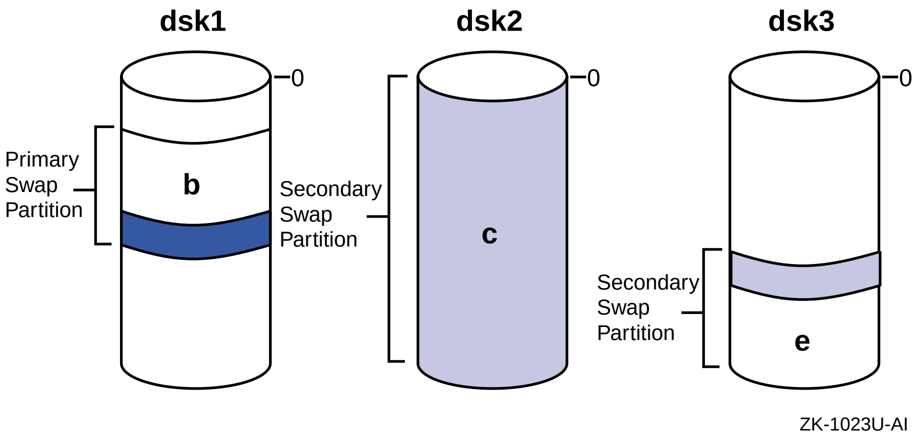
Figure 14–1: Default dump_sp_threshold Attribute Setting



The system can write 38 MB of dump information to the primary swap partition shown in Figure 14–1. 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 14.3.1.). Figure 14–2 illustrates how a crash dump is written to secondary swap partitions on multiple devices.

Figure 14–2: Crash Dump Written to Multiple Devices



If a noncompressed crash dump fills partition `e` in Figure 14–2, the system writes the remaining crash dump information to the end of the primary swap partition. Note that 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 might be corrupted by swapping that occurs as the system reboots.

If a compressed crash dump needs space on the primary swap partition, it always begins 2 MB into the partition, taking what space it requires.

Estimating Crash Dump Size Using dumpsys

To estimate the size of crash dumps, you can use the dump utility `dumpsys`, which produces a run time or continuable dump. See Section 14.4.1 for information on using `dumpsys`. Note that you may have 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, and you can use the `expand_dump` utility to produce a noncompressed crash dumps or use the `-u` option with the `dumpsys` utility.

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 how large the first crash dump file created by your system is, issue 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 would also occupy 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 secondary 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 secondary swap partitions.

14.2.4 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. For information about modifying the size of swap partitions, see the swapon(8) reference page.

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 /etc/sysconfigtab, the savecore command might be unable to find the omitted partition.

The sections that follow give guidelines for estimating the amount of space required for partial and full crash dumps. In addition, setting the dump_sp_threshold attribute is described.

14.2.5 Planning and Allocating File System Space for Crash Dump Files

Using the information on typical crash dump sizes for your system, you can also plan and allocate the file system space that you need for /var/adm/crash.

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 14.5 for information about archiving crash dump files.

By default, savecore 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 the mount(8) reference page.) If you expect your crash dump files to be large, you might need to use a Logical Storage Manager (LSM) file system to store crash dump files. For information about creating LSM file systems, see the *Logical Storage Manager* manual.

If your system cannot save crash dump files due to insufficient disk space, the system returns to single-user mode. This return to single-user mode prevents system swapping from corrupting the crash dump. Once 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
```

Once savecore has saved the crash dump files, you can bring your system to multiuser mode.

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, savecore copies the crash dump to files in the default directory, which is normally /var/adm/crash.

You can control the default location of the crash directory 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
```

14.3 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 might want the system to generate full crash dumps if you have a recurring crash problem and partial crash dumps have not been 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.

14.3.1 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. As described in Section 14.2, the value you assign to this attribute indicates the amount of space that you normally want available for system swapping after a system crash.

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=20480
```

In the preceding example, the `dump_sp_threshold` attribute, which is in the `generic` subsystem, is set to 20,480 512-byte blocks (40 MB). In this example, the system attempts to leave the entire primary swap partition completely 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 using the sysconfigdb command. For information about the sysconfigtab database and the sysconfigdb command, see the sysconfigdb(8) reference page.

Note

Once 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.

14.3.2 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, you may be directed by your technical support person to include user page tables.

If you want the system 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 vm 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 using the sysconfigdb command or use 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).

14.3.3 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
```

To return to partial crash dumps, reset the partial_dump variable to 1.

Note that you can also use dxkerneltuner or sysconfigdb to modify kernel entries and preserve the modifications across reboots.

14.3.4 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 dumpsys, as described in Section 14.4.1. Using the command ls -s, note the size of this dump as value a.
2. Use expand_dump utility to produce a noncompressed version of the dump. Using the command ls -s, note 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 expected_dump_compression.

Set the kernel's expected_dump_compression variable to the required value using sysconfig as follows:

```
# sysconfig -r generic expected_dump_compression=750
expected_dump_compression: reconfigured
# sysconfig -q generic partial_dump
generic:
expected_dump_compression=750
```

Note that you can also use dxkerneltuner or sysconfigdb to modify kernel entries and preseve the modifications across reboots.

14.3.5 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 real-time operations. On reboot after a crash, the crash dump utility, savecore automatically detects that the dump is compressed, using information in the crash dump header in swap. It then copies the crash dump file from swap 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.

This type of compressed crash dump file can be used with some debugging tools such as dbx while still compressed, 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 are always using tools or scripts that will not work with the compressed format, and it is not convenient to use the expand_dump utility. To do this, use the following sysconfig command:

```
# sysconfig -r generic compressed_dump=0
```

Using this method will only temporarily change the mode of dumping to noncompressed and it will revert to compressed on the next reboot. To make a more permanent change, you can use sysconfigdb to update the value of compressed_dump in the /etc/sysconfigtab file or use dxkerneltuner to modify the value in the generic subsystem.

Refer to the reference pages savecore(8), expand_dump(8), and sysconfig(8) for information on crash dump compression and how to produce a noncompressed crash dump file.

14.4 Generating a Crash Dump Manually

The following sections describes how you can manually create a crash dump file under two conditions:

- Continuable dump – Using 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 – Using the console command, crash, to cause a crash dump file to be created on a system that is not responding (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.

14.4.1 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 dumpsys takes a snapshot, memory may be changing as it is copied. Analysis of the resulting dump may show things like 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 `dumpsys` and you may need to halt the system and force a crash dump as described in Section 14.4.2. By default, the `dumpsys` command writes the crash dump in `/var/adm/crash`.

The text file `/var/adm/crash/minfree` specifies the minimum number of kilobytes that must be left on the file system after `dumpsys` 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 `/minfree` using the `-i` option. The `-s` option displays the approximate number of disk blocks that full and partial dumps will 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 taken on a system with 512 KB of physical memory. Note that this is a noncompressed crash dump, and dumps will normally be 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
```

Refer to the `dumpsys`(8) reference page for information on the command options. See *Kernel Debugging* for information on analyzing the continuable crash dump.

14.4.2 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. Press Ctrl/p at the console.

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 Ctrl/p.

14.5 Storing and Archiving Crash Dump Files

If you are working entirely with compressed (`vmzcore.`*n*) crash dump files, they should be sufficiently compressed for efficient archiving. The following sections discuss certain special cases.

Section 14.5.1 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 14.5.2 describes how to uncompress partial crash dump files that have been compressed from `vmcore.`*n* files.

14.5.1 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, on the other hand, uses a special compression method that makes it readable by debuggers and crash analysis tools without requiring decompression. A `vmzcore.`*n* file is substantially compressed compared to the equivalent `vmcore.`*n* file, but not as much as if the latter had been compressed using a standard UNIX compression utility, such as `gzip`. Standard compression applied to a `vmzcore.`*n* file will make 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 using the expand_dump command (see expand_dump(8)). The following example creates an uncompressed file named vmcore.3 from the file vmzcore.3:

   ```
   # 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 a vmzcore.*n* file has been uncompressed into a vmcore.*n* file with expand_dump, you cannot compress it back into a vmzcore.*n* file.

14.5.2 Uncompressing a Partial Crash Dump File

This section applies only if you are uncompressing a partial crash dump file that was 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, that 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 that are invoked as described in Chapter 1.

A.1 The X11 Graphical User Interfaces (CDE Application Manager)

The X11–based graphical utilities (GUIs) are available in the CDE Application Manager or from the command line. In some cases, the GUIs have analogous SysMan utilities or have been superseded by a SysMan Menu task. Invoke these utilities as described in Chapter 1.

You may want to continue using a more familiar utility until you become familiar with the SysMan Menu. You may also have created command shells that perform administrative tasks, using the familiar commands. To help you understand the utilities that you can use, Table A–1 to Table A–6 list the utilities and each table provides the following information:

- The subsystem to be configured and utilities that are invoked from icons in the CDE Application Manager. These may be:
  - X11–compliant Graphical User Interfaces (GUIs), such as the Kernel Tuner.
  - SysMan Menu utilities that can run in a number of different user environments.
  - Command-line scripts. Note that you must install the `OSFRETIREDXXX` subsets to access some of these scripts, as described in the *Installation Guide*.
- 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 invokes the specific utility to configure the local system as a DNS client.
- The commands (command-line options) that perform the equivalent task.

In addition to the icons that invoke the SysMan Menu and SysMan Station and System Setup, 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 monitoring system operation and performance tuning.
- Storage_Management – Table A–5 lists the utilities that are used for administering file systems and storage.
- Tools – Table A–6 lists utilities that provide system statistics.

Table A–1: System Administration Configuration Applications

Subsystem to Configure	SysMan Menu Option	Command-Line Interface
ATM (atmsetup(8))	Configure Asynchronous Transfer Mode (ATM) [atm] atmsetup(8)	atmconfig(8)
CDE Setup (dtsetup(8))	none	none
DHCP Server (xjoin(8), DHCP(7))	Network Setup Wizard	none
DNS (BIND) (bindconfig(8))	Domain Name Service (DNS) (BIND) [dns, dns_client, dns_server, dns_deconfigure]	bindsetup(8)
DOP (dop(8))	Configure Division of Privileges (DOP) [dopconfig]	dopconfig(8)
Disk (diskconfig(8))	Note that some file system tasks performed by diskconfig can be found under the Storage options.	disklabel(8), newfs(8)
Mail (mailconfig(8))	Configure mail [mailsetup]	mailsetup(8)
NFS (nfsconfig(8))	Network File System Configuration (NFS) [nfs]	nfssetup(8)

Table A–1: System Administration Configuration Applications (cont.)

Subsystem to Configure	SysMan Menu Option	Command-Line Interface
NIS (nissetup(8))	Configure Network Information Service (NIS) [nis]	none
NTP	Network Time Protocol (NTP) [ntp, ntp_config, ntp_status, ntp_start, ntpstop]	ntpsetup(8), ntp_manual_setup(7)
Network	Network Setup Wizard [net_wizard]	none
PPP (pppd(8))	Point-to-Point Protocol (PPP) [ppp]	none
Print (printconfig(8))	Configure line printers [lprsetup]	lprsetup(8)
SLIP	Serial Line Networking [serial_line]	slconfig(8)
Audit Configuration	Audit Configuration [auditconfig]	none
Security	Security Configuration [secconfig]	none
latsetup	Configure Local Area Transport (LAT) [lat]	latsetup(8), lat_manual_setup(8)

Table A–2: System Administration Daily Admin Applications

Subsystem to Configure	SysMan Menu Option	Command-Line Interface
Account Manager (dxaccounts(8)). See also features offered under Advanced Server for UNIX (ASU)	Accounts [accounts]. Manage local users, [users]. Manage local groups, [groups]. Manage NIS users, [nis_users]. Manage NIS groups, [nis_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)

Table A–2: System Administration Daily Admin Applications (cont.)

Subsystem to Configure	SysMan Menu Option	Command-Line Interface
Event Viewer	View events [event_viewer]	evmget(1), evmshow(1), evmpost(1), an other associated commands, See EVM(5)
File Sharing dxfileshare(8)	Share Local Directory (etc/exports) [export]	mount(8), automount(8), exports(4)
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
Power Management dxpower(8)	none	sysconfig(8)
SysManShutdown	Shut down the system [shutdown]	shutdown(8)
System Information dxsysinfo(8)	none	du(8), df(8), swapon(8)

Table A–3: System Administration Monitoring and Tuning Applications

Subsystem to Configure	SysMan Menu Option	Command-Line Interface
Class Scheduler	Class Scheduling [class_sched, class_setup, class_start, class_stop]	class_admin(8)
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(8), renice(8), ps(1), kill(1)

Table A–4: System Administration Software Management Applications

Subsystem to Configure	SysMan Menu Option	Command-Line Interface
Software Management	Installation, [install]. List installed software, [setldlist]. Remove installed software, [setldd].	setld(8)
Update Installation Cleanup	Cleanup after an OS update (updadmin), [updadmin]	updadmin(8)

Table A–5: System Administration Storage Management Applications

Subsystem to Configure	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 (dxlsm(8))	Logical Storage Manager (LSM) Utilities [lsm]. Initialize the Logical Storage Manager (LSM) [volsetup].	lsmsetup(4)
Prestoserve I/O Accelerator (dxpresto(8X))	Configure the Prestoserve software [presto]	prestosetup(8))

Table A–6: System Administration Tools

Subsystem to Configure	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 Statistics [vmstat]	vmstat(1)

A.2 The SysMan Menu Tasks and Associated Utilities

The tables listed in Section A.1 identify the GUIs and command-line utilities that perform similar functions to the various SysMan Menu tasks. The following SysMan Menu utilities are available. Note that the accelerator keyword in brackets, such as `[accounts]`, can be used to launch a utility from the command line using the `sysman` command:

- Accounts [accounts] – Enables you to maintain user accounts and access to system resources. Refer to Chapter 7 for information on administering user accounts and the *Network Administration* guide for information on NIS. 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
  - NIS Users [nis_passwd] – Administer NIS user accounts
  - NIS Groups [nis_groups] – Administer NIS user groups
- Hardware [hardware] – Enables you to display information about system hardware and peripheral devices. Refer to Chapter 5 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, its buses and all devices attached to the buses.
  - 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.
  - View central processing unit (CPU) information [hw_cpus] – Displays the type of processors on the system and their status, such as time on line.
- Mail [mail] – Enables you to configure email 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 email for system account holders.
- 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 the EVM(5) reference page and Chapter 13 for more information.
- Set up Insight Manager [imconfig] – Enables you to configure the Insight Manager. Refer to Chapter 1 for more information.
- Class Scheduling [class_sched] – Enables you to allocate CPU time resources to groups of processes. See the `class_admin`(8) reference page 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 also set a current schedule.
  - ☐ [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 the `vmstat`(1) reference page for more information.
- View Input/Output (I/O) Statistics [iostat] – Enables you to monitor I/O (input/output) statistics. See the `iostat`(1) reference page for more information.
- View Uptime statistics [uptime] – Enables you to monitor how long the system has been up, and what the average workload has been since the last boot. See the `uptime`(1) reference page for more information.

• 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. Refer to Chapter 1 for a brief overview. See the *Network Administration* guide 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 Network Interface Card(s) [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. The most common form of communication with local and remote networks. See the *Network Administration* guide for more information.

☐ Set up routing services (gated, routed, or an 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* guide 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 (/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. (Refer to the *Security* guide 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.

☐ Configure asynchronous transfer mode (ATM) [atm] – Set up ATM services and configure ATM adapters. See the *Network Administration* guide for more information.

– 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)) Configuration [dns] – Configure domain name services on the local system. The following utilities are provided:

  ☐ Configure the system as DNS client [dns_client]

  ☐ Configure the system as DNS server [dns_server]

  ☐ 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:

- ☐ Option files [ppp_options]
- ☐ Modify pap-secrets file [pap]
- ☐ Modify chap-secrets file [chap]

See the *Network Administration* guide for more information.

☐ Configure the 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* guide 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* guide for more information. The following utilities are available:

- ☐ Configure system as an NTP client [ntpconfig]
- ☐ View status of NTP Daemon [ntp_status]
- ☐ [Re]start NTP daemon [ntp_start]
- ☐ Stop NTP Daemon [ntp_stop]

– Network File System Configuration (NFS) [nfs] — Enables you to configure Network File System, and share file systems between hosts. See the *Network Administration* guide 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]
- ☐ [Re]start NFS daemons [nfs_start]
- ☐ Stop NFS daemons [nfs_stop]

– Configure Network Information Service Configuration [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* guide for more information.

– Configure Local Area Transport (LAT) [lat] – Enables you to set up LAT. See the *Network Administration* guide for more information.

- View network daemon status [dmnstatus] – Enables you to check the status of the various network daemons such as `gated` or `rwhod`.
- [Re]start 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.

• 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.

• Security [security] – Enables you to administer system security, system auditing, and privileged user access to administrative utilities. See the *Security* guide 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 the *Security* guide and the `dop`(8) reference page for more information.
  - Security Configuration [secconfig] – Enables you to configure base or enhanced security.
  - Audit Configuration [auditconfig] – Enables you to set up and start security auditing.

• Software [software] – Enables you to manage operating system and layered software installations and updates. 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 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).

• 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:
  - ☐ 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 the `advfs`(4) reference page and *AdvFS Administration* guide 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) Utilities [lsm] – Enables you to perform basic administration of Logical Storage Manager (LSM) volumes. See the *Logical Storage Manager* guide for more information. The following utility is provided:
  - ☐ Initialize the Logical Storage Manager (LSM) [volsetup]
- 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.

• 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 Customer Support services. See Chapter 3 and the `sys_check`(8) reference page for information. The following utilities are available:

- Create escalation report [escalation] – Enables you to prepare a system census report for delivery to Technical Support services.
- Create configuration report [config_report] – Enables you to prepare a system census report as a baseline, troubleshooting, or tuning purposes.

- 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 the `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. Refer to the *Guide to Prestoserve* for more information.
  - Configure the 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. Refer to the *Installation Guide — Advanced Topics* 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.

B

Device Mnemonics

This appendix identifies and defines the mnemonics that identify hardware or software devices connected to your system, or identifies a source file where you can look up the supported mnemonics.

Normally, these mnemonics are assigned by default when a system is installed. Others are created automatically by utilities such as `dsfmgr`, which detects new SCSI devices added to a system and creates the appropriate device special files in `/dev`.

If required, you can specify some mnemonics with the `MAKEDEV` command to create the character or block special files that represent each of the devices. You also use the mnemonics to specify device special files for the loadable drivers described in the `/etc/sysconfigtab` configuration database file. Similarly, you can create BSD-style pseudoterminals (ptys). The default SVR4 ptys are created with `SYSV_PTY`.

Table B–1 lists the mnemonics in six categories: generic, consoles, disks, tapes, terminals, and printers. The generic category lists the mnemonics of a general nature and includes memory, null, trace, and tty devices. The consoles category lists mnemonics for the system console devices that the Tru64 UNIX operating system uses. The disks, tapes, terminals, and printers categories identify the appropriate mnemonics for those devices.

The Description column in Table B–1 identifies the corresponding device name. It does not define the mnemonic's use. For detailed information on the use of each mnemonic in relation to the `MAKEDEV` command, the `cfgmgr` configuration manager daemon, and the system configuration file, use the `man` command. For example, enter the following command to display the reference page for the SCSI disk controller driver:

```
# man rz
```

Where appropriate, the reference page defines the device's syntax as it should appear in the `config` file. For additional software device mnemonics that the `MAKEDEV` command uses, refer to the `MAKEDEV`(8) reference page.

Table B–1: Device Mnemonics

Category	Mnemonic	Description
Generic	std	Standard devices with all console subsystems
	atm_cmm	ATM connection management module
	drum	Kernel drum device
	fd0	floppy disk interface
	kmem	Virtual main memory
	lat	Local area transport
	mem	Physical memory
	null	A null device
	sad	STREAMS administrative driver
	trace	A trace device
	tty	A tty device
	local	Customer-specific devices
	zero	Source of zeroes
Prestoserve	nvtc	DEC 3000 Model 300, DEC 3000 Model 400, DEC 3000 Model 500, DEC 3000 Model 600, DEC 3000 Model 800
Consoles	console	System console interface
Disks	dsk*	SCSI disks Refer to the rz(7) and disklabel(8) reference pages and /etc/disktab
	ra*	DSA disks (RA60/RA70/RA71/RA72/RA73/RA80/RA81/RA82/RA90/RA92)
Tapes	tape*	SCSI tapes Refer to the tz(7) reference page and /etc/ddr.dbase
	ta*	DSA tapes (TA78/TA79/TA90/TA91)

Table B–1: Device Mnemonics (cont.)

Category	Mnemonic	Description
Terminals	pty	Network pseudoterminals Refer to /dev
Modems		See the Software Product Description for a list of supported devices and refer to the Owners manual
Printers		See the file lprsetup.dat and refer to the lprsetup.dat(4) reference page

C

Support of the CI and HSC Hardware

The Computer Interconnect (CI) bus is a high-speed, dual-path bus that connects processors and Hierarchical Storage Controllers (HSCs) in a computer room environment. An HSC is an I/O subsystem that is a self-contained, intelligent mass storage controller that provides access to disks and tapes from multiple host nodes attached to the CI bus.

Note

The Tru64 UNIX implementation has the following limitations:

- You can attach a maximum of four HSCs to a CI bus.
- You can attach a single CI bus to a host.
- Under no circumstances can a Tru64 UNIX node participate as a VMS cluster member. A configuration that includes a VMS system and a Tru64 UNIX system residing on the same CI bus is not supported.

Tru64 UNIX supports the System Communication Architecture (SCA) for CI port adapters and HSCs. SCA implements port and class driver support, and standardizes the ways in which TMSCP (tms) and MSCP (ra) devices are handled. SCA separates features into different architectural layers, thus minimizing the effect that software changes to one layer have on other layers.

C.1 Hardware Setup, Restrictions, and Revision Levels

For information on physical components and setup, refer to the HSC hardware documentation and the hardware documentation for your processor and supported devices. Only processors with CI adapters can support HSC configurations.

When setting up the HSC controller hardware, you should attach a terminal to the HSC in order to use commands to get or set HSC parameters, to monitor connections between remote systems, and to identify the disk or tape status.

The maximum number of hosts on a CI bus is 16. The host number for any host on the CI bus must be between 0 and 15.

Note

Two parameters of particular importance are the system ID and the system name. Do not duplicate any system identification or names of nodes on the star coupler.

C.2 Software Installation and Restrictions

The installation software assists you in identifying and configuring the components of your system. You should be familiar with the basic installation guide for your processor before starting the actual installation.

During installation of the Tru64 UNIX software, each accessible MSCP (ra) disk device must be uniquely identified by its unit plug number as follows:

- The unit plug number must be between 0 and 254, inclusive.
- Each unit plug number must be unique. Two different disks cannot have the same unit plug number even if the disks are on separate controllers. For example, if the unit plug number for a disk on controller A is 5 and the unit plug number for a separate disk on controller B is also 5, you must change one of the numbers.
- You can connect a disk with a unique unit plug number to two different controllers (dual or porting). Refer to the `ra`(7) reference page for information on how to specify the device entry in the system configuration file.

C.3 Configuration File Entries

The installation software ensures that your HSC components are configured into the kernel and are included in the `/usr/sys/conf/`*NAME* system configuration file, where *NAME* specifies your system name in uppercase letters.

Chapter 4 provides information on the following entries that correspond to a CI or HSC configuration:

- Description of the `scs_sysid` parameter
- CI adapter specifications
- Controller and device specifications

C.4 Booting an HSC Controller or an HSC Disk

The Tru64 UNIX software supports booting an HSC disk on the DEC 7000 and DEC 10000 processors. If an HSC controller fails, any disks connected to that HSC controller are inaccessible. Attempts to access those disks will cause the accessing system to hang until the HSC reboots completely. Refer to your processor hardware documentation for explicit instructions on booting an HSC disk.

C.5 Sharing Disk and Tape Units Among Several Hosts

Although an HSC can be shared among several hosts, there is no software interlocking mechanism to prevent concurrent write operations to the same partition by multiple Tru64 UNIX systems. The following restrictions must be observed:

- Only multiple readers can share a disk unit; writable file systems cannot be shared.
- If a disk will be shared, it should be hardware write protected.
- Each host must mount the file systems to be accessed with the read-only (`-r`) option to the `mount` command.
- Only a single host can mount a disk that contains writable file systems.
- Use the Network File System (NFS) if multiple writers need to share partitions.

You should coordinate disk unit ownership among the hosts on the CI bus; for example, assign a range of disk unit numbers to each host. The HSC controller can also be directed to limit disk access to an exclusive host system. This limitation protects the disk from accidental access by another host on the CI bus. For more information, see the `radisk`(8) reference page, in particular the `–e` and `–n` options.

Tape drives that are attached to an HSC controller can be shared. This feature is recommended and provides greater use of tape drives. Be aware that the access mechanism provides serial sharing of the drives, not simultaneous access.

D

Administering Specific Hardware

D.1 Introduction

This appendix describes the procedures for adding and configuring certain hardware devices or options as follows:

- PCMCIA cards – This section provides an overview of configuration.
- CalComp graphics tablet – This section provides an overview of configuration.
- Logical partitions on the AlphaServer GS140 – This section provides full configuration instructions.

D.2 PCMCIA Support

Certain processors are able to support PCMCIA (PC Cards) as stated in the Owner's Manual and the *Software Product Description* of a given release of the UNIX operating system. Only a small number of cards have been qualified, but if the card adheres closely to standards it might work.

The steps involved in configuring a PCMCIA card are as follows:

1. Verify that your hardware and operating system support PCMCIA (PC Cards). Consult the information from the adapter vendor and card vendor for any additional configuration steps that are necessary and contact the vendor if you are uncertain.
2. Determine the bus type, which can be ISA or EISA. This step determines the method of console configuration you will use.
3. Install the adapter and configure it using the appropriate console commands.
4. Configure a custom kernel and create the device special files. This step might be unnecessary if an adapter was installed and a card was inserted during initial installation and configuration of the operating system. In this case, the operating system detects the card and creates the kernel configuration entries and device special files. See Chapter 4 for information on kernel configuration and Chapter 5 for information on device special files.
5. Update the `/etc/remote` file.

6. Insert and eject the card as required.

Refer to the pcmcia(7) reference page for specific information on configuring cards and on any current restrictions in a given release. See also the modem(7) reference page and the *Network Administration* guide for information on modem connections.

D.3 CalComp Graphics Tablet

Certain processors are able to support the CalComp DrawingBoard III Tablet as stated in the Owner's Manual and the *Software Product Description* of a given release of the UNIX operating system. Other input devices supported by the Xinput extension to the Xserver might work using a similar configuration. When the software for the tablet is installed on your system, you can configure it to emulate a system mouse.

The steps involved in configuring a tablet are as follows:

1. Ensure that the /usr/var/X11/Xserver.conf file contains a line similar to the following:

   ```
   input <
   <_dec_xi_db3  lib_dec_xi_db3.so  XiDb3Init /dev/tty00:1:12:12:16:\
   1:8:1000:1:1 >
   >
   ```

   The tty that is specified is the serial port (COMM) where the tablet is connected to your system.

2. Specify settings for the tablet in the last line of this file. This line has the following syntax:

   ```
   device:mode:tabletWidth:tabletHeight:numbtns:corePointer:mouseScale:\
   resolution:Xincrement:Yincrement
   ```

   (Refer to the calcomp(7) reference page for an explanation of the data fields.)

3. Connect the tablet to your system and turn it on.

4. Enter the following command to restart the Xserver so that the Xinput extension can recognize the tablet:

   ```
   # /usr/sbin/shutdown -r +5 \
   "Turning on support for the Calcomp Drawingboard III tablet"
   ```

   When the system comes back up, the tablet will be configured into the Xserver and ready to use.

Refer to the calcomp(7) reference page for more information and for restrictions on use.

D.4 AlphaServer GS140 Logical Partitions

A single AlphaServer GS140 system can be divided into a maximum of three logical partitions. Each partition is allocated its own dedicated set of hardware resources. A partition is viewed by the operating system and applications software as a single AlphaServer GS140 system.

Logical partitions employ a *share nothing* model. That is, all hardware resources (processors, memory, and I/O) allocated to a partition are isolated to that partition. A partition's hardware resources can be accessed only by the operating system instance running on that partition.

You can use logical partitions to reduce floor space requirements, power consumption, or improve heat dissipation (by reducing computer room cooling requirements). For example, two departments in an enterprise with different computing requirements might run different applications and require different configuration and tuning of the operating system. Logical partitioning allows you to configure a single AlphaServer GS140 computer to meet the computing needs of both departments.

D.4.1 Hardware Requirements

The hardware requirements for a partition are:

- An AlphaServer GS140 with a minimum of six center plane slots

  At the time this manual was published, only the AlphaServer GS140 6–525 is supported. Refer to the *Systems and Options Catalog* for information on newly-supported systems. The logical partitions feature is supported on the AlphaServer GS140 system. An AlphaServer 8400 (upgraded to a GS140 by replacing the processor modules) is also supported.

- A console device

  This console device can be a character cell video terminal or serial line connection to another system or terminal concentrator. Supported graphics devices can be used by the operating system's windowing software, but not as the console device.

  The restriction of a graphics device to the windowing software (which cannot be the console device) applies only to secondary partitions. A supported graphics device can be the console for the primary partition (partition 0). To use a graphics console, set the **console** environment variable to BOTH before initializing partitions. For example:

  ```
  P00>>> set console BOTH
  ```

  The AlphaServer GS140 includes one console serial port. This port becomes the console for the first partition (partition 0). Each additional partition requires the installation of a KFE72 option. This option includes two serial ports (port 0 is the console port). Refer to the hardware documentation for the KFE72 option information and installation instructions.

- One dual processor CPU module
- One IO Port (IOP) module

  The minimum requirement for a partition is one IOP module. A partition may include a second IOP module. The maximum number of IOP modules for the entire system (the sum of all partitions) is three.
- XMI hardware may be used with logical partitions. However, XMI controllers and devices must be configured into partition zero (0). This is a console firmware restriction.
- One memory module

  The minimum memory size supported for a partition is 512 MB. However, applications running in a partition may require more than the 512 MB minimum memory.
- A software load source device (CD-ROM drive or network adapter)
- A minimum AlphaServer GS140 console firmware revision level of Version 5.4–19

  When installing and configuring logical partitions on a system, refer to the release notes for the operating system release that you are installing, and update the firmware revision if required. Refer to the *Installation Guide* for information on updating the firmware.

The remainder of this section describes the tasks you perform to configure partitions, and provides information about managing a partitioned AlphaServer GS140 system. The topics covered describe the following activities:

- Preparing to install and operate a partitioned system
- Verifying system hardware is properly configured for partitions
- Verifying the revision level of your system's console firmware and upgrade the firmware if necessary
- Configuring partitions for your system by creating the logical partitioning console firmware `environment variables`(EVs)
- Initializing partitions and bootstrap secondary partitions to console mode (the `P##>>>` prompt)
- Installing UNIX and applications software to each partition
- Operating and managing a partitioned system

D.4.2 Preparing to Install and Operate Logical Partitions

You should become familiar with the operation of your system by reading the hardware documentation supplied with your system. Of particular interest for partitioning are the operation of the system's OFF/SECURE/ENABLE/RESET switch and several console commands (such as: `boot`, `create`, `init`, `set`, and `show`).

Before setting up your partitions, make sure the system hardware is fully installed and passes all self-test diagnostics.

Note

Before installing the operating system software to any partition, you should read all subsections of this document. There are certain aspects of managing a partitioned system you must be aware of prior to making the system operational. Precautions must be taken to prevent actions by the console on a partition from interfering with operation of another partition.

The next section describes logical partitioning terms used throughout the rest of this document. After reviewing these terms, proceed to section Section D.4.3.

D.4.2.1 Definition of Commonly Used Terms

You should become familiar with the following terms before configuring your partitions.

logical partition

A logical grouping of hardware resources (CPU, IO, MEMORY, and console) within a single system for exclusive use by an instance of the operating system. A single physical system may have several logical partitions, each running a separate instance of the operating system.

primary partition

Partition number zero. The partition with the only active console terminal when partitioning is disabled (that is, all hardware resources are in one partition).

secondary partition

Partition with a number other greater than zero. One of the partitions that display the console prompt after the `lpinit` command is executed on the primary partition's console.

primary console

The console terminal connected to the primary partition. The only active console terminal when partitions are disabled.

secondary console

The console terminal connected to a secondary partition. Only active when partitions are enabled.

power OFF/ENABLE switch

The four position switch located on the AlphaServer GS140 control panel. The four positions perform the following functions:

- OFF – System power (all partitions) is off.
- SECURE – Power is applied to the system (all partitions). The primary console's `ctrl/p halt` function is disabled.
- ENABLE – Power is applied to the system (all partitions). The primary console's `ctrl/p halt` function is enabled.
- RESET – This is a momentary position. Moving the switch to RESET and then releasing it will cause a complete initialization of the system. All secondary partitions will be immediately terminated. The primary partition will display the normal power on self-test messages and enter console mode.

console prompt

The prompt displayed on the console terminal of a partition to indicate the console firmware is ready to accept commands, which has the following appearance:

```
P##>>>
```

Where ## is the processor number on which the console firmware is currently executing. This is normally the primary processor of the current partition as shown in the following examples:

- For partition 0 with CPU 0:

  ```
  P00>>>
  ```

- For partition 1 with CPU 4:

  ```
  P04>>>
  ```

`ctrl/p halt`

On the primary console terminal only, holding down the control key and typing the letter `p` will cause the primary processor for partition 0 to halt and enter console mode (`P00>>>` prompt). The `halt` operation can be disabled by setting the power switch to the `SECURE` position. The halt operation is ignored on secondary partitions.

P##>>>stop N

Typing `stop` *N* at the console prompt (P##>>>) will cause processor *N* to halt and enter console mode. Issuing this command on the primary console terminal can stop any processor in any partition. For example, if the primary processor for partition 1 is processor 4, the following command will cause processor 4 to enter console mode:

```
P00>>>stop 4
```

P##>>>continue N

If processor `N` entered console mode as the result of a `ctrl/p halt` or `stop N` command, typing `continue N` at the `P##>>>` prompt will cause the processor to resume program execution. Foe example:

```
P##>>>continue 4
```

If only a single processor was halted, then the processor number, `N`, can be omitted.

P##>>>init

Typing `init` at the console (`P##>>>`) prompt of any partition causes a complete reinitialization of the entire system. All active partitions are immediately terminated and the system is reset (as if the power switch was momentarily moved to the `RESET` position). If partitions are enabled, the console will request verification of the `init` command by displaying the following prompt:

```
Do you really want to reset ALL partitions? (Y/<N>)
```

Type `Y` to complete the `init` command or `N` to cancel it.

D.4.3 Logical Partitions Configuration and Installation Tasks

Each of the following sections describes a task you perform to partition your AlphaServer GS140 system. Each task is performed in the order presented, although some tasks may be skipped in certain cases.

If you have read this section previously, and only require a summary of the normal sequence of startup commands, they are as follows:

```
P00>>> set lp_count  n
            (Set the count of n logical partitions)

P00>>> init
            (Initialize the primary partition)

P00>>> lpinit
            (Start the secondary partitions)

P00>>> boot
            (Boot the primary partition)
```

```
P##>>> boot
            (boot the secondary partitions)
```

Improper operation results if the `lpinit` command is omitted. The console firmware prevents this by automatically executing the `lpinit` command if the `lp_count` is nonzero and a boot command is issued on the primary partition's console terminal.

On startup, each secondary partition displays configuration information. It is possible for this message to be proceeded by a series of Y characters as described in Section D.4.3.8. This is not an error and can be ignored.

D.4.3.1 Verifying Your System's Hardware Configuration

You need to verify that your hardware is properly configured for logical partitioning. You also need to record certain information about your hardware configuration for later use (when you configure partitions). Follow these steps to verify your hardware configuration:

1. Power on your system by setting the power `OFF/ENABLE` switch to the `ENABLE` position.

   Note

   A newly installed system (with factory installed software) or an existing system with the `auto_action` console EV set to BOOT or RESTART, will automatically boot the operating system disk after the hardware's self-test is completed.
   In this case, you need to interrupt the automatic boot by typing `ctrl/c` at the console terminal. If the automatic boot could not be interrupted, allow the operating system boot completely, then shut down the operating system (do not type `ctrl/p` to halt the automatic boot). Refer to the *Installation Guide* guide for information on factory installed software before attempting to set up logical partitions.

   The factory installed software disk may be used as the system disk for one of the partitions (see Section D.4.6 for information on installing the operating system).

2. After a short delay (about 15 seconds) configuration information (similar to the following example) will display on the primary console screen:

```
F   E   D   C   B   A   9   8   7   6   5   4   3   2   1   0   NODE #
                            A   A   M   .   M   P   P   P   P   TYP
                            o   o   +   .   +  ++  ++  ++  ++   ST1
                            .   .   .   .   .  EE  EE  EE  EB   BPD
                            o   o   +   .   +  ++  ++  ++  ++   ST2
                            .   .   .   .   .  EE  EE  EE  EB   BPD
```

```
                            +   +   +   .   +  ++  ++  ++  ++   ST3
                            .   .   .   .   .  EE  EE  EE  EB   BPD

                                .   +   +   +   .   +   +   +   C0 PCI +
    .   .   .   .   .   .   +   .   .   +   .   .   +   +       C1 XMI +

.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C4
                +   .   +   +   .   .   .   .   +   .   .   +   C5 PCI +
.   .   .   .   .   .   .   .   .   .   .   .   .   .   .   .   C6
                +   .   +   +   .   +   +   +   .   .   .   +   C7 PCI +
                            .   .   .   +   .   .   .   .          EISA +

                            .   .  A1   .  A0   .   .   .   .   ILV
                            .   . 1GB   . 1GB   .   .   .   .   2GB
Compaq AlphaServer GS140 8-6/525, Console V5.4 15-MAR-99 10:07:33
SROM V1.1, OpenVMS PALcode V1.48-3, Tru64 UNIX PALcode V1.45-3
System Serial =  , OS = UNIX, 12:58:49  March 15, 1999
Configuring I/O adapters...
isp0, slot 0, bus 0, hose0
isp1, slot 1, bus 0, hose0
tulip0, slot 2, bus 0, hose0
isp2, slot 4, bus 0, hose0
isp3, slot 5, bus 0, hose0
tulip1, slot 6, bus 0, hose0
demna0, slot 1, bus 0, xmi0
kzmsa0, slot 2, bus 0, xmi0
kzmsa2, slot 5, bus 0, xmi0
kzpsa0, slot 3, bus 0, hose5
tulip2, slot 8, bus 0, hose5
tulip3, slot 9, bus 0, hose5
pfi0, slot 11, bus 0, hose5
tulip4, slot 12, bus 0, hose7
floppy0, slot 0, bus 1, hose7
kzpsa1, slot 4, bus 0, hose7
tulip5, slot 4, bus 2, hose7
tulip6, slot 5, bus 2, hose7
tulip7, slot 6, bus 2, hose7
tulip8, slot 7, bus 2, hose7
pfi1, slot 6, bus 0, hose7
pfi2, slot 8, bus 0, hose7
kzpsa2, slot 9, bus 0, hose7
P00>>>
```

3. The line ending with `NODE #` indicates the slot number (referred to later in the configuration process). Your system will have up to nine slots, each of which is labelled with its slot number. The next line (ending with `TYP`) indicates the type of module in each slot. Record the type of module in each slot:

```
P = CPU (dual processor CPU module)
M = MEM (memory module)
A = IOP (IO port module)
```

```
  8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+
|   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |
+---+---+---+---+---+---+---+---+---+
```

4. Divide your system into logical partitions by assigning slots (and therefore modules) to each partition. Each partition must be assigned at least one dual CPU module, one MEM module, and one IOP module. With a total of nine slots, the AlphaServer GS140 can be configured for a maximum of three partitions.

> **Note**
>
> Each CPU module has two processors, both of which must be assigned to the same partition.

5. If your system meets the minimum requirements, proceed to the next section. Otherwise, you need to take corrective action (such as installing additional hardware), then proceed to the next section.

D.4.3.2 Verifying the Firmware Revision Level

Logical partitions require console firmware support. Version 5.4 is the minimum firmware revision when this manual was produced, but you should refer to the *Release Notes* for changes to the minimum revision. To verify that your system's firmware includes support for logical partitions, use the following command at the primary console to display the firmware revision level:

```
P00>>>show version
```

The console will display a message similar to the following:

```
version  V5.4,  15-MAR-1999 10:07:33
```

Verify the revision of your firmware is Version 5.4 or later. If you need to upgrade your system's firmware, refer to the firmware upgrade instructions in the hardware documentation. The firmware CD-ROM is shipped with the software kit, or you can download the firmware from the World Wide Web or using `ftp`. The information on finding and updating the firmware is in the *Installation Guide*.

D.4.3.3 Configuring Logical Partitions

You configure and enabled (or disabled) logical partitions using a set of console environment variables (EVs). Two console EVs take the form of hexadecimal numbers, which are bit masks in which a bit position in the mask corresponds to a module or processor number. Hardware configuration rules require modules to be installed in specific slot numbers, based on the module type, as follows:

- IO port (IOP) modules are installed in slots 8, 7, and 6 in descending order with a maximum of three IOP modules allowed.

- CPU (dual processor) modules are installed in slots 0 through N in ascending order (N depends on the number of CPU modules installed). The value of N is limited by the number of IOP and MEM modules.
- MEM (memory) modules are installed in any available slot between the highest numbered CPU module and the lowest numbered IOP module.

The processor mask (lp_cpu_mask) is set by left shifting the number 3 by two times the slot number of the CPU module. Possible CPU masks for each slot are:

```
Processors 00 and 01  (slot 0): 3 << (2 * 0) = 003
Processors 02 and 03  (slot 1): 3 << (2 * 1) = 00c
Processors 04 and 05  (slot 2): 3 << (2 * 2) = 030
Processors 06 and 07  (slot 4): 3 << (2 * 4) = 0c0
Processors 08 and 09  (slot 5): 3 << (2 * 5) = 300
Processors 10 and 11  (slot 6): 3 << (2 * 6) = c00
```

The lp_cpu_mask is formed by combining (logical or) the masks for individual CPU module slots. For example, to assign the four processors on the CPU modules in slot 0 and 1 to partition 0, lp_cpu_mask0 would be 00f.

The IO port mask (lp_io_mask) is set by left shifting the number 1 by the slot number of the IOP module. Possible IOP masks for each slot:

```
IO Port module in slot 8: 1 << 8 = 100
IO Port module in slot 7: 1 << 7 = 080
IO Port module in slot 6: 1 << 6 = 040
```

If a partition has two IOP modules, the lp_io_mask is formed by combining (using logical or) the masks for individual IOP module slots. For example, if you assign IOP modules in slots 7 and 8 to partition 1, the value of lp_io_mask1 will be 180.

When assigning IOP modules to secondary partitions, it is important to remember that one of the IOPs assigned to the partition must be connected to a DWLPB option with a KFE72 option installed. The KFE72 option provides the console serial port for secondary partitions.

D.4.3.4 Determining and Setting Environment Variables

To create the console environment variables for your logical partitions, first determine the number of partitions and which slots (that is, CPU, MEM, and IOP modules) will be assigned to each partition (using the module types and slot numbers you recorded previously). Then, create the console EVs.

A summary of console EVs and values follows:

Console EV	Value
lp_count	Number of partitions
lp_cpu_mask*N*	CPU assignment mask for partition *N*
lp_io_mask*N*	IOP module assignment mask for partition *N*
lp_mem_mode	Memory isolation mode

The following table shows a sample configuration of two partitions based on the configuration information in Section D.4.3.3, with the following modules:

- 4 CPU modules (in slots 0 through 3)
- 2 MEM modules (in slots 4 and 6)
- 2 IOP modules (in slots 7 and 8)

Partition	Module(s)
Partition 0	CPU modules in slots 0 and 1 (CPU 0-3, mask = 00F
	IOP module in slot 8 (IO Port, mask = 100)
	MEM module in slot 6 (2GB memory)
Partition 1	CPU modules in slots 2 and 3 (CPU 4-7, mask = 0F0
	IOP module in slot 7 (IO Port, mask = 080)
	MEM module in slot 4 (1GB memory)

There is no console EV mask for memory. The console firmware assigns memory modules to partitions. The firmware will attempt to balance the amount of memory assigned to each partition.

To create or change the EVs, execute the following commands at the console prompt. The values used are for the two partition example described at the start of this section. The actual values you enter depend on your hardware configuration and your partition layout.

Note that the `lp_count` EV is created with a value of zero (it will be changed later).

The following command will display the console EVs if they have been created or will display no output if they do not exist.

```
P00>>>show lp*
```

If the console EVs do not exist (were not previously created) use the following commands to create the EVs.

Note that there will be a 10 second delay after you issue each command and that the console will display the value of each EV after you create it.

```
P00>>>create -nv lp_count 0
P00>>>create -nv lp_cpu_mask0 f
P00>>>create -nv lp_cpu_mask1 f0
P00>>>create -nv lp_io_mask0 100
P00>>>create -nv lp_io_mask1 80
P00>>>create -nv lp_mem_mode isolate
```

If the console EVs already exist (previously created), use these commands to set their values:

```
P00>>>set lp_count 0
P00>>>set lp_cpu_mask0 f
P00>>>set lp_cpu_mask1 f0
P00>>>set lp_io_mask0 100
P00>>>set lp_io_mask1 80
P00>>>set lp_mem_mode isolate
```

Use the information in the following two sections to display (and if necessary correct) the console EV settings.

D.4.3.5 Displaying Console Environment Variables

The value of a console EV may be displayed on the console of any partition using the `show` command. For example, to display the value of `lp_count` enter the following:

```
P00>>>show lp_count
```

To display all the partitioning EVs, enter the following:

```
P00>>>show lp*
```

If the console EVs are correct, skip the next section and proceed to section Section D.4.3.7 . Otherwise, continue with the next step and make any necessary corrections.

D.4.3.6 Correcting Console Environment Variables

Note

The console EVs with `lp_` prepended to the EV name must be set only by the console of the primary partition (partition 0). Their values must not be changed on any secondary partition.

Use the `set` command to change the value of any or all the console EVs. For example, to change all the EVs, enter the following:

```
P00>>>set lp_count 0
P00>>>set lp_cpu_mask0 f
```

```
P00>>>set lp_cpu_mask1 f0
P00>>>set lp_io_mask0 100
P00>>>set lp_io_mask1 80
P00>>>set lp_mem_mode isolate
```

D.4.3.7 Disabling Automatic Boot Reset

The *Installation Guide* recommends setting the boot_reset console environment variable to ON. This setting is not compatible with logical partitions for which the boot_reset console EV must be set to OFF. This is required so booting a partition does not interfere with the operation of other (previously booted) partitions. If boot_reset is set ON, then a system wide reset is done when the boot command (P00>>>boot) is executed. This reset will immediately terminate operation of all partitions.

Execute the following command to disable boot_reset:

```
P00>>>set boot_reset off
```

D.4.3.8 Set Memory Interleave Mode

The interleave console EV must be set to none. When setting the interleave mode to none, the console may echo a series of Y characters to the console display screen (there could be several lines of Y characters). This is not a problem and can be ignored. This problem will be corrected in a future release of the firmware.

Use the following commands to set the memory interleave mode and proceed to the next section:

```
P00>>>set interleave none
P00>>>init
```

D.4.3.9 Set the Operating System Type to UNIX

Set the os_type console EV to UNIX as follows:

```
P00>>>set os_type UNIX
```

D.4.3.10 Set the auto_action Console Environment Variable

To halt the processor after a POWER-ON or RESET (using the reset switch), use the following command:

```
P00>>>set auto_action halt
```

To automatically boot the operating system after a POWER-ON or RESET, use the following command:

```
P00>>>set auto_action boot
```

D.4.4 Initializing Partitions

Before installing Tru64 UNIX to partitions you need to initialize the partitions. This operation assigns hardware resources (CPU, IOP, and MEM modules) to each partition and spawns a console for each secondary partition as follows:

1. Set the lp_count EV to the number of partitions. For example, to enable two partitions:

   ```
   P00>>>set lp_count 2
   ```

2. Initialize partition 0:

   ```
   P00>>>init
   ```

   Configuration information (as previously described) will display on the primary console screen, followed by the console prompt; P00>>>.

3. Initialize all secondary partitions.

   ```
   P00>>>lpinit
   ```

On the primary console a series of partition configuration messages will be displayed, including the starting address of physical memory for each partition. Record these addresses so you can determine if a kernel rebuild is needed in the event of a memory configuration change.

The following is a sample partition configuration display:

```
Partition 0: Primary CPU = 0
Partition 1: Primary CPU = 4
Partition 0: Memory Base = 000000000   Size = 080000000
Partition 1: Memory Base = 080000000   Size = 040000000
No Shared Memory
LP Configuration Tree = 128000
starting cpu 4 in partition 1 at address 040010001
starting cpu 5 in partition 1 at address 040010001
starting cpu 6 in partition 1 at address 040010001
starting cpu 7 in partition 1 at address 040010001
```

For each secondary partition configured, information will be displayed on the secondary console screens, followed by a console prompt such as P04>>>. Note that there will be a 20–second delay after the lpinit command before the secondary consoles display their configuration information.

D.4.5 Correcting Interleave Mode Errors

If the interleave EV is incorrectly set, the console will display the following error message:

```
Insufficient memory interleave sets to partition system.
Issue command "set interleave none" then reset system.
```

To recover from this error, enter the following commands:

```
P00>>>set interleave none
P00>>>set lp_count 0
P00>>>init
```

Then, repeat the steps in this section.

D.4.6 Installing the Operating System

When partitions are configured and initialized, you can install the operating system to each partition. Install the operating system by following the instructions in the *Installation Guide*.

AlphaServer GS140 systems ship with Tru64 UNIX preinstalled on one of the disks. You can use this disk as the root disk for one of the partitions (usually partition 0). To use the preinstalled disk, boot it and follow the instructions for completing the installation. By default, the `bootdef_dev` console EV should be set to automatically boot the preinstalled disk. If it is not, use the `bootdef_dev` value you recorded in section Section D.4.3.1.

Note

Depending on how you assigned IOP modules, the name of the factory installed software (FIS) disk may change and might not be assigned to partition 0. You can use the following command in each partition to locate the disk:

```
P##>>> show device
```

The operating system can also be installed from a CD-ROM or over the network from a Remote Installation Server (RIS). It may not always be practical to configure a CD-ROM drive on all partitions and a RIS server may not be available. One alternative (assuming a local network is available) is to install the operating system to one partition from a CD-ROM, then configure that partition as a RIS server for the other partitions. Refer to *Sharing Software on a Local Area Network* for instructions on setting up a Remote Installation Server.

D.4.7 Managing a Partitioned System

The operating system running in each partition can be managed as if it were running on a system that is not partitioned. However, there are some AlphaServer GS140-specific operational characteristics that you must be aware of and take into account when managing a partitioned system. These topics are documented in the following sections.

D.4.7.1 Operational Characteristics

During the course of normal partitioned system operations you may need to repeat some of the configuration and initialization tasks. Some of these tasks require special precautions to prevent interference between partitions. The following sections describe these tasks.

D.4.7.1.1 Console init command (P##>>>init)

Typing the `init` command at the console prompt in any partition reinitializes the entire system. This immediately terminates the operating system on all partitions. Therefore, you should not execute the `init` command unless you need to reinitialize the entire system.

If you execute the `init` command, the console will print a message asking you to confirm that you actually want to reset all partitions. Answer `no` to abort the `init` command or `yes` to continue with the `init` command.

D.4.7.1.2 Shutting Down or Rebooting the Operating System

To shut down the operating system running in a partition and return to console mode (`P##>>>` prompt), use the `shutdown` command. For example:

```
# /usr/sbin/shutdown -h +5 "Shutting down the OS"
```

The `shutdown` command can also shut down and reboot the operating system. For example:

```
# /usr/sbin/shutdown -r +5 "Rebooting the OS"
```

D.4.7.2 Recovering an Interrupted Operating System Boot

An incomplete or interrupted operating system boot may leave the console boot drivers in an inconsistent state. In this case, the console will display the following message:

```
Inconsistent boot driver state.
System is configured with multiple partitions.
A complete INIT must be performed before rebooting.
```

Use the following procedure to recover from this condition:

1. Shut down the operating system in all running partitions.
2. Execute the following commands on the primary console:

   ```
   P00>>>set lp_count 0
   P00>>>init
   P00>>>set lp_count N
   ```

   (where `N` is the number of partitions)

```
P00>>>init
P00>>>lpinit
```

3. Boot the operating system in each partition. For example:

```
P00>>>boot
P04>>>boot
```

D.4.7.3 Halting Processors

Under normal operating conditions, it is not necessary to manually halt processors. The processor will halt and enter console mode when the operating system is shut down. However, you will need to manually halt the processor if the operating system hangs for some reason (for example, while debugging a loadable device driver).

Note

In the unlikely event that the processor cannot be halted the system must be reset by momentarily setting the four way OFF/ENABLE switch to the RESET position, then releasing it.

The following procedures only work if the Power OFF/ENABLE switch is in the ENABLE position.

Primary Partition

Pressing Ctrl/p on the primary console terminal will force the primary processor to enter console mode and display the `P##>>>` prompt. You can use the `stop` *N* command (where *N* is a processor number) to stop secondary processors (though this is not normally necessary). See Section D.4.2.1 for definitions of the console prompt and the `stop` command.

Secondary Partitions

Secondary partitions will not halt in response to a Ctrl/p command on the secondary console terminal. To force a secondary partition to enter console mode as follows:

1. Shut down the operating system on the primary partition as follows:

```
# /usr/sbin/shutdown -h +5 "Shutting down the OS"
```

2. Stop the primary processor of the secondary partition.

```
P00>>>stop N
```

   Where *N* is the CPU number of the primary processor of the secondary partitions (normally the lowest numbered CPU assigned to the secondary partition). For example:

```
P00>>>stop 4
```

D.4.7.4 Power OFF/ENABLE Switch Position

During normal system operation, the Power OFF/ENABLE switch should be set to the SECURE position. This will prevent accidentally halting the processor with `ctrl/p`.

D.4.7.5 Reconfiguring Partitions by Changing Console EVs

The console EVs that control logical partitions (names begin with `lp_`) must not be changed on any secondary partition. These console EVs can only be changed by shutting down all partitions and setting new values on the primary partition's console terminal.

Once you have determined the layout of the new partition, follow these steps to reconfigure your partitions:

1. Shut down the operating system in each partition:

   ```
   # /usr/sbin/shutdown -h +5 "Shutting down to
   reconfigure partitions"
   ```

2. Disable partitions and reset the system as follows:

   ```
   P00>>>set lp_count 0
   P00>>>init
   ```

3. Use the console `set` command to change the value of any or all of the console EVs. For the two partition example discussed in Section D.4.3.4, you would use the following commands:

   ```
   P00>>>set lp_count 2
   P00>>>set lp_cpu_mask0 f
   P00>>>set lp_cpu_mask1 f0
   P00>>>set lp_io_mask0 100
   P00>>>set lp_io_mask1 80
   P00>>>set lp_mem_mode isolate
   ```

4. Initialize the primary partition as follows:

   ```
   P00>>>init
   ```

5. Initialize all secondary partitions as follows:

   ```
   P00>>>lpinit
   ```

6. Boot the operating system in each partition using commands similar to the following:

   ```
   P00>>>boot
   P04>>>boot
   ```

D.4.7.6 Checking Other Console EVs Before Booting

Before booting the operating system in each partition, you should use the console `show` command to verify the correct state of the console EVs as follows:

`P0##>>>`**`show boot_reset`**

The `boot_reset` EV must be `off`.

`P0##>>>`**`show interleave`**

The `interleave` EV must be `none`.

`P0##>>>`**`show auto_action`**

The `auto_action` EV can be set to `HALT` or `BOOT`.

`P0##>>>`**`show os_type`**

The `os_type` EV should be set to UNIX.

D.4.7.7 Logical Partitioning Informational Messages at Boot Time

If logical partitions are set up and enabled, several informational messages will be displayed by the operating system on the console terminal for each partition near the beginning of the bootstrap process. The following example shows typical messages for a two partition system:

```
Partition 0
-----------
LP_INFO: 2 partition(s) established via lp_count
LP_INFO: primary processor for partition 0 is CPU 0
LP_INFO: partition 0 CPU allocation mask = 0xf
LP_INFO: partition 0 IOP allocation mask = 0x100
LP_INFO: Memory partitioning mode set to isolate
LP_INFO: partition 0 memory starting address = 0x0

Partition 1
-----------
LP_INFO: 2 partition(s) established via lp_count
LP_INFO: primary processor for partition 1 is CPU 4
LP_INFO: partition 1 CPU allocation mask = 0xf0
LP_INFO: partition 1 IOP allocation mask = 0x80
LP_INFO: Memory partitioning mode set to isolate
LP_INFO: partition 1 memory starting address = 0x80000000
```

These messages provide the following information:

- The number of active partitions
- The number of the primary processor for the current partition
- Which processors are allocated to the current partition
- Which IO port modules are allocated to the current partition
- The memory partitioning mode (should always be set to `isolate`)
- The starting address of memory for the current partition

D.4.8 Hardware Management and Maintenance

For the AlphaServer GS140, partitions share a common physical enclosure and hardware (such as power supplies, system bus, and control panel power switch). The following hardware management and maintenance tasks cannot be performed on individual partitions. You must disable partitions and reset the system to a unpartitioned state.

Tasks that require a complete system reinitialization are:

- Performing corrective or preventive maintenance on system hardware.
- Installing AlphaServer GS140 firmware upgrades, including IO controller firmware upgrades.
- Adding or removing system hardware components (CPUs, memory, IOPs, PCI busses, IO controllers, and IO devices [except for hot swappable disks]).
- Changing any partition's hardware resource assignments by modifying any console EV with `lp_` prepended to its name.
- Running the ECU Eisa Configuration Utility (ECU)or the Raid Configuration Utility (RCU) from the floppy disk drive.

D.4.8.1 Obtaining Technical Support

If you need to escalate a problem to your technical support organization, it is important that you make the Customer Services representative aware that the system is partitioned (particularly when the service is performed via remote diagnosis). When you place the service call, state that your system is using logical partitions.

The logical partitioning software provides two methods for the customer services representative to determine whether or not a system is partitioned. The `LP_INFO` messages printed during operating system startup are also entered into the binary error log as part of the `Startup ASCII Message`. The `sizer -P` command can be run on any instance of the operating system and will display the partitioning status of the system as follows:

```
# sizer -P
   Host hostname is instance 1 of 2 partitions.
   Physical memory starts at address 0x80000000.
   Memory mode is isolate.
   Processors assigned to instance 1: 4  5  6  7
   IO Port (s) assigned to instance 1: slot 7
```

If the system is not partitioned, the following message is displayed, where `hostname` is the name of the system:

```
Host hostname is not partitioned.
```

D.4.8.2 Performing Hardware Management and Maintenance Tasks

Before performing any management or maintenance tasks, you must terminate operation of all partitions and return the system to an unpartitioned state. Use the following steps to shut down partitions:

1. Shut down the operating system in each partition.

   ```
   # /usr/sbin/shutdown -h +5 "Shutting down for maintenance"
   ```

2. Disable partitions by executing the following command at the primary console terminal:

   ```
   P00>>>set lp_count 0
   ```

3. Set the auto_action console EV for the primary partition to HALT as follows:

   ```
   P00>>>set auto_action halt
   ```

   Note that you may need to reset the auto_action EV in step 1 of the next procedure, initializing and rebooting the partitions.

4. Reinitialize the system by typing this command on the primary console terminal.

   ```
   P00>>>init
   ```

When the system returns to the P00>>> prompt you can perform system management and maintenance tasks. After completing system management and maintenance tasks, use the following procedure to reinitialize and reboot your partitions:

1. Verify the console EVs are set to the correct values as follows:

   ```
   P00>>>show lp*
   P00>>>show boot_reset
   P00>>>show interleave
   P00>>>show auto_action
   ```

   The boot_reset EV should be set to off, the interleave EV should be set to none, and the auto_action EV should be set to either HALT or BOOT.

2. Set the lp_count EV to the correct number of partitions. For example:

   ```
   P00>>>set lp_count 2
   ```

3. Initialize the primary partition as follows:

   ```
   P00>>>>init
   ```

4. Initialize all secondary partitions.

   ```
   P00>>>lpinit
   ```

5. Boot the operating system on each partition. If you changed the system's hardware configuration or reassigned any hardware resources to a different partition, a kernel rebuild may be required. Use the procedure in section Section D.4.9 to determine if you need to rebuild the kernel for any partition.

   If a kernel rebuild is not required for a partition, then boot the operating system as follows:

   ```
   P##>>>boot
   ```

   Where ## is the CPU number of the partition's primary processor.

D.4.9 Hardware Changes Requiring a UNIX Kernel Rebuild

A UNIX kernel rebuild may be required when you change your system's hardware configuration. The following table defines the various types of hardware configuration changes and whether or not a kernel rebuild is required:

Change	Requirements
Processors -- adding, removing, or reassigning CPU modules.	Changing the `lp_cpu_mask#` EV for any partition does not require a kernel rebuild. Remember that both processors on a dual CPU module must be assigned to the same partition.
IO Processors -- adding, removing, or reassigning IOP modules.	A kernel rebuild is required if a IOP module is added to or removed from a partition (rebuild the kernel for that partition). Moving a IOP module across partitions requires a kernel rebuild on both partitions. The `lp_io_mask#` EV assigns IOP modules.
	Adding or removing IO busses and IO controllers will require a kernel rebuild for the affected partition.
Memory Modules -- changing the memory module configuration.	For the primary partition (partition 0), changes to the memory module configuration do not require a kernel rebuild.
	The kernel for any secondary partition must be built to run at a specific memory address (that is, the physical memory starting address for the partition). Certain types of memory reconfiguration will change this address and require a kernel rebuild. A partition's memory starting address will change if the memory size for any lower numbered partition increases or decreases.
	For example, if a 2GB memory module in partition zero is replaced by a 4GB memory, then the memory starting address of partition one would increase by 2GB. In this example a kernel rebuild would be required.

If a secondary partition's kernel fails to boot after a memory module configuration change, you should rebuild the kernel.

The memory starting address for each partition is displayed at the primary console after each iteration of the `P00>>>lpinit` command.

D.4.9.1 How to Rebuild the UNIX Kernel for a Partition

The following steps describe how you rebuild the kernel, which is a special case of the typical kernel build instructions documented in Chapter 4. This procedure assumes that partitions are initialized as described in Section D.4.4 and the partition requiring a kernel rebuild is at the `P##>>>` console prompt. Refer to Chapter 4 for information on:

- Kernel booting and the single-user mode prompt.
- Saving and copying kernels.

1. Boot the generic kernel to single-user mode.

   ```
   P##>>>boot -fl s -fi genvmunix
   ```

2. Check and mount file systems.

   ```
   # bcheckrc
   ```

   Refer to Chapter 6 for more information on mounting file systems.

3. Set the host name (system name) for this partition.

   ```
   # hostname NAME
   ```

4. Rebuild the kernel using the `doconfig` command.

   ```
   # doconfig
   ```

   Note

   You must not use `doconfig` with the `-c` option to rebuild the kernel.

5. Save the current kernel as follows:

   ```
   # cp /vmunix /vmunix.save
   ```

6. Install the new kernel as follows, where `SYSNAME` is the local host name:

   ```
   # cp /sys/SYSNAME/vmunix /vmunix
   ```

7. Unmount the file systems as follows:

   ```
   # umount -a
   ```

8. Halt the operating system as follows:

```
# sync
# sync
# halt
```

9. Boot the new kernel as follows:

```
P##>>>boot
```

D.4.10 Handling Nonrecoverable Hardware Error Machine Checks

There are two main classes of hardware errors: recoverable and nonrecoverable. Recoverable errors are corrected by the hardware and reported to the operating system. The operating system logs recoverable errors in the binary error log and continues normal system operation. Non-recoverable hardware errors require immediate termination of normal system operation and some form of corrective action (such as a system reset).

Nonrecoverable hardware errors are reported to the operating system as a machine check. The operating system will crash with a panic message, such as the following:

```
panic (cpu 0): tlaser: \
MACHINE CHECK Non-recoverable hardware error
```

The system will then write out a crash dump, and reboot or halt (depending on the setting of the `auto_action` console EV, which can be BOOT or HALT). Some hardware errors require a complete system reset before the operating system can be rebooted.

For system-wide hardware faults, the operating system will force a system reset after writing the crash dump. After the reset is completed, if `auto_action` is set to BOOT, the console firmware will automatically reinitialize all partitions. Boot the operating system in each partition, using the following commands:

```
P00>>>boot
P##>>>boot
```

Otherwise, the system will halt and enter console mode (`P00>>>` prompt). If this occurs, enter the following commands to restart partitions and reboot the operating system (where *N* is the number of partitions):

```
P00>>>set lp_count N
P00>>>init
P00>>>lpinit
P00>>>boot
```

For each secondary partition, enter the boot command as follows:

```
P##>>>boot
```

For local hardware faults (contained within a partition), the operating system running in the affected partition will unconditionally halt after writing the crash dump. This allows other partitions to continue operating until a shut down can be scheduled. Restarting the affected partition requires a complete system reset, using the following procedure:

1. Shut down the operating system in each running partition as follows:

   ```
   # /usr/sbin/shutdown -h +5 "Shutting down for error recovery"
   ```

2. At the primary console terminal, enter the following commands:

   ```
   P00>>>set lp_count 0
   P00>>>init
   ```

3. The following prompt is printed at the console:

   ```
   Do you really want to reset ALL partitions? (Y/<N>)
   ```

   Type `Y` to perform the reset.

4. After the reset is completed and if `auto_action` is set to BOOT, the console firmware will automatically reinitialize all partitions. Boot the operating system in each partition, using the following commands:

   ```
   P00>>>boot
   P##>>>boot
   ```

   Otherwise, enter the following commands (where *N* is the number of partitions):

   ```
   P00>>>set lp_count N
   P00>>>init
   P00>>>lpinit
   P00>>>boot
   ```

   For each secondary partition enter the following:

   ```
   P##>>>boot
   ```

If these recovery procedures fail to restore full system operation for all partitions, reset the system manually by momentarily moving the OFF/ENABLE switch to the RESET position, then releasing it. After the reset is completed repeat the recovery procedure. If the failure persists, contact your technical support organization.

D.4.11 Logical Partitioning Error Messages

If an error condition occurs (such as an invalid partition configuration) a message is displayed on the partition's console terminal. After displaying the error message, the primary processor for the current partition will halt and return to the console prompt. To recover from any of these errors, correct the logical partitioning console EVs and reboot the partition.

The following error messages might be displayed:

```
LP_ERROR: invalid partition count (lp_count = #, max nodes
= #)
```

The `lp_count` console EV is set incorrectly. The value is less than zero or exceeds the maximum number of partitions supported for the AlphaServer GS140.

```
LP_ERROR: no CPUs for partition (check lp_cpu_mask)
```

The value of `lp_cpu_mask#` (# represents the current partition number) is set incorrectly. There are no processors allocated to this partition.

```
LP_ERROR: no IOP for partition (check lp_io_mask)
```

The value of `lp_io_mask#` (# represents the current partition number) is set incorrectly. There are no IO Port modules allocated to this partition.

```
LP_ERROR: lp_count > 1, but partitions not initialized
Please execute 'lpinit' command at >>> prompt
```

The message indicates that partitions were configured, but not initialized.

```
LP_ERROR: must set lp_mem_mode [share or isolate]
```

The `lp_mem_mode` console EV is not set or set incorrectly. For logical partitions, `lp_mem_mode` must be set to *isolate*.

```
Bootstrap address collision, image loading aborted
```

The kernel's link address does not match the memory starting address of the partition. Refer to section Section D.4.9 for instructions on how to recover from this error.

D.4.12 Understanding Console Firmware Error or Informational Messages

The console firmware implements several safety checks during certain events (such as system reset and partition startup). These checks help prevent cross-partition interference. If an anomaly is detected, one of the following messages will be displayed on the partition's console:

```
Do you really want to reset ALL partitions?  (Y/<N>)
```

This message displays when a system reset has been requested, either by the operation issuing the init command or as a result of booting with the boot_reset console EV set to ON. This message is a warning that if you continue with the reset it will immediately terminate all partitions and completely reset the system. If a reset is necessary, shut down the operating system in all operational partitions before proceeding with the reset.

```
Auto-Starting secondary partitions...
```

This message indicates the console firmware is automatically initializing logical partitions, that is, it automatically executed the lpinit command. Auto starting occurs after a system reset (or power on). The console firmware will also boot the operating system in all partitions if the auto_action console EV is set to BOOT and the reset was performed via the RESET switch on power-on (not through the init command).

```
Insufficient memory interleave sets to partition system.
Issue command "set interleave none" then reset system.
```

This message indicates that the interleave console EV is incorrectly set. You need to change the setting to none.

```
Insufficient memory modules to partition system.
```

Each partition requires a dedicated memory module. You need to reduce the number of partitions or install a memory module for each partition.

This message could indicate the lp_count console EV is not set correctly. For example, you have two partitions, but lp_count is set to four. In this case, set lp_count to match the actual number of partitions.

```
Inconsistent boot driver state.
System is configured with multiple partitions.
A complete INIT must be performed before rebooting.
```

An incomplete or interrupted operating system boot has caused the console boot drivers to enter a inconsistent state. Refer to Section D.4.7.2 for instructions on recovering from this state.

```
Do you want to attempt to boot secondary partitions anyway?
(Y/<N>).
```

This message indicates that the console detected an inconsistency in your partitions set up (probably due to incorrect setting of lp_ console

EVs). Unless you are certain it is safe to proceed, you should answer no (N) to this question and correct the inconsistency.

```
TIOP # not configured in any partition.
Non-existent TIOP # configured in a partition.
```

These messages (together or separately) indicate incorrect setting of the lp_io_mask# console EV. The mask may be set to zero or to the wrong IOP module slot number. You should correct the setting and retry the lpinit command.

```
Secondary partitions have already been started.
```

This message most likely indicates you issued a second lpinit command after starting partitions. Before booting the operating system, you should check the values of the lp_ console EVs.

```
CPU # not configured in any partition.
No valid primary processor specified for partition #.
```

In this message, the CPU number (#) may be a single CPU or a list of CPUs.

These messages (together or separately) indicate incorrect setting of the lp_cpu_mask# console EV. The mask may be set to zero or to incorrect CPU numbers. You should correct the setting and retry the lpinit command.

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

How to Order Tru64 UNIX Documentation

You can order documentation for the Tru64 UNIX operating system and related products at the following Web site:

`http://www.businesslink.digital.com/`

If you need help deciding which documentation best meets your needs, see the Tru64 UNIX *Documentation Overview* or call **800-344-4825** in the United States and Canada. In Puerto Rico, call **787-781-0505**. In other countries, contact your local Compaq subsidiary.

If you have access to Compaq's intranet, you can place an order at the following Web site:

`http://asmorder.nqo.dec.com/`

The following table provides the order numbers for the Tru64 UNIX operating system documentation kits. For additional information about ordering this and related documentation, see the *Documentation Overview* or contact Compaq.

Name	Order Number
Tru64 UNIX Documentation CD-ROM	QA-6ADAA-G8
Tru64 UNIX Documentation Kit	QA-6ADAA-GZ
End User Documentation Kit	QA-6ADAB-GZ
Startup Documentation Kit	QA-6ADAC-GZ
General User Documentation Kit	QA-6ADAD-GZ
System and Network Management Documentation Kit	QA-6ADAE-GZ
Developer's Documentation Kit	QA-6ADAG-GZ
Reference Pages Documentation Kit	QA-6ADAF-GZ

Reader's Comments

Tru64 UNIX
System Administration
AA-RH9FB-TE

Compaq welcomes your comments and suggestions on this manual. Your input will help us to write documentation that meets your needs. Please send your suggestions using one of the following methods:

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

If you are not using this form, please be sure you include the name of the document, the page number, and the product name and version.

Please rate this manual:

	Excellent	Good	Fair	Poor
Accuracy (software works as manual says)	☐	☐	☐	☐
Clarity (easy to understand)	☐	☐	☐	☐
Organization (structure of subject matter)	☐	☐	☐	☐
Figures (useful)	☐	☐	☐	☐
Examples (useful)	☐	☐	☐	☐
Index (ability to find topic)	☐	☐	☐	☐
Usability (ability to access information quickly)	☐	☐	☐	☐

Please list errors you have found in this manual:

Page	Description
________	____
________	____
________	____
________	____

Additional comments or suggestions to improve this manual:

____

____

____

____

What version of the software described by this manual are you using? ____________________

Name, title, department ______________________________________

Mailing address ______________________________________

Electronic mail ______________________________________

Telephone ______________________________________

Date ______________________________________

Do Not Cut or Tear - Fold Here and Tape

COMPAQ

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 33 MAYNARD MA

POSTAGE WILL BE PAID BY ADDRESSEE

COMPAQ COMPUTER CORPORATION
UBPG PUBLICATIONS MANAGER
ZKO3-3/Y32
110 SPIT BROOK RD
NASHUA NH 03062-2698

Do Not Cut or Tear - Fold Here

Cut on This Line