

# Tru64 UNIX

---

## System Administration

Part Number: AA-RH9FD-TE

**June 2001**

**Product Version:** Tru64 UNIX Version 5.1A or higher

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

---

© 2001 Compaq Computer Corporation

COMPAQ, the Compaq logo, AlphaServer, Compaq Insight Manager, and TruCluster Registered in U.S. Patent and Trademark Office. Tru64 is a trademark of Compaq Information Technologies Group, L.P.

Microsoft, Windows, Windows NT are trademarks of Microsoft Corporation. UNIX and The Open Group are registered trademarks of the Open Group. All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from Compaq required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

---

# Contents

## About This Guide

### 1 System Administration Methods and Utilities

1.1	Overview of the SysMan Menu and Other Utilities .....	1-1
1.2	Related Documentation .....	1-3
1.3	Setting Up Your System .....	1-5
1.4	Administrative Methods .....	1-8
1.5	Administrative Utilities Under CDE .....	1-11
1.5.1	Accessing SysMan Under CDE .....	1-12
1.5.2	System Setup .....	1-14
1.6	SysMan Menu .....	1-15
1.7	Using the SysMan Command Line .....	1-18
1.8	SysMan Station .....	1-21
1.8.1	Using SysMan Station Status Options .....	1-23
1.8.2	Using SysMan Station Views .....	1-23
1.8.3	Using SysMan Station Menu Options .....	1-26
1.9	Compaq Insight Manager .....	1-27
1.10	Using SysMan on a Personal Computer .....	1-30
1.11	Setting Up a Serial Line Console .....	1-31
1.11.1	Setting Up a Console Port .....	1-32
1.11.1.1	Connecting the Modem to COMM1 .....	1-32
1.11.1.2	Setting the Configurable DCD Timer Value .....	1-33
1.11.1.3	Setting the Console Environment Variables .....	1-33
1.11.1.4	Verifying the Modem Setup .....	1-34
1.11.2	Initiating a Console Port Connection .....	1-34
1.11.2.1	Using the Console Port .....	1-34
1.11.2.1.1	Turning Off Console Log Messages .....	1-35
1.11.2.1.2	Shutting Down the Remote System .....	1-35
1.11.2.1.3	Ending a Remote Session .....	1-35
1.11.2.2	Troubleshooting the Modem Connection .....	1-35

### 2 Starting Up and Shutting Down the System

2.1	Overview of the Shutdown and Boot Operations .....	2-2
2.1.1	Shutdown Methods .....	2-2
2.1.2	Boot Methods .....	2-2

2.1.3	Related Documentation .....	2-3
2.1.4	System Files .....	2-4
2.1.5	Related Utilities .....	2-5
2.2	Understanding the Boot Operation .....	2-6
2.2.1	Booting Automatically or Manually .....	2-6
2.2.2	Booting to Single-User or Multiuser Mode .....	2-6
2.3	Preparing to Boot the Installed System .....	2-7
2.3.1	Preparing to Boot a Powered-Down System .....	2-8
2.3.2	Preparing to Boot a Powered-Up, Halted System .....	2-8
2.3.3	Preparing to Transition from Single-User Mode .....	2-9
2.3.4	Preparing to Boot a Crashed System .....	2-9
2.3.5	Preparing to Boot a System Taken Off the Network .....	2-10
2.4	Booting the System .....	2-12
2.4.1	Defining the Console Environment Variables and Using the Boot Commands .....	2-12
2.4.2	Overriding the Boot Commands .....	2-15
2.4.3	Using Interactive Boot to Verify the Root File System .....	2-16
2.5	Identifying System Run Levels .....	2-17
2.6	Changing System Run Levels .....	2-17
2.6.1	Changing Run Levels in Single-User Mode .....	2-18
2.6.2	Changing Run Levels from Multiuser Mode .....	2-18
2.6.2.1	Changing to a Different Multiuser Run Level .....	2-19
2.6.2.2	Changing to Single-User Mode .....	2-19
2.6.2.3	Reexamining the inittab File .....	2-20
2.7	Symmetric Multiprocessing .....	2-20
2.7.1	Adding CPUs to an Existing System .....	2-20
2.7.2	Unattended Reboots on Multiprocessor Systems .....	2-20
2.8	Setting and Resetting the System Clock .....	2-21
2.9	Troubleshooting Boot Problems .....	2-21
2.10	Shutting Down the System .....	2-22
2.11	Stopping Systems While in Multiuser Mode .....	2-23
2.11.1	Using SysMan shutdown .....	2-24
2.11.2	Shutting Down the System and Warning Other Users .....	2-25
2.11.3	Shutting Down and Halting the System .....	2-27
2.11.4	Shutting Down and Automatically Rebooting the System ..	2-27
2.11.5	Shutting Down and Halting Systems Immediately .....	2-28
2.12	Stopping Systems While in Single-User Mode .....	2-29
2.12.1	Stopping and Rebooting Systems with the reboot Command .....	2-29
2.12.2	Stopping Systems with the fasthalt Command .....	2-30
2.12.3	Stopping Systems with the fastboot Command .....	2-30

### 3 Customizing the System Environment

3.1	Identifying and Modifying the System Initialization Files .....	3-2
3.1.1	Using the /etc/inittab File .....	3-4
3.1.1.1	Specifying the Initialization Default Run Level .....	3-7
3.1.1.2	Specifying wait Run Levels .....	3-7
3.1.1.3	Specifying Console Run Levels .....	3-7
3.1.1.4	Specifying Terminals and Terminal Run Levels .....	3-8
3.1.1.5	Specifying Process Run Levels .....	3-8
3.1.1.6	Securing a Terminal Line .....	3-9
3.1.2	Using the init and rc Directory Structure .....	3-9
3.1.2.1	The init.d Directory .....	3-9
3.1.2.2	The rc0.d Directory and rc0 Run Command Script .....	3-10
3.1.2.3	The rc2.d Directory and rc2 Run Command Script .....	3-11
3.1.2.4	The rc3.d Directory and rc3 Run Command Script .....	3-12
3.1.3	Using the crontabs Directory .....	3-14
3.2	Using National Language Support .....	3-16
3.2.1	Setting Locale .....	3-18
3.2.2	Modifying Locale Categories .....	3-19
3.2.3	Limitations of Locale Variables .....	3-20
3.2.4	Setting Environment Variables for Message Catalogs and Locales .....	3-20
3.3	Customizing Internationalization Features .....	3-21
3.4	Customizing Your Time Zone .....	3-21
3.5	Customizing CPU Resource Allocation .....	3-24
3.5.1	Class Scheduler Overview .....	3-25
3.5.1.1	Related Utilities .....	3-26
3.5.1.2	Invoking the Class Scheduler .....	3-26
3.5.2	Planning Class Scheduling .....	3-27
3.5.3	Configuring Class Scheduling .....	3-27
3.5.4	Creating and Managing Classes .....	3-29
3.5.4.1	Creating a Class .....	3-30
3.5.4.2	Managing Identifier Types Within Classes .....	3-30
3.5.4.3	Enabling the Class Scheduler .....	3-31
3.5.4.4	Adding Members to a Class .....	3-32
3.5.4.5	Deleting Members From a Class .....	3-32
3.5.4.6	Other Class Management Options .....	3-32
3.5.5	Using the runclass Command .....	3-33
3.5.6	Using the Class Scheduling Graphical Interface .....	3-34
3.5.7	Creating or Modifying a Database .....	3-35
3.6	Customizing Power Management .....	3-37
3.6.1	Using the dxpowers Utility's Graphical User Interface .....	3-38

3.6.2	Using the sysconfig Command .....	3-39
3.6.2.1	Changing Power Management Values .....	3-39
3.6.2.2	Changing a Running Kernel or X Server .....	3-41
3.6.3	Using the SysMan Station .....	3-41
3.7	Adding Swap Space .....	3-41
3.7.1	Related Documentation and Utilities .....	3-43
3.7.2	Allocating Swap Space .....	3-43
3.7.3	Estimating Swap Space Requirements .....	3-45
3.7.4	Selecting the Swap Space Allocation Method .....	3-46
3.7.5	Correcting an Apparent Lack of Swap Space .....	3-46

## 4 Configuring the Kernel

4.1	Related Documentation and Utilities .....	4-2
4.2	System Configuration at Installation Time .....	4-4
4.3	Deciding When and How to Reconfigure Your Kernel .....	4-4
4.4	Dynamic System Configuration .....	4-6
4.4.1	Configuring Subsystems .....	4-7
4.4.2	Listing the Configured Subsystems .....	4-7
4.4.3	Determining the Subsystem Type .....	4-8
4.4.4	Unloading a Subsystem .....	4-9
4.4.5	Maintaining the List of Automatically Configured Subsystems .....	4-9
4.4.6	Managing Subsystem Attributes .....	4-10
4.4.6.1	Determining the Current Value of Subsystem Attributes .....	4-11
4.4.6.2	Identifying Run-time Configurable Subsystem Attributes .....	4-12
4.4.6.3	Modifying Attribute Values at Run Time .....	4-13
4.4.7	Managing Subsystems and Attributes Remotely .....	4-13
4.4.8	Managing the Subsystem Attributes Database .....	4-14
4.4.8.1	Listing Attributes in the Database .....	4-15
4.4.8.2	Adding Attributes to the Database .....	4-16
4.4.8.3	Merging New Definitions into Existing Database Entries .....	4-16
4.4.8.4	Updating Attributes in the Database .....	4-17
4.4.8.5	Removing Attribute Definitions from the Database ...	4-17
4.4.8.6	Deleting Subsystem Entries from the Database .....	4-18
4.5	Static System Configuration .....	4-18
4.5.1	Building the Kernel to Add Support for a New Device .....	4-20
4.5.2	Building the Kernel to Add Selected Kernel Options .....	4-24
4.5.3	Building a Kernel After Modifying System Files .....	4-26
4.6	Configuration Files .....	4-28

4.6.1	Configuration Files in /usr/sys/conf .....	4-29
4.6.2	Extensions to the Target Configuration File .....	4-30
4.6.3	The param.c File .....	4-31
4.6.4	System Configuration File Entries .....	4-32

## 5 Administering Hardware

5.1	Understanding Hardware .....	5-2
5.2	Reference Information .....	5-6
5.2.1	Related Documentation .....	5-6
5.2.2	Identifying Hardware Management System Files .....	5-7
5.2.3	WWIDs and Shared Devices .....	5-8
5.2.4	Related Commands and Utilities .....	5-9
5.3	Using the SysMan Hardware Tasks .....	5-10
5.3.1	Viewing the Hardware Hierarchy .....	5-11
5.3.2	Viewing the Cluster .....	5-13
5.3.3	Viewing Device Information .....	5-13
5.3.4	Viewing CPU Information .....	5-15
5.3.5	Using the SysMan Station .....	5-15
5.4	Using hwmgr to Manage Hardware .....	5-16
5.4.1	Understanding the Hardware Management Model .....	5-17
5.4.2	Understanding hwmgr Command Options .....	5-18
5.4.3	Configuring the hwmgr Environment .....	5-19
5.4.4	Using hwmgr to Manage Hardware .....	5-21
5.4.4.1	Locating SCSI Hardware .....	5-21
5.4.4.2	Viewing the System Hierarchy .....	5-21
5.4.4.3	Viewing System Categories .....	5-22
5.4.4.4	Obtaining Component Attributes .....	5-23
5.4.4.5	Setting Component Attributes .....	5-25
5.4.4.6	Viewing the Cluster .....	5-26
5.4.4.7	Viewing Devices .....	5-26
5.4.4.8	Viewing Transactions .....	5-28
5.4.4.9	Deleting a SCSI Device .....	5-29
5.4.4.10	Creating a User-Defined SCSI Device Name .....	5-30
5.4.4.11	Replacing a Failed SCSI Device .....	5-33
5.4.4.12	Using hwmgr to Replace a Cluster Member's Boot Disk .....	5-34
5.4.4.13	Viewing the Persistence Database for the name Subsystem .....	5-35
5.4.4.14	Deleting and Removing a Device from the name Persistence Database .....	5-36
5.5	Device Naming and Device Special Files .....	5-37

5.5.1	Related Documentation and Commands .....	5-38
5.5.2	Device Special File Directories .....	5-39
5.5.2.1	Legacy Device Special File Names .....	5-40
5.5.2.2	Current Device Special File Names .....	5-41
5.5.2.3	Converting Device Special File Names .....	5-42
5.5.3	Managing Device Special Files .....	5-43
5.5.3.1	Using <code>dn_setup</code> to Perform Generic Operations .....	5-44
5.5.3.2	Displaying Device Classes and Categories .....	5-45
5.5.3.3	Verifying and Fixing the Databases .....	5-46
5.5.3.4	Deleting Device Special Files .....	5-48
5.5.3.5	Moving and Exchanging Device Special File Names ..	5-49
5.6	Manually Configuring Devices Using <code>ddr_config</code> .....	5-49
5.6.1	Dynamic Method to Reconfigure the Kernel .....	5-49
5.6.1.1	Understanding Dynamic Device Recognition .....	5-50
5.6.1.1.1	Conforming to Standards .....	5-50
5.6.1.1.2	Understanding DDR Messages .....	5-51
5.6.2	Changing the DDR Database .....	5-51
5.6.3	Converting Customized <code>cam_data.c</code> Information .....	5-52
5.6.4	Adding Pseudoterminals and Devices Without Using DDR .....	5-52
5.6.4.1	Adding Pseudoterminals .....	5-53
5.6.4.2	Adding Other Devices .....	5-55
5.7	Using Device Commands and Utilities .....	5-56
5.7.1	Finding Device Utilities .....	5-57
5.7.2	SCSI and Device Driver Utilities .....	5-59
5.7.2.1	Using the SCSI Configuration Utility, <code>scu</code> .....	5-59
5.7.2.2	Using the Device Switch Manager, <code>devswmgr</code> .....	5-60
5.7.3	Partitioning Disks Using <code>diskconfig</code> .....	5-61
5.7.4	Manually Partitioning Disks .....	5-63
5.7.4.1	Checking for Overlapping Partitions .....	5-66
5.7.5	Copying Disks .....	5-67
5.7.6	Monitoring Disk Use .....	5-68
5.7.6.1	Checking Available Free Space .....	5-69
5.7.6.2	Checking Disk Use .....	5-70

## 6 Administering File Systems

6.1	Introduction to File Systems .....	6-2
6.1.1	Directory Hierarchy for File Systems .....	6-4
6.1.2	Disk Partitions .....	6-4
6.1.3	UFS Version 4.0 .....	6-6
6.1.4	File System Structures: UFS .....	6-8



6.1.5	Directories and File Types .....	6-10
6.1.6	Device Special Files .....	6-10
6.2	Context-Dependent Symbolic Links and Clusters .....	6-11
6.2.1	Related Documentation .....	6-12
6.2.2	Description of CDSLs .....	6-12
6.2.2.1	Structure of a CDSL .....	6-14
6.2.3	Maintaining CDSLs .....	6-15
6.2.3.1	Checking CDSL Inventory .....	6-15
6.2.3.2	Creating CDSLs .....	6-16
6.3	Creating UFS File Systems Manually .....	6-16
6.3.1	Using newfs to Create a New File System .....	6-16
6.3.2	Making File Systems Accessible to Users .....	6-20
6.3.3	Using the /etc/fstab File .....	6-20
6.3.4	Using the mount Command .....	6-23
6.3.5	Using the umount Command .....	6-23
6.3.6	Extending the UFS File System .....	6-24
6.3.6.1	Extending a Dismounted File System .....	6-26
6.3.6.2	Extending a Mounted File System .....	6-27
6.4	Administering UFS File Systems Using SysMan .....	6-27
6.4.1	File System Tasks in the SysMan Menu .....	6-29
6.4.2	Using SysMan to Dismount a File System .....	6-30
6.4.3	Using SysMan to Display Mounted File Systems .....	6-30
6.4.4	Using SysMan to Mount File Systems .....	6-32
6.4.5	Using SysMan to Share a Local Directory .....	6-35
6.4.6	Using SysMan to Mount a Network File System .....	6-37
6.4.7	Using SysMan to Create a UFS File System .....	6-40
6.5	Managing Quotas .....	6-41
6.5.1	Hard and Soft Quota Limits .....	6-42
6.5.2	Activating File System Quotas .....	6-42
6.5.3	Setting File System Quotas for User Accounts .....	6-44
6.5.4	Verifying File System Quotas .....	6-44
6.6	Backing Up and Restoring File Systems .....	6-45
6.7	Monitoring and Tuning File Systems .....	6-45
6.7.1	Checking UFS Consistency .....	6-45
6.7.2	Monitoring File System Use of Disks .....	6-46
6.7.2.1	Checking Available Free Space .....	6-46
6.7.2.2	Checking Disk Use .....	6-47
6.7.3	Improving UFS read Efficiency .....	6-49
6.8	Troubleshooting File Systems .....	6-49

## 7 Administering User Accounts and Groups

7.1	Account Administration Options and Restrictions .....	7-1
7.1.1	Administrative Utilities .....	7-1
7.1.2	Notes and Restrictions on Using the Utilities .....	7-3
7.1.3	Related Documentation .....	7-4
7.1.4	Related Utilities .....	7-5
7.2	Account Administration - Quick Start .....	7-6
7.2.1	Creating Primary Accounts During System Setup .....	7-6
7.2.2	Using the Account Manager (dxaccounts) GUI .....	7-6
7.2.3	Using the SysMan Menu Accounts Option .....	7-7
7.2.4	Using the Command-Line Utilities .....	7-8
7.2.5	Advanced Server for UNIX .....	7-9
7.3	Understanding User Accounts and Groups .....	7-9
7.3.1	System Files .....	7-9
7.3.2	Understanding Identifiers – UIDs and GIDs .....	7-10
7.3.3	Understanding the Password File .....	7-11
7.3.4	Understanding the Group File .....	7-14
7.4	Administering User Accounts .....	7-15
7.4.1	Using the SysMan Menu Accounts Options .....	7-15
7.4.1.1	Gathering Account Information .....	7-16
7.4.1.2	Setting Filter and Display Options .....	7-18
7.4.1.3	Using Filter Options .....	7-19
7.4.1.4	Creating or Modifying Local Accounts .....	7-20
7.4.1.5	Deleting Local Accounts .....	7-21
7.4.1.6	Creating or Modifying NIS Accounts .....	7-22
7.4.1.7	Deleting NIS Accounts .....	7-23
7.4.2	Using the Account Manager (dxaccounts) .....	7-23
7.4.2.1	Adding and Modifying Accounts .....	7-24
7.4.2.2	Deleting Accounts .....	7-25
7.4.2.3	Finding and Selecting Accounts .....	7-25
7.4.2.4	Copying Accounts .....	7-26
7.4.2.5	Using the Password Option .....	7-27
7.4.2.6	Account Manager (dxaccounts) General Options .....	7-27
7.5	Administering Groups .....	7-28
7.5.1	Using the SysMan Menu Accounts Group Options .....	7-28
7.5.1.1	Gathering Group Information .....	7-29
7.5.1.2	Creating or Modifying Groups .....	7-29
7.5.2	Using the Account Manager (dxaccounts) .....	7-30
7.5.2.1	Adding Groups .....	7-31
7.5.2.2	Modifying Groups .....	7-32
7.5.2.3	Deleting Groups .....	7-32

7.5.2.4	Finding Groups .....	7-32
7.6	Administering Windows Domain Accounts and Groups .....	7-33
7.6.1	Administering Synchronized Accounts .....	7-35
7.6.1.1	Using SysMan Menu Accounts and Groups Options ...	7-36
7.6.1.2	Using the Account Manager (dxaccounts) .....	7-36
7.6.1.3	Using Command-Line Utilities .....	7-37
7.6.1.4	Using the ASU User Manager for Domains .....	7-40
7.6.1.5	Using ASU net Commands .....	7-40
7.6.2	Windows 2000 Single Sign-On .....	7-41
7.6.2.1	Single Sign-On Installation Requirements .....	7-41
7.6.2.2	Installing the Single Sign-On Software .....	7-42
7.6.2.3	UNIX Requirements for Creating Single Sign-On Accounts .....	7-42
7.6.2.4	Creating Single Sign-On Accounts and Groups .....	7-44
7.6.2.5	Single Sign-On System Files .....	7-44

## 8 Administering the Print Services

8.1	Print Administrative Tasks .....	8-1
8.1.1	Advanced Printing Software .....	8-2
8.1.2	Printer Configuration Methods .....	8-2
8.1.3	Related Documentation .....	8-4
8.1.4	System Files .....	8-5
8.1.5	Related Utilities .....	8-6
8.2	Configuring Printers .....	8-7
8.2.1	Gathering Information .....	8-7
8.2.1.1	Direct and Network Printers .....	8-7
8.2.1.2	Remote Printers .....	8-9
8.2.1.3	Printer Data Worksheet .....	8-10
8.2.2	Installing a Local Printer .....	8-11
8.2.3	Setting Up Remote Printers .....	8-13
8.2.4	TCP/IP Printing .....	8-14
8.2.4.1	Setting Up TCP/IP Printing Using printconfig .....	8-15
8.2.4.2	Setting Up TCP/IP Printing Manually .....	8-16
8.2.5	Configuring PC Print Queues .....	8-18
8.2.6	Using lprsetup to Install a Printer .....	8-18
8.2.6.1	Print Symbols for Advanced Printing Services .....	8-22
8.3	Routine Print System Maintenance .....	8-22
8.3.1	Adding Printers .....	8-22
8.3.2	Modifying Printers .....	8-23
8.3.3	Removing Printers .....	8-23
8.3.4	Controlling Local Print Jobs and Queues .....	8-24

8.3.5	Enabling Printer Accounting .....	8-26
8.4	Reference Information .....	8-27
8.4.1	The /etc/printcap File .....	8-28
8.4.2	Data in /etc/printcap .....	8-29
8.4.2.1	Printer Name .....	8-29
8.4.2.2	Printer Type .....	8-30
8.4.2.3	Printer Synonyms .....	8-30
8.4.2.4	Device Special File .....	8-31
8.4.2.5	Connection Type .....	8-32
8.4.2.6	Spooling Directories .....	8-32
8.4.2.6.1	Spooling Directory Files .....	8-32
8.4.2.6.2	Creating a Spooling Directory .....	8-34
8.4.2.7	Baud Rate .....	8-34
8.4.3	Line Printer Daemon .....	8-34
8.4.4	Error Logging .....	8-35
8.4.5	Line Printer Daemon Filter Directory .....	8-36
8.4.6	General Purpose Print Filter .....	8-36
8.4.7	Flag Bits .....	8-37
8.4.8	Mode Bits .....	8-39
8.4.9	Remote Printer Characteristics .....	8-39
8.4.10	Pagination and Imaging Parameters .....	8-40
8.4.11	Generic Print Filters .....	8-41
8.5	Known Restrictions on the Use of Filters .....	8-41
8.6	Testing and Troubleshooting Printers .....	8-42

## 9 Administering the Archiving Services

9.1	Understanding Backup Tasks .....	9-2
9.2	Backing Up Data and System Files .....	9-3
9.3	Choosing a Backup Schedule .....	9-4
9.4	Backup Methods .....	9-5
9.5	Preparing to Perform a Backup .....	9-6
9.5.1	System Files .....	9-7
9.5.2	Related Utilities .....	9-7
9.5.3	Prerequisite Tasks .....	9-8
9.6	Using the dump Command .....	9-11
9.6.1	Performing a Full Backup .....	9-11
9.6.2	Performing an Incremental Backup .....	9-13
9.6.3	Performing a Remote Backup .....	9-13
9.6.4	Using Backup Scripts .....	9-14
9.7	Restoring Data .....	9-14
9.7.1	Restoring a File System .....	9-15

9.7.2	Restoring Files Manually .....	9-16
9.7.3	Restoring Files Interactively .....	9-17
9.7.4	Restoring Files Remotely .....	9-19
9.7.5	Restoring or Duplicating a System (Root) Disk .....	9-20
9.7.5.1	Preparing for Recovery or Duplication .....	9-21
9.7.5.2	Determining the Restoration Requirements .....	9-22
9.7.5.3	Applying the Procedure .....	9-24
9.7.5.4	Using Alternative root Disk Duplication Methods .....	9-28
9.7.6	Restoring the /usr and /var File System .....	9-29
9.8	Using the Command-Line Utilities, tar, pax, and cpio .....	9-30
9.9	Using dxarchiver .....	9-31
9.10	Creating a Standalone System Kernel on Tape .....	9-34
9.10.1	Tape Device Requirements .....	9-34
9.10.2	Using the btcreate Utility .....	9-35
9.10.2.1	Gathering Information .....	9-35
9.10.2.2	Creating the SAS Kernel .....	9-36
9.10.3	Using the btextract Utility .....	9-37
9.10.4	Using the SysMan Menu boot_tape Option .....	9-38

## 10 Administering the System Accounting Services

10.1	Accounting Overview .....	10-1
10.1.1	Accounting Shell Scripts and Commands .....	10-3
10.1.2	Accounting Files .....	10-5
10.2	Setting Up Accounting .....	10-9
10.2.1	Enabling Accounting in the rc.config File .....	10-10
10.2.2	Verifying the qacct, pacct, and fee Files .....	10-10
10.2.3	Editing the holidays File .....	10-11
10.2.4	Modifying the crontab Files .....	10-11
10.3	Starting Up and Stopping Accounting .....	10-12
10.4	Connect Session Accounting .....	10-13
10.4.1	The wtmpfix Command .....	10-15
10.4.2	The fwtmp Command .....	10-16
10.4.3	The acctwtmp Command .....	10-17
10.4.4	The ac Command .....	10-17
10.4.5	The acctcon1 Command .....	10-18
10.4.6	The acctcon2 Command .....	10-19
10.4.7	The prctmp Shell Script .....	10-19
10.4.8	The lastlogin Shell Script .....	10-20
10.4.9	The last Command .....	10-20
10.5	Process Accounting .....	10-21
10.5.1	The accon Command .....	10-23

10.5.2	The turnacct Shell Script .....	10-23
10.5.3	The ckpacct Shell Script .....	10-24
10.5.4	The acctcom Command .....	10-24
10.5.5	The sa Command .....	10-26
10.5.6	The acctcms Command .....	10-27
10.5.7	The acctprc1 Command .....	10-28
10.5.8	The acctprc2 Command .....	10-29
10.5.9	The lastcomm Command .....	10-30
10.6	Disk Usage Accounting .....	10-30
10.6.1	The dodisk Shell Script .....	10-31
10.6.2	The diskusg Command .....	10-31
10.6.3	The acctdusg Command .....	10-32
10.6.4	The acctdisk Command .....	10-33
10.7	System Administration Service Accounting .....	10-33
10.8	Printer Accounting .....	10-34
10.9	Creating Daily, Summary, and Monthly Report Files .....	10-35
10.9.1	The runacct Shell Script .....	10-35
10.9.1.1	Correcting runacct Shell Script Errors .....	10-36
10.9.1.2	Examples of Errors and Corrective Actions .....	10-38
10.9.2	The acctmerg Command .....	10-39
10.9.3	The prtacct Shell Script .....	10-40
10.9.4	The prdaily Shell Script .....	10-40
10.9.5	The monacct Shell Script .....	10-41

## 11 Monitoring and Testing the System

11.1	Overview of Monitoring and Testing .....	11-2
11.1.1	Guidelines for Monitoring Systems .....	11-2
11.1.2	Summary of Commands and Utilities .....	11-3
11.1.2.1	Command-Line Utilities .....	11-4
11.1.2.2	SysMan Menu Monitoring and Tuning Tasks .....	11-5
11.1.2.3	X11-Compliant Graphical Interfaces .....	11-6
11.1.2.4	Advanced Monitoring Utilities .....	11-8
11.1.3	Related Documentation .....	11-9
11.2	Configuring and Using Monitoring Utilities .....	11-10
11.2.1	Using collect to Record System Data .....	11-10
11.2.2	Using the sys_check Utility .....	11-11
11.2.3	Using the Monitoring Performance History Utility .....	11-13
11.3	Environmental Monitoring .....	11-14
11.3.1	Loadable Kernel Module .....	11-14
11.3.1.1	Specifying Loadable Kernel Attributes .....	11-15
11.3.1.2	Obtaining Platform-Specific Functions .....	11-15

11.3.1.3	Server System MIB Subagent .....	11-16
11.3.2	Monitoring Environmental Thresholds .....	11-17
11.3.2.1	Environmental Monitoring Daemon .....	11-17
11.3.2.2	Customizing the envmond Daemon .....	11-17
11.3.3	User-Definable Messages .....	11-18
11.4	Using System Exercisers .....	11-18
11.4.1	Running System Exercisers .....	11-19
11.4.2	Using Exerciser Diagnostics .....	11-19
11.4.3	Exercising a File System .....	11-20
11.4.4	Exercising System Memory .....	11-20
11.4.5	Exercising Shared Memory .....	11-21
11.4.6	Exercising a Disk Drive .....	11-22
11.4.7	Exercising a Tape Drive .....	11-23
11.4.8	Exercising the Terminal Communication System .....	11-24

## 12 Administering the Basic System Event Channels

12.1	Understanding the Basic Event-Logging Facilities .....	12-1
12.1.1	System Event Logging .....	12-2
12.1.2	Binary Event Logging .....	12-3
12.2	Configuring Event Logging .....	12-4
12.2.1	Editing the Configuration Files .....	12-5
12.2.1.1	Editing the syslog.conf File .....	12-5
12.2.1.2	Configuring syslog to Use EVM .....	12-8
12.2.1.3	Editing the binlog.conf File .....	12-9
12.2.2	syslog Security and Remote Messages .....	12-12
12.2.3	Creating the Special Files .....	12-13
12.2.4	Starting and Stopping the Event-Logging Daemons .....	12-14
12.2.4.1	The syslogd Daemon .....	12-14
12.2.4.2	The binlogd Daemon .....	12-15
12.2.5	Configuring the Kernel Binary Event Logger .....	12-16
12.3	Recovering Event Logs After a System Crash .....	12-17
12.4	Managing Log Files .....	12-18
12.5	Startup Log Messages in /var/adm/messages .....	12-18

## 13 Event Management Using EVM

13.1	EVM Overview .....	13-1
13.1.1	Understanding EVM Events .....	13-3
13.1.2	EVM Components .....	13-5
13.1.2.1	EVM Command-Line Utilities .....	13-7
13.1.2.2	EVM Application Programming Interface .....	13-8

13.1.2.3	EVM System Files .....	13-8
13.1.3	Related Utilities .....	13-10
13.2	Administering EVM .....	13-11
13.2.1	Configuring EVM .....	13-11
13.2.1.1	EVM Daemon Configuration .....	13-12
13.2.1.2	EVM Channel Configuration .....	13-13
13.2.1.3	EVM Logger Configuration .....	13-14
13.2.1.4	Secondary Logger Configuration Files .....	13-17
13.2.2	Security Considerations .....	13-18
13.2.2.1	User Authentication .....	13-18
13.2.2.2	User Authorization .....	13-18
13.2.2.3	Remote Access .....	13-20
13.2.3	Managing Log Files .....	13-20
13.2.4	Event Templates .....	13-21
13.2.5	Installing New EVM Clients .....	13-22
13.2.6	Configuring binlog Event Translation Utilities .....	13-22
13.3	Using EVM in System Administration .....	13-26
13.3.1	Displaying Events Using evmshow .....	13-26
13.3.2	Introducing Event Filters .....	13-28
13.3.3	Retrieving Stored Events Using evmget .....	13-30
13.3.4	Sorting Events Using evmsort .....	13-32
13.3.5	Using the -A Option to Simplify the Command String .....	13-33
13.3.6	Monitoring Events Using evmwatch .....	13-34
13.3.7	Posting Quick Message Events Using evmpost .....	13-35
13.3.8	Listing Registered Events .....	13-36
13.3.9	Posting Events from a Shell Script .....	13-37
13.3.10	Understanding the EVM Mark Event .....	13-40
13.3.11	Viewing Events Using the SysMan Event Viewer .....	13-41
13.3.12	Advanced Selection and Filtering Techniques .....	13-43
13.3.12.1	Filtering By Time .....	13-43
13.3.12.2	Using the Event-Id to Select Events for Detailed Display .....	13-44
13.3.12.3	Searching for Reserved Component Names .....	13-45
13.3.12.4	Using Filter Files .....	13-45
13.3.13	Logging and Forwarding Events .....	13-46
13.3.13.1	Logging Events .....	13-47
13.3.13.2	Using Forwarding to Handle Events Automatically ...	13-47
13.4	Troubleshooting EVM .....	13-49

## 14 Administering Crash Dumps

14.1	Overview of Crash Dumps .....	14-1
14.1.1	Related Documentation and Utilities .....	14-2



14.1.2	Files Used During Crash Dumps .....	14-3
14.2	Crash Dump Creation .....	14-3
14.2.1	Setting Dump Kernel Attributes in the Generic Subsystem .....	14-3
14.2.2	Crash Dump File Creation .....	14-5
14.2.3	Crash Dump Logging .....	14-6
14.2.4	Swap Space .....	14-7
14.2.5	Planning Crash Dump Space .....	14-11
14.2.6	Planning and Allocating File System Space for Crash Dump Files .....	14-12
14.3	Choosing the Content and Method of Crash Dumps .....	14-13
14.3.1	Adjusting the Primary Swap Partition's Crash Dump Threshold .....	14-13
14.3.2	Including User Page Tables in Partial Crash Dumps .....	14-14
14.3.3	Selecting Partial or Full Crash Dumps .....	14-15
14.3.4	Expected Dump Compression .....	14-15
14.3.5	Selecting and Using Noncompressed Crash Dumps .....	14-16
14.3.6	Dumping to Exempt Memory .....	14-17
14.3.7	Dumping to a Remote Host .....	14-18
14.4	Generating a Crash Dump Manually .....	14-19
14.4.1	Continuable Dumps on a Running System .....	14-19
14.4.2	Forcing Crash Dumps on a Hung System .....	14-20
14.5	Storing and Archiving Crash Dump Files .....	14-21
14.5.1	Compressing a Crash Dump File .....	14-21
14.5.2	Uncompressing a Partial Crash Dump File .....	14-22

## A Administration Utilities

A.1	X11 Graphical User Interfaces (CDE Application Manager) ...	A-1
A.2	SysMan Menu Tasks and Associated Utilities .....	A-6

## B Device Mnemonics

## C Support of the CI and HSC Hardware

C.1	Hardware Setup, Restrictions, and Revision Levels .....	C-1
C.2	Software Installation and Restrictions .....	C-2
C.3	Configuration File Entries .....	C-2
C.4	Booting an HSC Controller or an HSC Disk .....	C-2
C.5	Sharing Disk and Tape Units Among Several Hosts .....	C-3

## D Administering Specific Hardware

D.1	Introduction .....	D-1
D.2	PCMCIA Support .....	D-1
D.3	CalComp Graphics Tablet .....	D-2
D.4	AlphaServer GS140 Logical Partitions .....	D-2
D.4.1	Hardware Requirements .....	D-3
D.4.2	Preparing to Install and Operate Logical Partitions .....	D-4
D.4.2.1	Definition of Commonly Used Terms .....	D-5
D.4.3	Logical Partitions Configuration and Installation Tasks ..	D-7
D.4.3.1	Verifying Your System's Hardware Configuration .....	D-7
D.4.3.2	Verifying the Firmware Revision Level .....	D-10
D.4.3.3	Configuring Logical Partitions .....	D-10
D.4.3.4	Determining and Setting Environment Variables .....	D-11
D.4.3.5	Displaying Console Environment Variables .....	D-13
D.4.3.6	Correcting Console Environment Variables .....	D-13
D.4.3.7	Disabling Automatic Boot Reset .....	D-13
D.4.3.8	Setting Memory Interleave Mode .....	D-14
D.4.3.9	Setting the Operating System Type to UNIX .....	D-14
D.4.3.10	Setting the auto_action Console Environment Variable .....	D-14
D.4.4	Initializing Partitions .....	D-14
D.4.5	Correcting Interleave Mode Errors .....	D-15
D.4.6	Installing the Operating System .....	D-15
D.4.7	Managing a Partitioned System .....	D-16
D.4.7.1	Operational Characteristics .....	D-16
D.4.7.1.1	Console init command (P##>>>init) .....	D-16
D.4.7.1.2	Shutting Down or Rebooting the Operating System .....	D-17
D.4.7.2	Recovering an Interrupted Operating System Boot .....	D-17
D.4.7.3	Halting Processors .....	D-17
D.4.7.4	Power OFF/ENABLE Switch Position .....	D-18
D.4.7.5	Reconfiguring Partitions by Changing Console EVs ...	D-18
D.4.7.6	Checking Other Console EVs Before Booting .....	D-19
D.4.7.7	Logical Partitioning Informational Messages at Boot Time .....	D-20
D.4.8	Hardware Management and Maintenance .....	D-20
D.4.8.1	Obtaining Technical Support .....	D-21
D.4.8.2	Performing Hardware Management and Maintenance Tasks .....	D-21
D.4.9	Hardware Changes Requiring a UNIX Kernel Rebuild ....	D-22
D.4.9.1	How to Rebuild the UNIX Kernel for a Partition .....	D-23

D.4.10	Handling Nonrecoverable Hardware Error Machine Checks .....	D-24
D.4.11	Logical Partitioning Error Messages .....	D-26
D.4.12	Understanding Console Firmware Error or Informational Messages .....	D-27

## Index

### Examples

2-1	A Typical Shutdown Sequence .....	2-26
6-1	Default Partitions for RZ1DF-CB Disk and HSZ RAID Devices .....	6-5
7-1	Changing the Default Environment Variables Using usermod .....	7-33
12-1	Sample Translated Event .....	12-3
12-2	Sample syslog_evm.conf File Entries .....	12-9
13-1	Sample EVM Daemon Configuration File Entries .....	13-12
13-2	Sample EVM Channel Configuration File .....	13-13
13-3	Sample EVM Logger Configuration File Entries .....	13-15
13-4	Sample EVM Authorization File Entries .....	13-19
13-5	A binlogd Event Showing the DECEvent Translation .....	13-23

### Figures

1-1	System Setup Graphical Interface .....	1-6
1-2	Quick Setup .....	1-7
1-3	Custom Setup .....	1-8
1-4	CDE Tool Drawer and SysMan Station Icons .....	1-11
1-5	SysMan Applications Panel .....	1-13
1-6	The SysMan Menu .....	1-17
1-7	SysMan Station Main Window .....	1-22
1-8	AdvFS_Fileystems View .....	1-24
1-9	Hardware View .....	1-25
4-1	Configuration Files Directory Hierarchy .....	4-28
13-1	Event Model .....	13-4
13-2	EVM Component Model .....	13-5
14-1	Default dump_sp_threshold Attribute Setting .....	14-9
14-2	Crash Dump Written to Multiple Devices .....	14-10

### Tables

1	New and Changed Features by Chapter .....	xxii
---	---	------

2-1	Console Environment Variables .....	2-12
2-2	Options to the boot_osflags Variable .....	2-13
3-1	Locale Support Files .....	3-16
3-2	Locale Environment Variables .....	3-19
5-1	Tape Device Suffix for Legacy Device Special Files .....	5-41
5-2	Sample Current Device Special File Names .....	5-41
5-3	Sample Device Name Translations .....	5-43
5-4	Device Utilities Documented in the Guides .....	5-57
5-5	Device Utilities Documented in the Reference Pages .....	5-58
6-1	Disk Partition Tables .....	6-19
7-1	Utilities for Administering Accounts and Groups .....	7-2
7-2	Account Administration Worksheet .....	7-17
7-3	Account Administration Worksheet .....	7-18
7-4	Group Administration Worksheet .....	7-29
8-1	TCP/IP Socket Numbers .....	8-16
8-2	lprsetup Options .....	8-19
8-3	lpc Command Arguments .....	8-25
8-4	Communication Ports and Printer Device Special Files .....	8-31
8-5	Flag Bits .....	8-38
8-6	Mode Bits .....	8-39
8-7	Non-PostScript and PostScript Filters .....	8-41
9-1	Recovery Preparation .....	9-22
10-1	Accounting Commands and Shell Scripts .....	10-3
10-2	Database Files in the /var/adm Directory .....	10-5
10-3	Daily Files in the /var/adm/acct/nite Directory .....	10-6
10-4	Summary Files in the /var/adm/acct/sum Directory .....	10-8
10-5	Monthly Files in the /var/adm/acct/fiscal Directory .....	10-9
10-6	The utmp ASCII Conversion Structure Members .....	10-15
10-7	The tacct File Format .....	10-22
11-1	Parameters Defined in the Kernel Module .....	11-15
11-2	get_info() Function Types .....	11-16
11-3	Mapping of Server Subsystem Variables .....	11-16
13-1	EVM Command-Line Utilities .....	13-7
13-2	EVM Administrative Utilities .....	13-7
A-1	System Administration Configuration Applications .....	A-2
A-2	System Administration Daily Admin Applications .....	A-3
A-3	System Administration Monitoring and Tuning Applications ..	A-4
A-4	System Administration Software Management Applications ..	A-5
A-5	System Administration Storage Management Applications ...	A-5
A-6	System Administration Tools .....	A-6
B-1	Device Mnemonics .....	B-2

---

## About This Guide

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

### Audience

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

---

#### Note

---

This guide describes only the administration of the operating system on an AlphaServer processor and devices that are local to the processor, such as single-spindle disks. For important information on storage configurations, including the configuration and maintenance of storage arrays, see your StorageWorks documentation. You use StorageWorks software applications, such as the StorageWorks Command Console (SWCC) in addition to the utilities provided by the operating system. See *Related Documents* for resources on the Web.

---

### New and Changed Features

If you are updating your system from an older version of the UNIX operating system, you might 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.tru64unix.compaq.com/docs/>
- At the preceding web site, you can obtain technical updates for any information not included in the documentation provided with your media.

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

The following administrative reference pages are updated with no impact on the content of this guide:

- Kernel configuration – `dxkerneltuner(8)`, `sysconfig(8)`, and `sysconfigdb(8)`
- Hardware management – `tz(7)`, `rz(7)`, `emx(7)`, and `fd(7)`
- File systems – `fuser(8)`, `mount(8)`, `mountd(8)`, `ncheck(8)`, and `hier(5)`
- Monitoring – `collect(8)`, `psradm(8)`, `ps(1)`, `psrinfo(1)`, and `vmstat(1)`
- File formats – `proc(4)`, `processor_sets(4)`, `ffm(4)`, `fstab(4)`, and `dvdfs(4)`.

Where reference page revisions have impact in this guide, *Table 1* identifies the revised commands and utilities with their related chapter. This table has the following key:

- N (new) – Information added to the guide.
- C (changed) – Information revised since the previous release.
- R (removed) – Information deleted from the guide and possibly relocated in another source.

See *Organization* for a list of chapter titles and a description of their content.

**Table 1: New and Changed Features by Chapter**

Chapter	Change
1	N Added information on new SysMan Menu options is added.
	C Revised the <code>sysman(8)</code> , <code>sysman_clone(8)</code> , and <code>sysman_menu(8)</code> reference pages
2	C Revised and verified examples and procedures.
	R Removed information on the optional command syntaxes.
3	N Added information on correcting an apparent lack of swap space.
	N Added the <code>armtech(5)</code> introductory reference page.
	C Revised information on the <code>cron</code> command and there are changes to <code>cron(8)</code> .
	C Revised information on time zones and made changes to <code>zic(8)</code> , <code>timezone(8)</code> , and <code>settz(8)</code> .

**Table 1: New and Changed Features by Chapter (cont.)**

**Table 1: New and Changed Features by Chapter (cont.)**

Chapter	Change
	C Revised the following class scheduler API reference pages: <code>class_configure(3)</code> , <code>class_create(3)</code> , <code>class_delete(3)</code> , <code>class_load_database(3)</code> , <code>class_open(3)</code> , and <code>class_restore_database(3)</code> .
5	N Added information on shared devices and the redirect operation is added.
	C Revised and reorganized <code>hwmgr(8)</code> by moving information to four new pages: <code>hwmgr_get(8)</code> , <code>hwmgr_ops(8)</code> , <code>hwmgr_show(8)</code> , and <code>hwmgr_view(8)</code> .
	C Revised the syntax of all <code>hwmgr</code> command options. Examples and procedures are updated to reflect the syntax changes. (The previous syntax is still supported).
	N Added information on new options for CPU hot swap (OLAR) to <code>hwmgr_ops(8)</code> . Information on CPU hot swap procedures is provided in the new <i>Managing Online Addition and Removal</i> guide.
	C Revised information on device naming to reflect updates to <code>dsfmgr(8)</code> and <code>disklabel(8)</code> .
	R Revised and relocated to <i>Chapter 9</i> information on restoring the root disk. Similar information is provided for AdvFS in the <i>AdvFS Administration</i> guide.
8	N Added information on new supported printers is added to <code>lprsetup.dat(4)</code> .
	C Added information on new options for print symbols and printer types is added. There are related revisions to the following reference pages: <code>printcap(4)</code> , and <code>lpr(1)</code> .
9	N Added information on new procedure for restoring a UFS root disk.
	N Added information on new options for bootable tape, including support for a custom pre-restore.
	C Revised <code>btcreate(8)</code> and <code>btextract(8)</code> .
	C Revised the section on recovering files.
	C Revised examples and procedures to reflect changes in hardware management commands; <code>hwmgr(8)</code> , <code>dsfmgr(8)</code> and <code>disklabel(8)</code> .
	C Revised <code>dxarchiver(8)</code> and <code>vrestore(8)</code> .
11	N Added information on a new API for monitoring the status of vital processes: <code>cfg_psm_catops(3)</code> , <code>cfg_psm_memops(3)</code> , and <code>psm.h(4)</code> .
	R Removed information on the command syntax for system exercisers and revised the examples.

**Table 1: New and Changed Features by Chapter (cont.)**

Chapter	Change
	R Moved information on the memory troller to the new <i>Managing Online Addition and Removal</i> guide.
	N Added a new <code>memory_trolling(5)</code> introductory reference.
12	C Revised <code>binlogd(8)</code> , <code>syslogd(8)</code> , and <code>syslog.conf(4)</code> and updated the chapter with this information. Revised examples and procedures.
	C Revised information on log files.
	R Removed information on class codes. See the reference pages.
13	C Added information on new command options. Revised examples and procedures. Information on <code>binlogd</code> events is revised.
	C Revised information on the logger and log files. See also <code>EVM(5)</code> for updates to related reference pages.
	C Revised information on <code>DECEvent</code> , <code>Compaq Analyze</code> and the <code>uerf</code> command is revised.
14	N Added Information on new dump kernel attributes.
	N Added Information on dumping to exempt memory.
	N Added Information on controlling the number of dumps.
	C Revised the <code>savecore(8)</code> and <code>vmzcore(7)</code> reference pages are revised.
A	N Added information on new SysMan Menu options.
	R Removed information on retired interfaces.

Except for the correction of errors, there are no changes to the technical content of *Chapter 4*, *Chapter 6*, *Chapter 7*, *Appendix B*, *Appendix C*, and *Appendix D*.

## Organization

This guide consists of the following chapters and appendices:

<i>Chapter 1</i>	Describes the methods and tools that you use to perform system administration tasks.
<i>Chapter 2</i>	Explains how to start up and shut down the operating system. Additionally, explains how to recover from an unexpected shutdown.
<i>Chapter 3</i>	Describes how to customize operating system files and operating system components to tailor the operating system environment.



<i>Chapter 4</i>	Describes how to dynamically and statically configure an operating system kernel.
<i>Chapter 5</i>	Describes hardware and device administration and device naming, how you administer storage devices (principally disk devices), and tasks such as adding swap space.
<i>Chapter 6</i>	Explains how to administer the UFS file system.
<i>Chapter 7</i>	Explains how to administer accounts for operating system users and groups of users.
<i>Chapter 8</i>	Explains how to administer the print services system and configure printers.
<i>Chapter 9</i>	Explains how to administer the archiving services of the operating system in order to back up and restore mass storage devices.
<i>Chapter 10</i>	Explains how to administer the resource accounting services of the operating system.
<i>Chapter 11</i>	Describes the monitoring and testing utilities.
<i>Chapter 12</i>	Explains how to set up and administer the basic event logging services of the operating system.
<i>Chapter 13</i>	Explains how to set up and administer EVM, the advanced event management and logging mechanism.
<i>Chapter 14</i>	Explains how to set up and administer crash dumps.
<i>Appendix A</i>	Lists the administration utilities.
<i>Appendix B</i>	Lists the device mnemonics.
<i>Appendix C</i>	Contains information about the CI bus and the Hierarchical Storage Controller (HSC) configuration.
<i>Appendix D</i>	Contains information on specific hardware devices.

---

## Related Documents

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

- The *Release Notes* provide important information such as restrictions on using certain operating system features.
- The *Installation Guide* and *Installation Guide — Advanced Topics* describe how to install your operating system. Several important administrative tasks are described in detail in these guides, such as installing software and installation cloning.

- The *Network Administration: Services* and *Network Administration: Connections* guides describes how to set up, configure, and troubleshoot your network.
- The Advanced Server for UNIX (ASU) *Concepts and Planning Guide* and *Installation and Administration Guide* provide information on administering Windows domain accounts and sharing printers with PC users. These documents are supplied with the ASU software on the *Associated Products CD-ROM, Volume 2*.
- The *Security* 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.

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

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

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

### Icons on Tru64 UNIX Printed Manuals

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

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

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

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

## Reader's Comments

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
```

Please include the following information along with your comments:

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

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate 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      MBN refers to the mouse button that you must press to select an item or initiate an action.

%	
\$	A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells.
#	A number sign represents the superuser prompt.
<i>file</i>	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, <code>cat(1)</code> indicates that you can find information on the <code>cat</code> command in Section 1 of the reference pages.
Ctrl/ <i>x</i>	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, <code>Ctrl/C</code> ).
<code>Return</code>	In an example, a key name enclosed in a box indicates that you press that key.

---

# 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 introduces Compaq Insight Manager, which you can use to view system status, and launch the SysMan Menu and the SysMan Station from a Web browser.
- Section 1.10 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 the user environments:

- 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, and Compaq 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, there might 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 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 might 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 displays 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 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 Compaq 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 might be able to invoke an administrative utility in several different user environments, or you might 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. See `curses(3)` 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 Compaq Insight Manager Web-based Management page. The following information on administering the operating system is available from this page:
    - Using SysMan Menu and the SysMan Station.
    - Using an X-capable user environment such as CDE.
    - Using a personal computer (PC) running Microsoft Windows. This section provides links to the client software that you must download to your PC.

---

**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 `insightd` daemon to display online help.

---

- Links to product information on the World Wide Web.

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

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

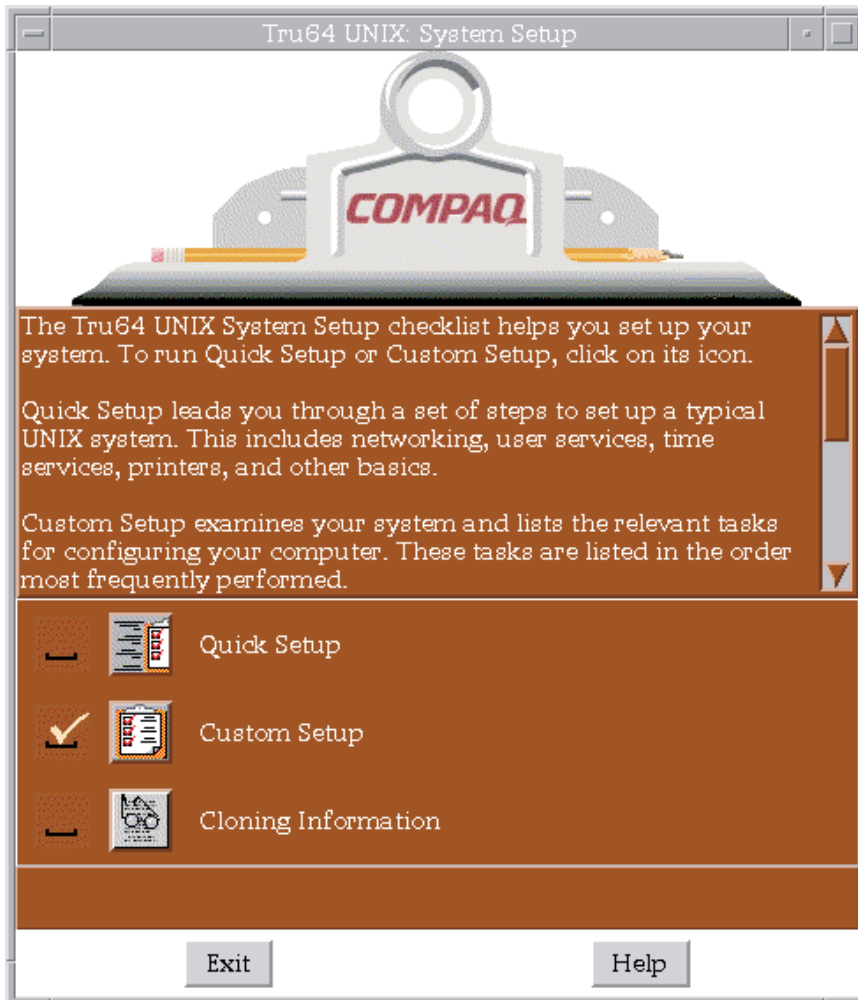
Note that there are restrictions on using Compaq Insight Manager, depending on your user environment. See Section 1.9 for information.

## 1.3 Setting Up Your System

The initial configuration of your system (setup) 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 might 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.

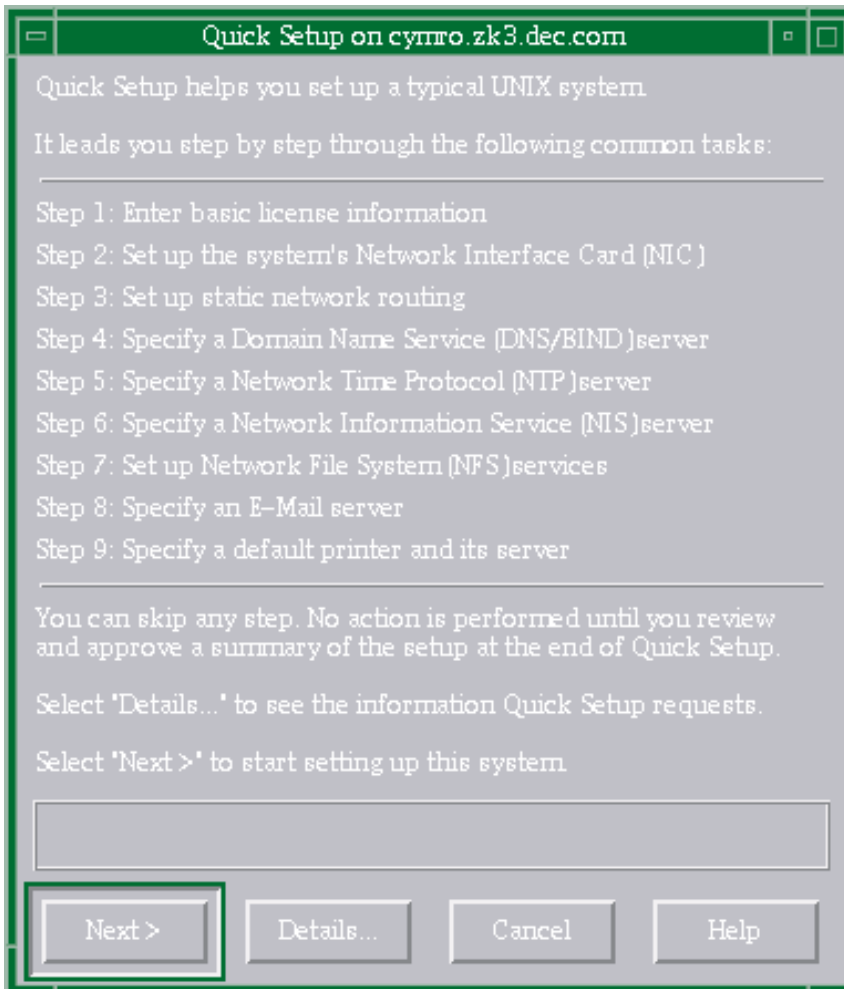
**Figure 1–1: System Setup Graphical Interface**



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 you can apply it to other systems. See the *Installation Guide — Advanced Topics* and `sysman_clone(8)`.

Refer to Section 1.5.2 for more information and see `setup(8)`.

## 1.4 Administrative Methods

You can accomplish most of the tasks described in this guide by using one or more 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 using Compaq Insight Manager

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 obtain detailed information on its properties or launch utilities that enable you to perform administrative tasks on the component. Unlike the SysMan Menu, the SysMan Station requires a graphics capability and cannot be run from the character-cell or `curses` user environments.

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 are 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 runs in other X-compliant windowing environments)

- The Accounts option on the SysMan Menu, available from Application Manager - System\_Admin in the CDE environment, or by invoking `sysman` from a terminal window.

You should migrate your system administration processes from the older command-line scripts to the appropriate SysMan Menu method. These command-line utilities are 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 might 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 `hier(5)`.

- 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 data is preserved. When you update your system the data is 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 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 might 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 run on other X11-compliant user environments. When you complete the full installation, the System Setup graphical interface is displayed to guide you through the process of configuring the system for initial use. From System Setup, you invoke the same graphical user interfaces (GUIs) that you use regularly to administer and customize the system. System Setup is described in Section 1.5.2.

Many of the administrative utilities that you invoke from within CDE start a SysMan Menu task option. However, some of the utilities are graphical, and either have no analogous SysMan Menu option, or offer features that 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. 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
```

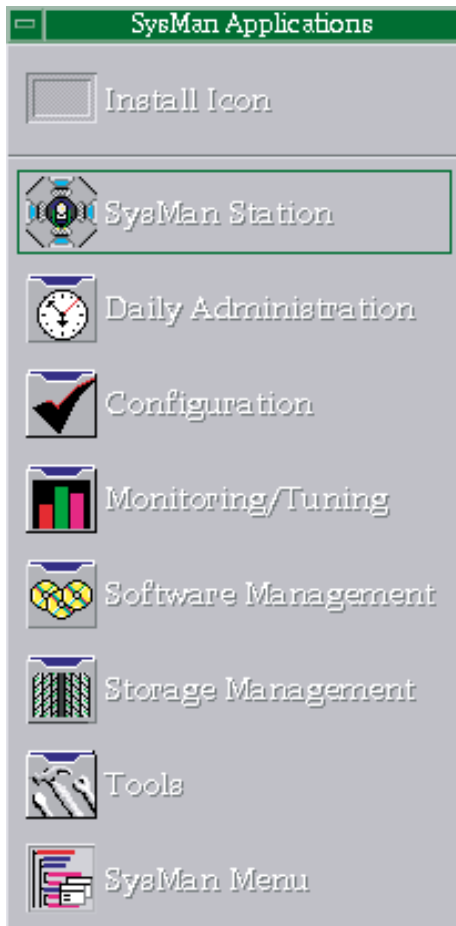
### 1.5.1 Accessing SysMan Under CDE

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 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. 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 applications 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, you can invoke the SysMan utilities from the command line. See `sysman_intro(8)` 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 might 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.

Not all configuration applications are available on all systems. The file `/etc/checklist.desc` contains a list of configuration applications. After you select an icon, the appropriate SysMan Menu utility, X11-based GUI, or character-cell script is invoked.

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 must 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 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.

You can perform all the tasks in the SysMan Menu by using an X11-capable display, a personal computer running Microsoft Windows, 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
```

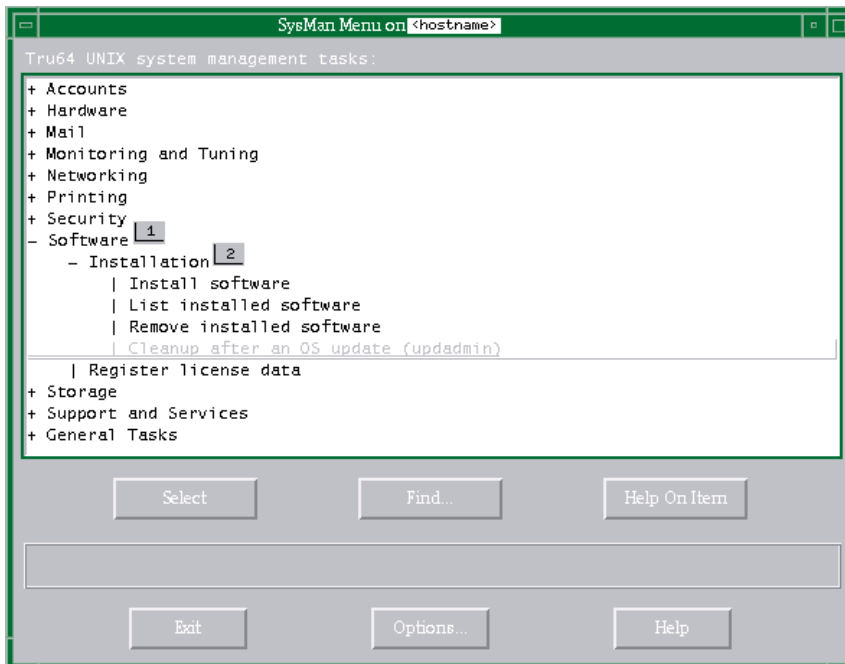
Use the following command to obtain a complete listing of the available tasks and their accelerators.

```
# /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. Each branch provides more options that might be further branches or might end in a task. You can collapse or expand each option if suboptions are available, as indicated by a character preceding each menu item. The plus sign (+) indicates that further menu items are available 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. The contents of this menu might 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. After you select a task, the appropriate utility is invoked.

How you move through and select menu items or invoke tasks is dependent on the user environment that you are using such as a *curses* terminal or a windowing environment. When using a terminal, you use the arrow keys or Tab key to move around the menu, highlighting options and buttons as you move. Use the Enter key to select an item, which expands a branch or selects 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 expands or contracts it. Selecting a task invokes its 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 invoke its associated utility in a format that is most appropriate for your current user environment. For example, the task is invoked in the X11-compliant windowing environment if you are using CDE, or in curses format if you are using a character-cell terminal.

See `sysman(8)` 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. See `sysman_cli(8)` for a complete list of options. 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 are prompted to enter them. The following command shows how you determine the keys for a particular table:

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

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

You 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 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, it displays an error message or dialog similar to the following:

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

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

---

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

Use the following procedure to start the SysMan Station from CDE:

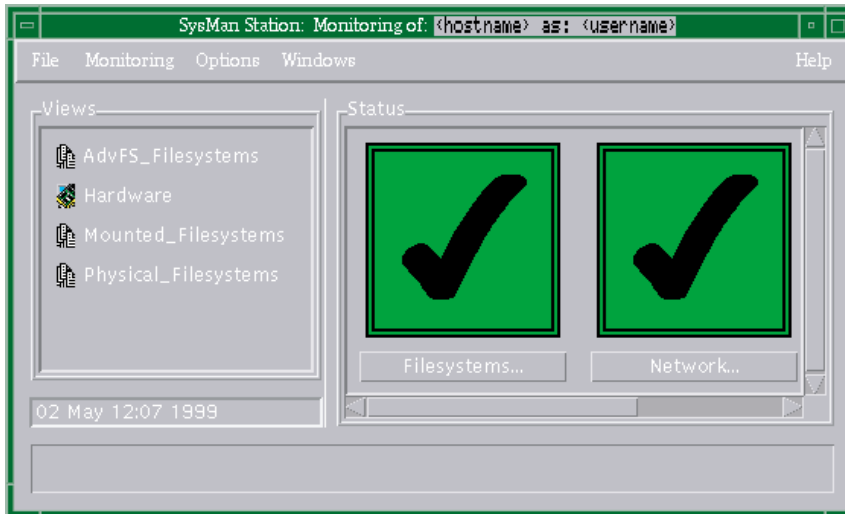
1. Log in as root and use the SysMan Station icon from the CDE Front Panel or from the SysMan Applications sub-panel. (This assumes the default CDE configuration, where the SysMan Station icon appears on the Front Panel under the SysMan Applications sub-panel.)
2. Choose the SysMan Station icon from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

To start SysMan Station from the command prompt, enter:

```
#sysman -station
```

After invoking SysMan Station, you are connected to the local host. The main SysMan Station window appears similar to the example shown in Figure 1–7, except that the default display shows the Filesystems..., Network..., and Storage... options that you can monitor. 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 is displayed.

The SysMan Station is a graphical representation of the system, in a hierarchical (tree) structure. For example, in the Storage option, you can view all disks on all buses for all processors on the system. You can select a specific device to monitor, and invoke utilities to administer that device. You can 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 changes color, becoming a cross (X) on a red background to indicate a serious problem. These icons also enable you to instantly display any system events posted by any component in the attention group.

The default attention groups that you can monitor are:

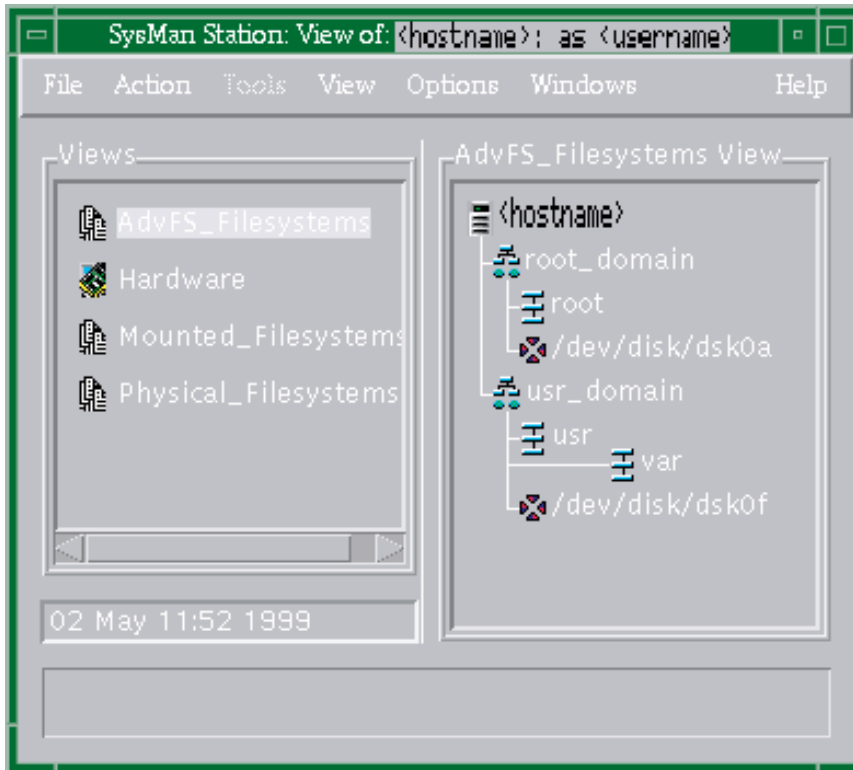
- File systems – Any UFS file systems or AdvFS domains.
- Network – The network and devices connected to the local host, such as `tu0`.
- Storage – Storage devices connected to buses and device interfaces, such as `floppy`, the floppy drive unit that is connected to an `fdi` interface such as `fdi0`.

### 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:

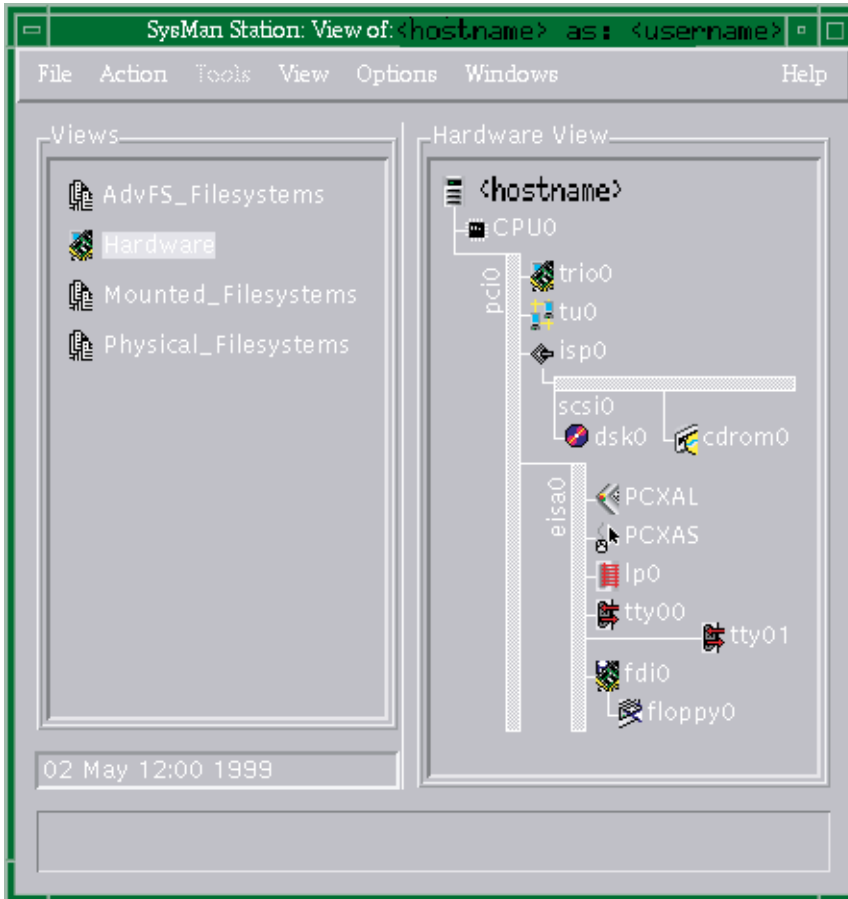
- `AdvFS_Fileystems` – A view of all AdvFS domains. Figure 1–8 shows a typical AdvFS domains view on a small single-disk system.

**Figure 1–8: AdvFS\_Fileystems 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_Fileystems` – A view of file systems that are currently mounted, in a similar format to the `AdvFS_Fileystems` view.
- `Physical_Fileystems` – A view of all (UFS, AdvFS) file systems available, in a similar format to the `AdvFS_Fileystems` view.

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

In any of the system component screens, you can click MB1 on any component to select individual system components and expand or collapse sections of the display hierarchy. On selecting a component, MB3 displays

a menu that contains one or more of the following options (depending on whether an option applies to the object that is selected):

- Display hierarchy functions:
  - Expand and Contract – These options display or remove the subcomponents under a component. For example, click on the Expand option when selecting a SCSI bus, and all its attached devices are 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` and all its attached devices are 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 you can launch 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 are 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 is 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 is blank if no utilities are applicable to the component, or if you select nothing.
- View – Allows you to control the current system view, and switch or cycle between views.
- Options – Allows you to control the appearance of the views, such as the icon size.
- Windows – Allows you to invoke other windows, such as the main window. You are prompted to save your custom view before you exit SysMan Station. You can then assign a name and an icon to the custom view. When you next invoke SysMan Station, your custom view is added to the Views pane.

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

## 1.9 Compaq Insight Manager

Compaq Insight Manager is a Web-based management utility, independent of operating environments. At present, some features are fully implemented in some operating environments, but are not yet implemented in others. This means that you can use all features on Windows NT systems, but you cannot use certain features on UNIX.

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

At present, the WBEM protocols (agents) are able to communicate between the following operating environments in addition to UNIX:

- Compaq OpenVMS
- IBM OS/2
- Microsoft Windows 2000, Microsoft Windows NT, Microsoft Windows 98, and Microsoft Windows 95

- Novell NetWare
- SCO OpenServer Release 5 and SCO UnixWare 7

Microsoft Windows on PC platforms is the only method considered in this guide, to give you an idea of the restrictions as they apply in an environment where UNIX systems serve PC clients. In future, all administrative tasks will be fully integrated and provide seamless Web-based enterprise management (WBEM) regardless of hardware platform or operating environment.

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

A device must have an operating environment that is recognized by Compaq Insight Manager so that you can manage it using WBEM. Such an operating environment must be able to communicate device information to the WBEM network, and to receive and execute instructions sent from other (authorized) devices in the WBEM network. The operating environments must be able to run the agents that communicate with each other using a standard protocol.

Devices, and their operating environments, provide information about hardware and software status using a data model, such as the Common Information Model (CIM), or a Management Information Base (MIB) and Simple Network Management Protocol (SNMP). You can think of these as a database of objects, with attributes and values, representing the manageable components of a device. There are industry standard MIBs and proprietary MIBs. Compaq 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 you can obtain (or manipulate).

In an environment consisting of client PCs and UNIX or Windows NT servers, you can use Compaq Insight Manager as your common interface to administrative tasks. For example, as an authorized (root) user working at your PC, you can invoke Compaq Insight Manager to view the general system status of an AlphaServer running UNIX, then invoke the 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 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 might run different operating systems with different disk and file management systems, such as AdvFS on UNIX and the FAT or NTFS file system on Windows NT. You might need to know how much space is available on disks, so that you can extend file systems before they run out of storage space. Compaq Insight Manager enables you to monitor such data at a single point, and (in some cases) you can use it to initiate corrective actions.

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

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

You can also specify the TCP/IP address, such as `20.111.333.10` in place of the host name and address. Once you connect 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 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 Compaq Insight Manager XE software to enable many of the remote administrative services on other Windows NT systems. The XE server has a port address of `:280`, to which you connect from your browser by specifying the URL as follows:

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

An 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 the SysMan Menu and the SysMan Station.

- 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 always has the lowest local TCP/IP network address (in the format `110.222.333.110`). It is important that you 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.

Compaq Insight Manager has an auto-discovery feature that allows an agent to discover any other 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.

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

- The Compaq 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 Compaq 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 Compaq Insight Manager on a single UNIX system, and connect via the Internet from your home PC to perform administrative tasks. The 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 remotely manage PC clients that are running Windows NT, or other systems that support an agent.
- Client systems, such as PCs running Windows NT, must have the 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. You can also reach this URL from the default home page on your UNIX system.
- PCs must be running the Internet Explorer Web browser. Netscape does not work.
- A designated master system must have the lowest TCP/IP address in the local network domain.

For information on configuring and using the WBEM agents, see `insight_manager(5)` the online documentation that is accessible from the Compaq Insight Manager home page.

## 1.10 Using SysMan on a Personal Computer

In addition to using java applets as described in `insight_manager(5)`, you can also install SysMan clients on a PC and launch them from the Windows Start Menu. This feature supports Microsoft Windows, MacOS, and Linux. Full information on this feature, together an address from which you can download the required software, is provided in a Web page available from the UNIX system. This page is located at

`http://<host>:2301/sysman/index.html`, where `<host>` is the host name and address or the TCP/IP address. The procedure 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 is 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

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

### 1.11.1 Setting Up a Console Port

The following sections provide an overview of the steps required to set up a serial line console port and set up the remote modem for dial-in. It is assumed that your local (dial-out) modem is 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. See `modem(7)` to obtain the correct modem settings and for instructions on how to create the appropriate system file entries. In particular, the `cons` entry in `/etc/inittab` file should be modified so that the `getty` or `uugetty` process sets up the `COMM` port correctly. This line is similar to the following example:

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

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

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

### 1.11.1.2 Setting the Configurable DCD Timer Value

You can configure the Carrier Detect (DCD) timeout value. 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.

Modify the timer by using 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 the value of `n` is 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.

If you change the baud rate, flow control, or modem setting (for example, using the `getty` command), the change is propagated down to the console level and the environment variables change automatically.

#### 1.11.1.4 Verifying the Modem Setup

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

### 1.11.2 Initiating a Console Port Connection

Using a terminal or X-terminal window, you can initiate a connection between the local and remote systems by using a command such as a `tip`, `kermit`, or `cu`. Alternatively, 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` is displayed.

See `tip(1)` 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. The following features might be useful for remote administration:

- The UNIX-to-UNIX system copy utility, `uucp` for copying scripts and files to the remote system. See `uucp(1)`.
- You can invoke the `ikdebug` kernel debugger and use it remotely. See `ikdebug(8)`. You might need to change an entry in the `/etc/remote`

file to correct the baud rate. For example you might 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 `syslog(1)`.

#### 1.11.2.1.2 Shutting Down the Remote System

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

```
# stty -hupcl
```

See `stty(1)`.

When the shutdown is complete, you 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.

#### 1.11.2.2 Troubleshooting the Modem Connection

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.  
See `modem(7)` 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. See `modem(7)` 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. See `modem(7)` 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 need to perform periodically. In some computing environments, it is important to keep the system running and available at all times, and to shut down intentionally only for scheduled maintenance or software upgrades.

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

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

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 to boot 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 the Shutdown and Boot Operations

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

### 2.1.1 Shutdown Methods

You can shut a system down automatically or manually. 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 enable you to shut down a local or remote system or cluster. The General Tasks branch of the SysMan Menu contains the task “Shutdown the System” that invokes the appropriate user interface, depending on how you access the SysMan Menu. You can also invoke the task from the command line by entering the following command:
 

```
# sysman shutdown
```

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

The Shutdown icon in the CDE Application Manager – DailyAdmin folder invokes the SysMan Menu task named “Shutdown the System”.

### 2.1.2 Boot Methods

You boot the operating system by using the system’s console. When a system is powered on, the symbol `>>>` indicates the console prompt. At this prompt, you enter commands or set system configuration variables, such as variables that control what happens when a system is booted. Throughout this chapter, the symbol `>>>` is referred to as the console prompt. The console is sometimes called the System Reference Manual (SRM) console or the

firmware console. 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 boot a remote system by using a network or modem connection, such as the remote console method documented in Chapter 1.
- You specify boot actions that happen after a shut down. For example, if you use SysMan Menu or the SysMan Station to initiate a shut down, you can set the system to reboot automatically to single user mode after the shutdown is completed.
- You can cause the system to boot automatically by setting the `auto_action` console variable. The system then boots 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 commands and variables. See `consvar(8)`, which describes `consvar`, a command that enables you to manipulate console environment variables from within the operating system, depending on the firmware revision.
  - 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 `sysman_station(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 online help is available:

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

See Chapter 1 for information on invoking online 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 you can use 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 to Section 1.11 in Chapter 1 for information on remote serial consoles if you administer systems at remote locations, or if there is a network failure that requires dial-up communications.
- 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` – The default name of the custom kernel. When you build a custom kernel, you can choose any legal file name.
- `/genvmunix` – The default name of the generic kernel. You boot the generic kernel to build a custom kernel, or if the custom kernel is corrupt and non-bootable.
- `/sbin/rc0`, `/sbin/rc2`, and `/sbin/rc3` – 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

You might also use the following utilities 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 gets, sets, lists, and saves console environment variables while the operating system is still running.

To see if your system supports `consvar`, use the following command:

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

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

## 2.2 Understanding the Boot Operation

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

### 2.2.1 Booting Automatically or Manually

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

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

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

### 2.2.2 Booting to Single-User or Multiuser Mode

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

Because the `init` operation does not invoke the startup script prior to turning control over to you, the root file system is mounted read only. Startup of the network and other daemons does not occur, file 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 the `init` operation continues with its startup tasks and boots to multiuser mode.

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

At the conclusion of the main initialization tasks (process 0), `init` (process 1) starts an additional set of tasks that includes reading the `/etc/inittab` file, acting on instructions found there, and executing the relevant `run` command scripts. These scripts contain entries that initiate activities such as mounting and 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 operating system allows you to boot an alternate kernel if your custom kernel is not bootable. You can boot the generic kernel (`/genvmunix`) to troubleshoot the problem with your system. You can also boot an alternate custom kernel to test new drivers or to add options to the existing kernel.

## 2.3 Preparing to Boot the Installed System

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

```
# mount -u /
```

---

### 2.3.1 Preparing to Boot a Powered-Down System

Follow these steps to power-up and boot your system:

1. Confirm that the hardware and all peripheral devices are connected. Refer to the operator's guide for your hardware for information and instructions for interpreting diagnostic output.
2. Power up peripheral devices. Refer to the operator's manual or the hardware user's guide for instructions on starting your peripheral devices.
3. Power up the processor.
4. 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.
5. Wait for the console prompt (`>>>`). If you enabled your system to boot automatically when it is powered up, press the halt button to display the console prompt. 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.
6. 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.
7. 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 superuser prompt (`#`), follow these steps to prepare to go to multiuser mode:

1. Decide if you need to continue in single-user mode or if you require multiuser mode:
  - If you have additional tasks to perform and you want the system to restrict access to all users but root, 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:

- If the diagnostic test indicates hardware failure, contact your field service representative. Because hardware damage is a serious problem, do not continue or try to bypass the defective hardware.
  - If you have enabled your system to boot automatically, press the halt button to display the console prompt. 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. 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 tries to start all the network services that are configured. This results in the boot process hanging, or taking a very long time to test for the presence of services. If you take a system out of a network without unconfiguring the services, or if a system crashes and you must disconnect it from the network, perform the additional steps before rebooting the system.

You might also want to remove a functioning system from a network, for example:

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

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

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

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

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

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

```
>>> boot
```

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

```
# mount -u /
```

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

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

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

---

#### Note

---

The integrity of the `/etc/rc.config`, `/etc/rc.config.common` and `/etc/rc.config.site` files is important for startup operations and for system configuration. Avoid modifying these files with anything other than the `rcmgr` command. If the format of the files is not correct, other subsystems or utilities might not parse the files correctly. See `rcmgr(8)` 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 complete the modifications, halt the system and reset any console environment variables. For example:

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

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

There are variations in the console commands depending on your system model and the firmware revision. 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.

This section provides examples of typical console settings. Refer to your hardware documentation that came with your system for specific information. See also the information on booting systems in the *Installation Guide* and *Installation Guide — Advanced Topics*.

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

**Table 2–1: Console Environment Variables**

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

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

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

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

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

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

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

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

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

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

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

**Table 2–2: Options to the `boot_osflags` Variable**

Option	Action
a	Boot to multiuser mode. (By default, the kernel boots to single-user mode.)
k	Use the <code>kdebug</code> debugger to debug the kernel. Refer to the <i>Kernel Debugging</i> 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 prompts are displayed). Refer to Section 2.4.3 for an example of an interactive boot.

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
```

- c. Set the default boot device.

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

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

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

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

For example:

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

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

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

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

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

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

The system displays informational messages.

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

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

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

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

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

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

To override the `bootdef_dev` console 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 the `boot_osflags` console variable.

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

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

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

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

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

- Overriding the `boot_file` console variable.

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

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

### 2.4.3 Using Interactive Boot to Verify the Root File System

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

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

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

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

2. The following message is displayed:

```
UNIX Boot - date
```

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

You options at this point are as follows:

- a. Enter the name of an alternate kernel and specify required boot options. Refer to the owner's manual for your system for a list of boot options.
  - b. Enter the following command to obtain help on the `ls` command:

```
# help
```

The `ls` command options are described in Step 3 below.
  - c. Enter the `quit` command to return to the console prompt.
  - d. Press Return to boot the default custom kernel (`/vmunix`) if no other kernel is specified by the `boot_file` console variable.
3. Use the `ls` command to list the content of root file system directories or to list specific files. If you do not specify a file name, the entire content of the directory is displayed. The following are examples of valid commands:

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

```
# ls /
```



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

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

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

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

## 2.5 Identifying System Run Levels

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

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

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

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

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 are not terminated in response to a `kill` signal from the `init` process.

When the system is initialized for the first time, it enters the default run level that is defined by the `initdefault` line entry in the `inittab` file. The system continues at that run level until the `init` process receives a signal to change run levels. The following sections describe these signals and provide instructions for changing run levels.

### 2.6.1 Changing Run Levels in Single-User Mode

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

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

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

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

### 2.6.2 Changing Run Levels from Multiuser Mode

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

Run Level	System State
0	Specifies the halt state.
2	Specifies a multiuser run level with local processes and daemons.
3	Specifies a multiuser run level with remote processes and daemons.

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

### 2.6.2.1 Changing to a Different Multiuser Run Level

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

```
# init 3
```

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

### 2.6.2.2 Changing to Single-User Mode

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

```
# init s
```

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

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

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

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

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

```
# /sbin/reboot
```

### 2.6.2.3 Reexamining the inittab File

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

```
# init q
```

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

See `getty(8)` for 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.

### 2.7.1 Adding CPUs to an Existing System

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

### 2.7.2 Unattended Reboots on Multiprocessor Systems

If a processor in a multiprocessor system fails, the operating system records which processor failed, then automatically reboots the system. Although the operating system continues, you must 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 might need to synchronize system time with standard time.

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

---

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

---

For example, to set the date to 09:34:00 a.m. 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 change the year, update the system disk with the new year information. In single-user mode, enter the `mount -u /` command after you enter a date containing a new year. This command writes the new year into the superblock on the system disk. The root file system is mounted read-write.

## 2.9 Troubleshooting Boot Problems

If your system does not boot, the following 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:

- You specified an incorrect boot path.

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 is corrupt, boot the generic kernel (`/genvmunix`). This provides you with a fully functional system and you can begin debugging procedures by using the `kdbx` or `dbx` utilities to analyze crash dumps. Refer to `kdbx(8)` or `dbx(1)` 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 the `fsck` process finds something wrong, you are prompted to choose a recovery option. Use extreme care under these circumstances so that you do not inadvertently overwrite or remove any files. See `fsck(8)` for more information.

If you have an Advanced File System (AdvFS), disk corruption is very unlikely. AdvFS provides disk recovery during the mount procedure that corrects the disk structures. You do not need to run the `fsck` command or any other command. Consequently, recovery of AdvFS is very rapid. 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 for both controlled and unexpected shutdowns. The first part discusses procedures for controlled shutdowns. The second part discusses guidelines and recommendations for recovering from unexpected shutdowns.

Typical reasons for shutting down a system are:

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

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

A system that has panicked or crashed presents you with a different set of circumstances than a system that has shut down in an orderly fashion. This chapter discusses orderly shutdowns only. 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. See `shutdown(8)` for more information.

## 2.11.1 Using SysMan shutdown

Use the `sysman shutdown` command to invoke the SysMan Menu shutdown task. You can also invoke this interface 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.

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

The following options are available:

- Shutdown type – Use this option menu to select one of the following shutdown options:
  - Halt – Halt the operating system and display the console prompt
  - Reboot – Shut down and halt the system, then automatically reboot it
  - Single user – Shut down to single-user mode, displaying the superuser prompt (#)
  - Message only – Broadcast a message to all current system users without shutting down the system
- Minutes until shutdown – hold down mouse button 1 (MB1) and move the slider bar to select the elapsed time in minutes before the shutdown operation begins (the shutdown delay). The time is displayed adjacent to the bar. You can select from a range of 0–60 minutes by using the slider bar. In some user environments, such as on a character-cell terminal, the slider bar is not available and you type a number to specify the shutdown delay. In these interfaces you can specify a time greater than 60 minutes.
- Shutdown message – Type a message to users warning of the impending shutdown and requesting that they log out. This message, if any, is in addition to the message that is sent by default.

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

- Broadcast message to NFS clients – Check this box if you want to broadcast a message to remote users of local NFS-served file systems. If a remote user is connected to any file system that is exported by the local system, that user receives a warning of the impending shutdown.



To send such messages, ensure that the `rwall` daemon is running on the remote user's system.

- **Execute run-level transition scripts** – Check this box if you want to run the existing run-level transition scripts in the `/sbin/rc[N.d]/[Knn_name]` file. For example, `/sbin/rc0.d/K45.syslog`. See the `-s` option in `shutdown(8)` for more information.
- **Preshutdown Script** – Specify a path to a custom script that you want to run before the shutdown completes. The script is run at shutdown time and completes any tasks that you specify prior to shutting down the system. If your script (or any intermediate scripts that it calls) fails to complete successfully, the system might not shut down correctly.
- **Other options** – Check this box to enable options that make the shutdown faster:
  - **Fast** – Performs a fast shutdown, bypassing messages to users and NFS clients
  - **No disk sync** – Shuts down without synchronizing the disks by using the `sync` operation.

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

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

## 2.11.2 Shutting Down the System and Warning Other Users

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

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

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

```
# cd /
```

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

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

Example 2–1 shows a typical shutdown sequence.

## Example 2–1: A Typical Shutdown Sequence

---

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

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

System going down in 60 seconds
    ...Maintenance shutdown, please log off
System going down in 30 seconds
    ...Maintenance shutdown, please log off
System going down immediately
    ...Maintenance shutdown, please log off [5]

.
.  process shutdown messages [6]
.
Halting processes ...
INIT: SINGLE USER MODE [7]
# halt
.
. <hardware reset messages> [8]
.
resetting all I/O buses
>>> [9]
```

---

- [1] This command initiates a shutdown, delayed for six minutes, and broadcasts a message to all users warning them to log off.
- [2] This message is immediately echoed to the console terminal, and to the terminal window from which you invoked the `shutdown` command.
- [3] These messages are immediately echoed to the console terminal, and to the terminal window from which you invoked the `shutdown` command. The messages are repeated at intervals, depending on the length of the original shutdown delay, becoming more frequent as shutdown time approaches.
- [4] When five minutes remain, new logins are automatically disabled. If anyone attempts to login at this time, this message is displayed at the log in terminal and it is not broadcast to other users.
- [5] This final message warns that the system is shutting down immediately and user processes are halted. The system stops processes such as accounting and error logging and logs the shutdown in the log file.

It then sends the `init` program a signal that causes the system to transition to single-user mode.

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

- 6 As processes are stopped, notification messages are displayed to the console and are logged.
- 7 As the system halts, all login terminals (or graphical displays, such as CDE and XDM) are halted, and output is redirected to the console. Various system messages are displayed at the console as processes are shut down and the shutdown ends in single-user mode, displaying the superuser prompt (`#`). Only the root user can 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 `shutdown` command to shut down and halt the system. For example, to shut down and halt the system in 5 minutes with a warning to users that the system is going down for maintenance, enter:

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

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

### 2.11.4 Shutting Down and Automatically Rebooting the System

Use this procedure to shut down the system from multiuser mode, warn all users, and automatically reboot the system to multiuser mode. You 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 `shutdown` to initiate a shut down followed by an automatic reboot. For example, to shut down and automatically reboot the system in 15 minutes with a warning to users that the system is going down for a reboot, enter the following command:

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

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

---

**Note**

---

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

---

### 2.11.5 Shutting Down and Halting Systems Immediately

Use the following procedure to shut down and halt the system immediately. 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. For example, enter the following command:

```
# cd /
```

2. Enter the `shutdown` command as follows:

```
# shutdown -h now
```

The system begins to shut down as described in Example 2–1 except that the shutdown is immediate and without prior warning to users. When all processes are shut down, the system is halted and the console prompt (`>>>`)

is displayed. You can turn off power to the system, reboot it, or use the console commands as described in the owner's manual for your system.

---

**Note**

---

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

---

## 2.12 Stopping Systems While in Single-User Mode

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

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

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

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

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

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

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

### 2.12.1 Stopping and Rebooting Systems with the `reboot` Command

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

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

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

```
# reboot -q
```

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

### 2.12.2 Stopping Systems with the `fasthalt` Command

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

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

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

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

### 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, as follows:

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

When you invoke the `fastboot` command without options, it creates a file named `/fastboot`, halts the system, then immediately reboots the system without checking file systems by using the `fsck` command. For a description of the `fastboot` command, see `fastboot(8)`.

# 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 might have already performed some of these tasks. As your system needs change, you might 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 might 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.

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

- The *Technical Overview* briefly describes the security components of the operating system.

- The *Security* guide is the principal source of security-related information for users, administrators, and programmers dealing with the security components.
- The *Network Administration: Connections* and *Network Administration: Services* guides are the principal sources of information for customizing the system's networking components.

### 3.1 Identifying and Modifying the System Initialization Files

To define and customize the system environment, you modify certain initialization files that specify and control processes and run levels. The operating system provides you with default files that define the available run levels and the processes associated with each run level. You can 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 `init(8)` for a description of the program and its behavior.

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

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

`/etc/inittab`

One of the key initialization files whose entries define run levels and associated processes and administer terminals. Section 3.1.1 describes this file.

`/etc/securettys`

A text file that marks whether a given terminal (`tty`) line allows root logins. Section 3.1.1.6 describes this file.



`/sbin/bcheckrc`

A system initialization run command script associated with 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/rcn .d`

The `/sbin` directory contains a set of individual subdirectories that correspond to the various run levels. Each subdirectory contains linked files that the system acts on when starting or changing a particular run level. There are three `/sbin/rcn .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/rcn`

These are the run command scripts that correspond to a particular run level. There are three `/sbin/rcn` scripts available: `/sbin/rc0`, `/sbin/rc2`, and `/sbin/rc3`. Section 3.1.2.2, Section 3.1.2.3, and Section 3.1.2.4 describe the contents and use of these scripts.

`/etc/rc.config` and `/etc/rc.config.common`

This is a file that contains run-time configuration variables. Scripts in the `/sbin/init.d` directory use these variables to configure various subsystems (for example, NFS or NTP). You (or a program) can use the `rcmgr` command to define or access variables in the `/etc/rc.config` file. See `rcmgr(8)` and the *Network Administration* manual for more information.

`/etc/sysconfigtab`

This is the database file that contains information about dynamically configurable subsystems. Chapter 4 describes this file.

`/usr/sbin/getty`

This is the executable file that sets and manages terminal lines. Section 3.1.1.3 and Section 3.1.1.4 describe this program. See `getty(8)` for more information.

`/etc/gettydefs`

The file used by `getty` that contains entries to identify and define terminal line attributes. See `gettydefs(4)` for more information.

`/var/spool/cron/crontabs/*`

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

`/var/spool/cron/atjobs/*`

This is a file that contains entries to identify and define the once-only activation of specific processes. See `at(1)` for more information.

The following files contain information on kernel configuration:

`/usr/sys/conf/NAME`

This is a text file that defines the components that the system builds into your configuration. The `NAME` variable usually specifies the system name. Chapter 4 describes this file.

`/usr/sys/conf/NAME .list`

The optional configuration file that stores information about the layered product subsystems and is used to automatically configure static subsystems. The `NAME` variable usually specifies the system name. Chapter 4 describes this file.

`/usr/sys/conf/param.c`

The text file that contains default values for some tunable system parameters used in building the system's kernel. Chapter 4 describes this file.

### 3.1.1 Using the `/etc/inittab` File

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

The operating system provides you with a basic `/etc/inittab` file that contains line entries for the most common and necessary initialization

processes. For example, the `/etc/inittab` file available with the distribution software would look similar to the following:

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

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

*Identifier: Runlevel: Action: Command*

*Identifier*

This 14-character field uniquely identifies an object entry.

*Runlevel*

This 20-character field defines the run levels in which the object entry is to be processed. The `Runlevel` variable corresponds to a configuration of processes in a system. Each process spawned by the `init` command is assigned one or more run levels in which it is allowed to exist. The run levels are as follows:

---

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

---

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

### *Action*

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

#### `respawn`

If the process does not exist or dies, `init` starts it. If the process currently exists, `init` does nothing and continues scanning the `inittab` file.

#### `wait`

When `init` enters a run level that matches the run level of the entry, it starts the process and waits for its termination. While `init` continues in this run level, it does not act on subsequent reads of the entry in the `inittab` file.

#### `initdefault`

A line with this action is processed when `init` is first invoked. The `init` program uses this line to determine which run level to enter. To do this, it takes the highest run level specified in the run-level field and uses that as its initial state. If the run-level field is empty, this is interpreted as `0s23`, so `init` enters run level 3. If `init` does not find an `initdefault` line in the `inittab` file, it requests an initial run level from the operator.

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

### *Command*

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

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

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

The following sections provide information 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, either the `getty` program or the `xm` program must run. These programs set up a process that runs the login and shell programs for each terminal or workstation. Because a large portion of your initial work is done at the system console, the `/etc/inittab` file contains an entry for setting up a `getty` process for the console. The `xm` process is started by a run-level script in the `/sbin/rc3.d` directory.

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

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

The `init` program is instructed to invoke the `getty` program, which sets the terminal line attributes for the system console (`/dev/console`). The run-level field specifies that the `getty` process should execute at run levels 1, 2, 3, and 4. The `respawn` keyword tells `init` to 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 `terminfo(4)` for specific details on creating and compiling source files.

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

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

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

#### 3.1.1.5 Specifying Process Run Levels

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

```

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

```

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

### 3.1.1.6 Securing a Terminal Line

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

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

```

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

```

## 3.1.2 Using the `init` and `rc` Directory Structure

The operating system provides you with an initialization and run command directory structure. The structure has four main components: the `init.d`, `rc0.d`, `rc2.d`, and `rc3.d` directories. In addition, each of the `rcn.d` directories has a corresponding `rcn` 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 `rc0(8)` for more information.

### 3.1.2.3 The `rc2.d` Directory and `rc2` Run Command Script

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

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

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.

See `rc2(8)` for more information.

### 3.1.2.4 The `rc3.d` Directory and `rc3` Run Command Script

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

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

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.

See `rc3(8)` for more information.

### 3.1.3 Using the crontabs Directory

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

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

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

Each entry has the following syntax:

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

```
% crontab temp_adm
```

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

You can use the `/usr/spool/cron/crontabs/root` file to back up and clean system log files. The `root` crontab file `/usr/var/spool/cron/crontabs/root` contains a model entry to clean up the `/var/adm/wtmp` log file at 2:00am every Sunday. One compressed backup of the log file is retained until the next cleaning. This crontab entry is enabled by default as follows:

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

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

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

If you want to preserve your log files for a longer period of time, you can either change the frequency of the cleanup or comment out the applicable `./crontabs/root` entry. You might 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.

See `crontab(1)` for more information.

## 3.2 Using National Language Support

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

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

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

- **Locale files** – These files contain month and day names, date formats, monetary and numeric formats, valid yes/no strings, character classification data, and collation sequences. These files reside in the `/usr/lib/nls/loc` directory.
- **Message catalogs** – These files contain translations of messages that are used by programs. These files reside in the `/usr/lib/nls/msg/locale-name` directory.

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

**Table 3–1: Locale Support Files**

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

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

Language/Territory	Locale Filename
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, see:

<code>code_page(5)</code>	Lists the coded character sets that are used on Microsoft Windows and Windows NT systems.
<code>iconv_intro(5)</code>	Provides an introduction to codeset conversion.
<code>iconv(1)</code>	Documents the command to convert encoded characters to another codeset.
<code>i18n_intro(5)</code>	Provides an introduction to internationalization (I18N).
<code>i18n_printing(5)</code>	Provides an introduction to internationalization (I18N) printer support.
<code>l10n_intro(5)</code>	Provides an introduction to localization (L10N).
<code>locale(1)</code>	Provides information about locales.

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

### 3.2.1 Setting Locale

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

To change the system-wide default locale for Bourne and Korn shell users, edit the `/etc/profile` file and include the name of the locale you want to be the system-wide default. 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
```

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 might 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 `f00`. Now suppose you reset `LANG` to a Spanish locale and then use the `grep` command for something in `f00`. 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. To make this option available, you must install at least one international support software subset. You can also launch this option from the CDE Application Manager. See Chapter 1 for information on using CDE.

### 3.4 Customizing Your Time Zone

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 be consistent with the city of New York on the American continent:

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

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

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
```

```
Mm.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:

```
:America/New_York
```

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

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

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

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

See `timezone(3)` for more information.

## 3.5 Customizing 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 are 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/sbin/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 might 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. You can assign the test database to a different class to prevent it from affecting your daily operations.

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 necessary to perform tasks in a specific sequence. There are two methods of accessing the class scheduler:

- |                     |  |
|---------------------|--|
| Manual              | By executing <code>class_admin</code> 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.<br><br>See Chapter 1 for information on running the SysMan Menu. Section 3.5.6 describes how you use the graphical interface. See 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)`
- `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. You can assign one or more applications or groups of applications 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

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.

After the database is established, you can enable class scheduling. This operation starts a class scheduling daemon and puts 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
- You can invoke the `iostat` and `vmstat` commands 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

The following sections focus on using the command-line method, and provide a brief introduction to using the graphical interface. See 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 might 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 your system is involved with critical real-time tasks that must take precedence over interactive processes, your course of action might 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.

Do not use the root account to create test processes when you configure class scheduling. Root account processes always take precedence over others, even when assigned to an existing restricted class.

### 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 a script invokes the `class_admin` command it uses the system defaults to configure the database automatically.

The following example shows an interactive configuration session using `class_admin`. 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 is not 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 you can specify larger values.

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 by using the appropriate identifiers such as UID or SESS.

### 3.5.4 Creating and Managing Classes

When the database is 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, see 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. 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

Unique system-assigned identifiers that the class scheduler recognizes (such as the PID, GID, or UID) identify which processes are members of a specific class. After you have created classes, you can add UIDs and GIDs or processes to one or more classes by 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   ttypl       0:01.30 -sh (csh)
root  12603 5176 12603  2908  1  R   + ttypl     0:00.05 ps aj

```

See `ps(1)` for more information and a definitive list of the process data items displayed when you use this command

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
```

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

See `class_admin(8)` 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. When you use the `runclass` command as root user (superuser) your processes have unrestricted access to CPU resources even when assigned to an existing class. By default, root processes are never restricted. This ensures that no user process can lock up resources needed by the root account. If you need to test a class scheduler configuration, ensure that you log in and create processes using a non-privileged user account. You might want to set up dummy user accounts to perform such testing.

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 is added to the class. You can use the following command to view the running process:

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

See `runclass(1)` 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`. See 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 by 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. 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 3.5.7. 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 was 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.

### 3.5.7 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. 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 are 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 starts and uses 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
```

You might need to spend some time monitoring tasks and system performance, tuning 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. Read the system owner's manual for information on whether your system supports power management. Power management utilities allow you to:

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

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

### 3.6.1 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 might 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) might be damaged by the activation of the DPMS feature. It is important that you check the specifications for your monitor in the owner's 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 `dpxpower` 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

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

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

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

### 3.6.3 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 might 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.

---

#### Note

---

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

If you do not observe any excessive use of swap space, and if you do not observe messages that specifically reference a lack of swap space, the problem might be a lack of per-process memory limits. See Section 3.7.5 for more information.

---

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

Virtual memory attempts to keep a process' most recently referenced virtual pages in physical memory. When a process references virtual pages, they are brought into physical memory from their storage locations on disk. Modified virtual pages can be moved to a temporary location on the disk (called swap space) if the physical pages (the pages in physical memory) that contain the virtual pages are needed by either a newly referenced virtual page or by a page with a higher priority. Therefore, a process' virtual address space can consist of pages that are located in physical memory, stored temporarily in swap space, and stored permanently on disk in executable or data files. Virtual memory operation involves:

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

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

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

Swapping involves moving a large number of virtual pages between physical memory and disk. The operating system requires a certain amount of physical memory for efficient operation. If the number of free physical pages drops below the system-defined limit, and if the system is unable to reclaim enough physical memory by paging out individual virtual pages or clusters of pages, the operating system selects a low priority process and reclaims all the physical pages that it is using. It does this by writing all 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.
- See `swapon(8)` and `swapon(2)` for information on 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. See `diskconfig(8)` 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. See `dxkerneltuner(8)` 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. See `sysconfig(8)` for information on modifying system attributes.
- `/sbin/disklabel` – This command-line interface can be used to modify kernel swap attributes in the system configuration file. See `disklabel(8)` for information on labeling a disk (otherwise known as formatting disk partitions).

---

#### 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. See 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 might want to add swap space to improve system performance or if you added more physical memory to your system. A cue to increase swap space is provided

by system console warning messages, stating that available swap space is depleted. Before adding swap space, check that any sudden lack of space is not due to a system problem. Use the following command to ensure that runaway processes or unusual user activities are not using up swap space:

```
# ps agx
```

(Alternatively, you can examine system log and event files for swap error messages.) If the resulting list of processes looks normal, you might 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 might 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
```

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

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

### 3.7.3 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 can result in one or more processes being involuntarily terminated.

For more information on virtual memory, See 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 might receive the following boot time informational messages when you switch to deferred mode or when you boot a system that is using the deferred method:

```
vm_swap_init: warning sbin/swapdefault swap device not found
vm_swap_init: in swap over-commitment mode
```

### 3.7.5 Correcting an Apparent Lack of Swap Space

There limits on the amount of virtual memory that an individual process can use. These limits are not related to the total amount of available swap space. Consequently, you might see error messages stating that a process has run out of virtual memory even though a swap monitor (such as the `dxsysinfo` utility), does not display a swap shortage. In some cases, a process might be automatically killed when it exceeds its allotted virtual memory.

If there is an actual shortage of swap space, you will see one of the following messages:

- The following message is displayed whatever swap mode you are using:

```
swap space below 10 percent free
```

- If you are using lazy swap mode, you might see a message similar to the following:

```
process (pid = 818) killed because of no swap space
```

- If you are using eager swap mode, you might see a message similar to the following:

```
Unable to obtain requested swap space
```

Use the following command to verify that your swap space is adequate:

```
# swapon -s
```

The data field titled `In-use space`: informs you if you are using most of your available swap space.

If you are not using all your available swap space but you are observing problems running large processes, you might need to assign more resources to the process. There are several kernel attributes in the `proc` subsystem that you can use to control the per-process virtual memory resources:

- **Stack limit** - The `per-proc-stack-size` and `max-per-proc-stack-size` attributes.
- **Data limit** - The `per-proc-data-size` and `max-per-proc-data-size` attributes.
- **Address space** - The `max-per-proc-address-space` and `per-proc-address-space` attributes.

If you encounter problems that appear to be due to a lack of memory, consider the following options:

- See the reference page that describes your command shell, such as `ksh(1)`. Command shells have a `limit` or `ulimit` option that enables you to modify the virtual memory resources for a process.
- Use the `sysconfigdb` command or the `dxkerneltuner` GUI to modify the value of the per-process resource limits in the `/etc/sysconfigtab` file. Instructions for modifying kernel attributes are provided in Chapter 4.
- See the *System Configuration and Tuning* guide for information on tuning virtual memory (`vm`) subsystem attributes, such as `vm-maxvas`. All subsystem attributes are documented in `sys_attrs(5)` and related reference pages.



---

## 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: Connections* and *Network Administration: Services* guides provide 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 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_attrsdli(5)`, `sys_attrs_dlp(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 specifying the new kernel.

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 perform on the `max-vnodes` attribute. The following list describes the values that can appear in this field:

- **C** – You can modify the attribute when the subsystem is 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

Use the `/sbin/sysconfigdb -u` command to update a subsystem that is already in the `/etc/sysconfigtab` database. For example, suppose the `generic` subsystem is defined as follows in the `/etc/sysconfigtab` file:

```
generic:
    lockmode = 4
    dump-sp-threshold = 6000
```

Suppose that you create a file named `update_attrs` for updating this entry, which contains the following information:

```
generic:
    lockmode = 0
    lockmaxcycles = 4294967295
```

To update the attributes, you enter the `sysconfigdb` command, as follows:

```
# /sbin/sysconfigdb -u -f update_attrs generic
```

After the command finishes, the entry for the `generic` subsystem in the database appears as follows:

```
generic:
    lockmode = 0
    lockmaxcycles = 4294967295
```

#### 4.4.8.5 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.

---

### Caution

---

This procedure is risky and you should ensure that you have a copy of your custom `/vmunix` kernel file, a copy of the generic kernel `/genvmunix`, and copies of the current configuration files. You may need the copies to revert to your previous configuration.

---

If you are unsure of the keyword you need to add to the target configuration file for your system, connect the new device to the system as directed in the hardware manual and use the following procedure:

1. 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 already 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/lsmbootstrap
```

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/ttynn` 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:

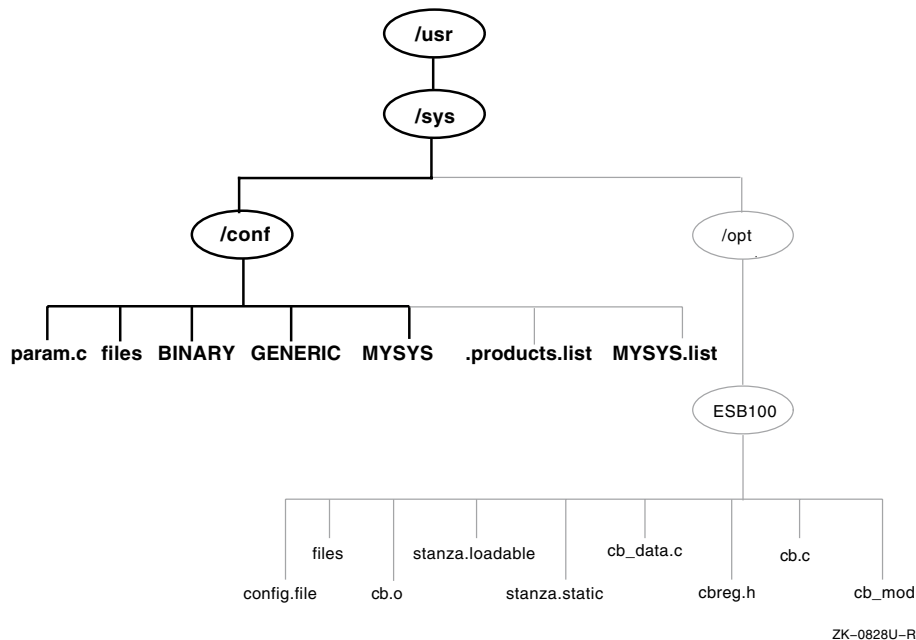
```
# shutdown -r now
```

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.

---

## Administering Hardware

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

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

Most operations require root user privileges but you can assign such privileges to nonroot users by using the SysMan division of privileges (DOP) feature. See `dop(8)` 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. This command provides full access to hardware management options.
- Section 5.5 describes how to use the `dsfmgr` command to manage device special files.
- Section 5.6 describes how to manually add components that you cannot add by using hardware manager, and how you create pseudodevices.
- Section 5.7 describes targeted utilities for managing hardware.

The `hwmgr` command also enables you to hot swap CPUs. For information on this feature, see `hwmgr_ops(8)` and the *Managing Online Addition and Removal* guide.

## 5.1 Understanding Hardware

A hardware component is any discrete part of the system such as a CPU, a networking card, or a hard disk. The system is organized in a hierarchy with the CPUs at the top and peripheral components 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 might 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 often specify commands using the host name. The CPUs are the top of the system hardware hierarchy, and all other system components are organized under the CPUs.

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 might have a number of main internal communication buses, which transfer data between components of the system. Adapters and controllers are physically plugged into buses and have both physical and logical addresses.

Buses might 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 components are usually automatically recognized and added to the system configuration on reboot, but you might 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 components.

- Controllers and Adapters – A system might have a number of controllers such as SCSI controllers, which control one or more storage devices. There might be other controllers, such as the floppy disk interface (`fdi`) that support one kind of disk and usually have only one physical disk attached to the controller. A network adapter might be connected to a



bus, but does not have any other components below it in the hierarchy other than the network cabling.

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

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

Storage (and other) devices might be shared by components or members of a cluster. This means that a component might have different names and identifiers associated with it depending how you access the component. Understanding how to identify a component, and how that component appears to the rest of the hierarchy, is an important aspect of hardware management. You often need to know both logical and physical locations of components.

When referring to SCSI devices in this chapter, the SCSI disk is most frequently referenced as an example. It is often the target of hardware management tasks and might appear to the system as a single device, or as a group or array. For example:

Single SCSI Disk or RZ devices

The operating system supports storage devices that conform to the Small Computer System Interface (SCSI) interface technology. Not all SCSI devices closely conform to this standard and the system might not be automatically detect and add such devices. You might need to use `ddr_config` as described in Section 5.6 to add such devices

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 might support features such as hot-swapping, failover, and redundancy, and be connected to the system by fibre channel controllers. Such storage arrays can be shared between many systems in a storage area network.

You use applications such as the Storage Works Console (SWCC) to manage storage arrays and storage area networks. In such configurations, you can accomplish only a small proportion of your storage management tasks using features of the operating system, such as the `hwmgr` command. Consult your StorageWorks

documentation for complete information on how you configure and manage storage arrays.

See RAID(7), SCSI(7) and rz(7) for more information on device characteristics. See tz(7) for more information on tape devices. See 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 user interface that displays topological views of the system component hierarchy and allows you to manipulate such views.

The majority of hardware management tasks are automated. When you add a supported SCSI disk to a system and reboot the system, the disk is automatically detected and configured into the system. The operating system dynamically loads required drivers and creates the device special files. You need only to partition the disk and create file systems on the partitions (described in Chapter 6) before you use it to store data. However, you must periodically perform some hardware management tasks manually, such as when a disk crashes and you need to bring a replacement disk online at the same logical location. You might also need to manually add components to a running system or redirect I/O from 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 component itself, but you often need to perform tasks such as checking the system for the optimum (or preferred) physical and logical locations for the new component.

Another important aspect of hardware management is preventative maintenance and monitoring. Use the following operating system features to maintain a healthy system environment:

- The Event Manager (EVM) – An event logging system that filters system events and then notifies you of selected events. 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 user interface that enables you to view and monitor the entire system (or cluster) hardware and launch applications to perform administrative tasks on components. You can also launch these applications from the SysMan Menu, and some

example applications are described later in this chapter (see Section 5.3). For information on using the SysMan tasks, refer to Chapter 1.

- The system census tool, `sys_check` – This command 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`. See `insight_manager(5)` for more information.

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

- Generic hardware management tools – These tools enable you to perform operations on all components of a type, classes of component such as SCSI tapes, or individual components. The tools might in some cases operate on all systems in a cluster. An example of such a tool is the SysMan Station, which provides you with a graphical 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 component that enable any application to access its device driver or pseudodriver.
- Targeted hardware management tools – These tools enable you to perform operations that are targeted to a specific component and perform a specific task. An example is the disk configuration command line interface, `disklabel`, and the analogous graphical user interface, Disk Configuration (`diskconfig`), which enable you to partition a disk by using the standard layouts or your own custom layouts.

Another way to think of this is that with a generic tool you can perform a task on many components, while with a targeted tool you can perform a task on only a single component. Unless stated, most operations are specific to a single system or to a cluster. See the TruCluster documentation for additional information on managing cluster hardware.

## 5.2 Reference Information

The following sections contain reference information related to documentation, system files, related software tools. Some tools described here are obsolete and scheduled for removal in a future release. Consult the *Release Notes* for a list of operating system features that are scheduled for retirement and migrate to its replacement as soon as possible. Check your site-specific shell scripts for any calls that might invoke an obsolete command.

### 5.2.1 Related Documentation

The following documentation contains information hardware management:

- Guides (available online or hardcopy):
  - Device documentation – Consult the device documentation for information on installing the device and for any required operating system or configuration settings.
  - *Network Administration: Connections* and *Network Administration: Services* – Provide information on configuring or connecting network components.
  - Device Driver Documentation Kit – Contains related documents such as: *Writing PCI Bus Device Drivers* and *Writing Device Drivers: Reference*.
  - *Logical Storage Manager* – The Logical Storage Manager (LSM) consists of physical disk devices, logical entities, and the mappings that connect them. Refer to this document for information on LSM concepts and commands.
- Reference pages:
  - `hwmgr(8)` – Summary information on the syntax and usage of the hardware manager command, `/sbin/hwmgr`.
  - `hwmgr_ops(8)` – System operation options for the `/sbin/hwmgr` command. Use these options to perform procedures such as CPU hot swap.
  - `hwmgr_show(8)` – Hardware information options for the `/sbin/hwmgr` command. Use these options to display information from the hardware databases.
  - `hwmgr_get(8)` – Component attribute information options for the `/sbin/hwmgr` command. Use these options to obtain and configure component attributes.
  - `hwmgr_view(8)` – Status information options for the `/sbin/hwmgr` command. Use these options to view component and system status.

- `dsfmgr(8)` – Contains complete information on the command syntax for the device special file management command. Use this command to create device special files in the `/dev` directory. Refer also to Section 5.5.
- `mknod(8)`, `MAKEDEV(8)`, `scu(8)`, `ddr_config(8)`, and `devswmgr(8)` – Reference pages that cover miscellaneous commands and utilities that you might use while administering devices.

The command line and graphical user interfaces also provide extensive online help.

## 5.2.2 Identifying Hardware Management System Files

The following system files contain static or dynamic information that the system uses to configure the component into the kernel. Do not edit these files manually even if they are ASCII text files. Some files are context-dependent symbolic links (CDSLs), as described in Chapter 6. If the links are accidentally broken, clustered systems cannot access the files until the links are recreated.

---

### Note

---

Although some hardware databases are text format, you must not edit the databases. Use only the appropriate command.

---

- The `/dev` directory contains device special files. Refer to Section 5.5 for more information.
- `/etc/ddr_dbase` – The device dynamic recognition (DDR) device information database. The content of this file is compiled into the binary file `/etc/ddr.db`, which the system uses 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 to `dev_t` are stored in this database.
- `/etc/dec_scsi_db` – This is a binary database owned by SCSI/CAM. It stores the worldwide 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 highest hardware identifier (HWID) assigned to a hardware component. The operating system uses this database to generate the next HWID that the system automatically assigns to a newly-installed hardware component. The system never reuses an HWID. For example, assume you add a disk to a system and it is assigned an HWID of 124. Even if you remove that disk permanently from the system, the HWID 124 is never reassigned to its replacement disk or to any other device. The only way that you can reset the HWID numbering sequence is to perform a fresh installation of the operating system.

### 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 worldwide 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 disks (and even some new disks 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 attempts to generate a WWID, and in the extreme case uses the device nexus (its SCSI bus/target/LUN) to create a WWID for the device.

Consequently, do not use devices that do not have a unique WWID 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 is created for each different path to the device. This can lead to data corruption if the operating system uses two different device special files to access the same 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 you can place it on a shared bus. Such devices are referred to as “cluster-shareable” because you can put them on a shared bus within a cluster.

---

#### Note

---

Exceptions 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. See the owner’s manual for the HSZ device and check the *Release Notes* for any current restrictions.

---

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

```
# hwmgr show comp -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  ..."
```

You might have a requirement to make a device available on a shared bus even though it does not have a unique WWID. Using such devices on a shared bus is not recommended, but there is a method that enables you to create such as configuration. See Section 5.4.4.10 for a description of how you use the `hwmgr edit scsi` command option to create a unique WWID.

## 5.2.4 Related Commands and Utilities

The following commands are also available to you for use in managing devices:

- The system exerciser utilities enable you to test devices for correct operation. See `diskx(8)`, `tapex(8)`, `cmx(8)`, `fsx(8)`, and `memx(8)`. See also Chapter 12.
- The `scu` command enables you to maintain and diagnose problems with SCSI peripherals and the CAM I/O subsystem. See `scu(8)` and the online help for the command.
- Use the `sysconfig` command 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 the `sysconfig` command to set some component attribute values. For information on using the `sysconfig` command, refer to Chapter 4, which also documents the Kernel Tuner (`dxkerneltuner`). The Kernel Tuner is a graphical user interface that you can also 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 user interfaces that you use to configure hardware such as ATM, Disk devices, Network devices, PPP (modem) devices, and LAT devices.
  - DailyAdmin – A graphical user interface for power management, which you use to set power attributes for certain devices.
  - SysMan Checklist, SysMan Menu, and SysMan Station – Provide interfaces to configure, monitor, and maintain system devices. You can invoke the SysMan Menu and SysMan Station 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. See Chapter 1 for more information.

## 5.3 Using the SysMan Hardware Tasks

The SysMan Menu provides tasks that you can use for basic hardware management. You can also use the SysMan Station to obtain information about hardware components and to launch hardware management tasks.

The SysMan tasks 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 is located in Section 5.4. See `hwmgr(8)` for a complete listing of the command syntax and options. Selecting the help option in one of the SysMan Menu hardware tasks invokes 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
- Manage CPUs
- Online Addition/Replacement (OLAR) policy information



These tasks launch basic hardware management tasks that are described in the following sections. See *Managing Online Addition and Removal* for information on online addition and removal (OLAR).

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

- Rerun – Runs the command again, updating the information in the display.
- Stop – Stops the command. Use the Rerun option to update the information or choose OK to exit.
- 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 `/sbin/hwmgr view hierarchy` command. 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
display truncated
```

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

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

- **HWID** – The hardware identifier (or `id`), an integer that is unique to each individual entry in the hierarchy.
- The component name, such as `pci` for the Peripheral Component Interconnect (PCI) bus.
- The component basename, a mnemonic followed by an integer that identifies the component such as `cdrom0`, which relates to the device special file for the component (`/dev/disk/cdrom0`). More information on device special file names is located 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 component is attached.
  - `targ-0` is the target number for this component on the bus, in this case the first target on bus 0.
  - `LUN-0` is the logical unit number (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 component relationships, such as the following sample output 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
```

Because the same component might be shared (for example, on a shared bus) it might appear in the hierarchy more than once and has a unique identifier each time it appears. An example of shared devices is provided 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 component attribute named `device_starvation_time` for the

component with the HWID (id) of 30. Component 30 is the SCSI disk 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 by using the following command option:

```
# /sbin/hwMgr set attr -id 35 -a device_starvation_time=30
```

Understand the impact of the changes before modifying the value of any component attribute. See the documentation provided with a device.

### 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:

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

You can specify the `Member ID` and the `HostName` 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 component information for the entire system or cluster. The output shows every component and pseudo-device (such as the `/dev/kevm` pseudo-device) that is connected to 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/hwmgrr view devices ...

/sbin/hwmgrr 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 component is any entity in the hierarchy that has the attribute `dev_base_name` and has an associated device special file (DSF). The output from this command provides the following information that you can use with the `hwmgrr` command to perform hardware management operations on the device:

- **HWID** – The hardware identifier (or `id`), an integer that is unique to each 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 shows 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 component is attached, in this case, it is SCSI bus 0.
  - `targ-0` – The target number for this component on the bus, in this case the first target on the bus.
  - `LUN-0` – The logical unit number, 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:

```
# /sbin/hwmgr flash light -dsf /dev/disk/dsk3 -nopause
```

The preceding command does not work for CD-ROM readers or for disks that are part of a managed array, such as an HSZ80.

### 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 task describes the processor and its status:

```
                                /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 user interface that runs under various windowing environments or from a web browser. See 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 are notified when a hardware problem occurs on the system by color changes to icons displayed by the GUI. System views are hierarchical, showing the complete system topology from CPUs down to discrete components such as tapes. You can observe the layout of buses, controllers, and adapters and see their logical addresses. You can see what components 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 component and view detailed attributes of that device. For example, if you select a SCSI disk and press the right mouse button, a menu of options is displayed. You can choose to view the component properties for the selected disk. If you opt to do this, an extensive table of component properties is 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 tasks

When you select a device, you can also choose to launch a command 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 component are posted.

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

## 5.4 Using `hwmgr` to Manage Hardware

The principal command that you use to manage hardware is the `hwmgr` command line interface (CLI). Other interfaces, such as the SysMan tasks provide a limited subset of the features provided by `hwmgr`. For example,

you can use `hwmgr` to set an attribute for all components 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 component so that the replacement component 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 for the `hwmgr` command
- Performing administrative tasks by using the `hwmgr` command

### 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. Application requests are passed by library routines to 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 by using the `dsfmgr` command.

The hardware set consists of data structures that describe all of the hardware components that are part of the system. A hardware component becomes part of the hardware set when registered by its driver. Many components support attributes that describe their function and content or control how they operate. Each attribute is assigned a value. You can read, and sometimes manipulate, these attribute values by using the `hwmgr` command.

The system hardware is organized into three parts, identified as subsystems by the `hwmgr` command. 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 components specified in the (binary) `/etc/dec_hwc_ldb` and `/etc/dec_hwc_cdb` databases. This includes most components 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 `hwmgrr` are as follows:

- It provides a wide range of hardware management functions under a single command.
- It enables you to manage (to a small extent) hardware components that are currently not connected to your system but were seen on a previous boot.
- It enables you to manage hardware components that are connected to multiple systems in a cluster.
- It enables you to propagate a management request to multiple members of a cluster.

## 5.4.2 Understanding `hwmgrr` Command Options

The `hwmgrr` command works with the kernel hardware management module, providing you with the ability to manage hardware components. Examples of a hardware component are storage peripherals, such as a disk or tape, or a system component such as a CPU or a bus. Use the `hwmgrr` command to manage hardware components on either a single system or on a cluster.

Operational commands are characterized by a subsystem identifier after the command name. The subsystems are: `component`, `scsi` and `name`.

Some `hwmgrr` operation commands are available for more than one subsystem. You should use the subsystem most closely associated with the type of operation you want to perform, depending on the parameter information that you obtained using the `view` and `show` command options.

Some commands require you to specify a subsystem name. However, if you specify the identity of a hardware component then you do not need to specify a subsystem name. The `hwmgrr` command is able to determine the correct subsystem on which to operate, based on the component identifier.

The command options are organized by task application. The list of command options, the subsystems on which they operate, and the nature of the operation is shown in the following table:

Option	Subsystem	Operation
<code>add</code>	<code>name</code>	Database management
<code>delete</code>	<code>component</code> , <code>name</code> , and <code>scsi</code>	Database management
<code>edit</code>	<code>name</code> , <code>scsi</code>	Database management



Option	Subsystem	Operation
locate	component	Hardware configuration
offline	component, name	Online Addition and Removal
online	component, name	Online Addition and Removal
power	component, name	Online Addition and Removal
redirect	scsi	Hardware configuration
refresh	component, scsi	Database management
reload	name	Driver configuration
remove	name	Database management
scan	component, name, and scsi	Hardware configuration
status	component	Hardware configuration
unconfigure	component, name	Hardware configuration
unindict	component	Online Addition and Removal
unload	name	Driver configuration

### 5.4.3 Configuring the hwmgr Environment

The `hwmgr` command provides environment settings that you can use to control the amount of information displayed. Use the following command to display the default environment settings:

```
# hwmgr view env

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

You can set the value of environment variables in your login script, or at the command line as shown in the following example:

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

You usually need to define only the value of the `HWMGR_HEXINTS`, `HWMGR_NOWRAP`, and the `HWMGR_VERBOSE` environment variables as follows:

- If the `HWMGR_HEXINTS` environment variable is defined as `TRUE`, any numerical data output from the `hwmgrr` command is displayed in hexadecimal numbers.
- If the `HWMGR_NOWRAP` environment variable is defined as `TRUE`, the output from the `hwmgrr` command is truncated at 80 characters. In some cases it is difficult to read the output from `hwmgrr` command options because it wraps. Setting the value of the `HWMGR_NOWRAP` environment variable to `TRUE` makes the output more legible at the console. A horizontal ellipsis marks truncated lines.
- If the `HWMGR_VERBOSE` environment variable is defined as `TRUE`, the output from the `hwmgrr` command contains more detailed information. The default setting of the `hwmgrr` command is to hide any errors that are not critical. To view more verbose information, you can also append the verbose switch to any of the `hwmgrr` command options.

For example, if you query an attribute that does not exist for all hardware components, by default the `hwmgrr` command displays only the output from hardware components that support the attribute, as shown in the following example:

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

Not all hardware components on the system support the `type` attribute. If the `HWMGR_VERBOSE` environment variable is not defined as `TRUE` the errors generated by the preceding command are suppressed. To see the errors, use the `-verbose` switch with the command line as follows:

```
# hwmgrr 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)

You can use the `verbose` switch 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 might need to perform by using the `hwmgr` command. Some of these examples might 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 the `hwmgr` command enables you to connect to an unfamiliar system, obtain information about its component 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

The `locate` option, which currently works only for some SCSI devices, enables you to identify a device. You might use this command when you are trying to physically locate a SCSI disk. The following command flashes the light on a SCSI disk for one minute:

```
# /sbin/hwmgr locate -id 42 -time 60
```

You can then check the disk bays for the component that is flashing its light. You cannot use this option to locate CD-ROM readers and disks that are part of an array (such as an HSZ80).

### 5.4.4.2 Viewing the System Hierarchy

Use the `view` command to view the hierarchy of hardware within a system. This command enables you to find what adapters are controlling devices, and discover where adapters are installed on buses. The following example shows typical output on a small system that is not part of a cluster:

```
# hwmgr view hier
HWID:  hardware hierarchy
-----
 1:  platform AlphaServer 800 5/500
 2:    cpu CPU0
 6:    bus pci0
 7:      connection pci0slot5
15:        scsi_adapter isp0
16:          scsi_bus scsi0
32:            disk bus-0-targ-0-lun-0 dsk0
33:            disk bus-0-targ-4-lun-0 cdrom0
34:            disk bus-0-targ-5-lun-0 dsk1
35:            disk bus-0-targ-6-lun-0 dsk2
```

```

36:          disk bus-0-targ-8-lun-0 dsk3
9:          connection pci0slot6
17:         graphics_controller s3trio0
output truncated

```

Some components might 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 shows two entries for the same device. You can obtain similar views of the system hardware hierarchy by using the SysMan Station GUI.

#### 5.4.4.3 Viewing System Categories

To perform hardware management options on all components of the same category, or to select a particular component in a category, you might need to know what categories of components are available. The hardware manager `get category` command fetches all the possible values for hardware categories.

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

```

# /sbin/hwmgr get category
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

```

Knowing the categories, you can focus your attribute query by specifying a category as follows:

```

# hwmgr get attr -category platform
1:
name = AlphaServer 800 5/500
category = platform

```

This output informs you that the system platform has a hardware ID of 1, and that the platform name is AlphaServer 800 5/500. See also the `get attribute` and `set attribute` command options.

#### 5.4.4.4 Obtaining Component Attributes

Attributes are characteristics of the component that might be read-only information, such as the model number of the device, or you might be able to set a value to control some aspect of the behavior of the device, such as the speed at which it operates. The `get attribute` command option fetches and displays attributes for a component. The hardware manager command is specific to managing hardware and fetches only attributes from the hardware set. All hardware components are identified by a unique hardware identifier, otherwise known as the hardware ID or HWID.

The following command fetches all attributes for all hardware components on the local system and directs the output to a file that you can search for information:

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

However, if you know which component category you want to query, as described 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 might not be sure if the network is working for some reason. You might 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 HWID for this Ethernet adapter.
- The fields and values listed below the HWID are the attribute names and their current values. Some values might 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 component name of `ln0`.
- You can then check the status of this network adapter by using the `ifconfig` command, as follows:

```
# ifconfig ln0
ln0: flags=c62 inet XX.XXX.XXX.XX netmask fffffff0 \
broadcast XX.XXX.XX.XXX ipmtu 1500
```

In some cases, you can change the value of a component attribute to modify component 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 `(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 components 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 components which support the `device_starvation_time` attribute. Reading the HWID in

the hierarchy output, you can determine 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 attributes. You cannot set all component attributes. When you use the `get` attribute command option, the output flags any configurable attributes by labeling them as (settable) next to the attribute value. A method of finding such attributes is described in Section 5.4.4.4.

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 component 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 component is still there, the driver posts 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 component 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 you cannot modify it. If you never set a value of an attribute, the default value applies. When you set the saved value, it persists across boots. You can think of it as a permanent override of the default.

When you set the current value, it 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 reverts to the saved value (if there is a saved value). If there is no saved value the attribute value reverts 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 get 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:

```

# /sbin/hwmgr view 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

```

You can also use this option to verify that the `hwmgr` command is aware of all cluster members and their current status.

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 by using the `-cluster` option are sent to members `bert` and `ernie`, but not to `bigbird` as that system is unavailable. Additionally, any `hwmgr` commands that you issue with the `-member bigbird` option fail because the cluster member state for that host is `DOWN`.

The `view cluster` command option works only 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 the `hwmgr` command to display all components that have a device special file name, such as `/dev/disk/dsk34` by using the `view devices` option. The hardware manager considers any hardware component that has the attribute `dev_base_name` to be an accessible device. (See Section 5.4.4.4 for information on obtaining the attributes of a device.)

The `view devices` option enables you to determine what components are currently registered with hardware management on a system, provides information that enables you to access these components through their device special file. For example, if you load a CD-ROM into a reader, use this output to determine whether you mount the CD-ROM reader as



/dev/disk/cdrom0. The view devices option is also useful to find the HWIDs for any registered devices. When you know the HWID for a device, you can use other hwmgr command options to query 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 output 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 the c partition. The c partition represents 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 components 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

```

Some devices, such as the disk with the HWID of 45:, appear more than once in this display. These are components that are on a shared bus between two cluster members. The hardware manager displays the component 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 must be synchronized across a cluster. The view transaction command option displays the state of any hardware management transactions that have occurred since you booted the system. Use this option to check for failed hardware management transactions.

If you do not specify the -cluster or -member option, the command displays status on transactions that are processed or initiated by the local host (the system on which the command is entered). The view transaction command option is primarily for debugging problems with hardware management in a cluster, and you are likely to use this command infrequently. The command has the following typical output:

```
# /sbin/hwmgr view transactions
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 describes a modification of the cluster database.

#### 5.4.4.9 Deleting a SCSI Device

Under some circumstances, you might want to remove a SCSI device from a system, such as when it is logging errors and you must replace it. 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 are removed from the appropriate `/dev` subdirectory. You cannot delete a SCSI component that is currently open. You must terminate all I/O connections to the device (such as mounts).

You might need to delete a SCSI component if you are removing it from your system and you do not want information about the component remaining on the system. You might also want to delete a SCSI component because of operating system, rather than hardware problems. For example, if the component operates correctly but you cannot access it through the device special file for some reason. In this case you can delete the component and use the `scan scsi` command option to find and register it.

To replace the SCSI component (or bring the old component back) you can use the `scan scsi` command option to find the component again. However, when you delete a component and then perform a `scan` operation to bring the component back on line, it does not always have the same device special file. To replace a component as an exact replica of the original, you must perform the additional operations described in Section 5.4.4.11. There is also no guarantee that a `scan` operation can find the component if it is not actively responding during the bus scan.

This option accepts the SCSI device identifier `-did`, which is not equivalent to the HWID. The following examples show how you check the SCSI database and then delete a SCSI device:

```

# hwmgr show scsi

```

SCSI HWID:	DEVICEID	HOST-NAME	DEVICE TYPE	DEVICE SUBTYPE	DRIVER OWNER	NUM PATH	DEVICE FILE	FIRST VALID 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 component for use. Therefore, you cannot delete SCSI component 23 because it is currently in use.

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 component is not currently accessible and you can delete it safely. When you delete the device, you also delete the /dev/disk/dsk8\* and /dev/rdisk/dsk8\* device special files.

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 HWID:	DEVICE ID	HOSTNAME	DEVICE TYPE	DEVICE SUBTYPE	DRIVER OWNER	NUM PATH	DEVICE FILE	FIRST VALID 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 component /dev/disk/dsk8 is successfully deleted.

#### 5.4.4.10 Creating a User-Defined SCSI Device Name

Most components have an identification attribute that is unique to the device. You can read it as the serial\_number or name attribute of a SCSI device. For example, the following hwmgr command returns both these attributes for the component with a HWID of 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 worldwide identifier (WWID) because it is unique for each component on the system.

Some components do not provide a unique identifier. The operating system creates such a number for the component by using valid path bus/target/LUN data that describes the physical location of the device. Because systems can share devices, each system that has access to the component sees a different path and creates its own unique WWID for that device. There is a possibility of concurrent I/O access to such shared devices, possibly resulting in data corruption. 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"
.
.
.
```

Some devices, such as the TL895 model media changer, do not support INQUIRY pages 0x80 or 0x83 and are unable to provide the system with a unique WWID). To support features such as path failover or installation into a cluster on a shared bus, you must manually add such devices to the system. This is the recommended method to add only media changers to a shared bus, it is not recommended for other types of devices such as disks, CD-ROM readers, tape drives, or RAID controllers. Other devices provide a unique string, such as a serial number, from which the system can create a unique WWID. You can use such a component on a shared bus because its WWID is always the same and the operating system always recognizes it as the same device.

You can use the `hwmgr` command to create a user-defined unique name that in turn enables you to create a WWID known to all systems that are sharing the device. Because the component has a common WWID it has one set of device special file names, preventing the risk of concurrent I/O.

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 component and differentiate it from others.

- Decide what component uses this name. When renamed, the component is seen as the same component on all systems. You must update the systems so that the component is seen.
- Each system that shares the component creates a new WWID by using the string and uses this new WWID for all subsequent registrations with the system. Internally, the component is still tracked by its default WWID (if one existed). However, all external representations 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**

---

You must update all systems that have access to the device.

---

The following example shows how you assign a user-defined name. Although the `edit scsi` command option is recommended only for devices that do not have a unique WWID, the example uses disks for the sake of simplicity.

```
# hwmgr show scsi
```

SCSI HWID:	DEVICEID ID	HOST NAME	DEVICE TYPE	DEVICE SUBTYPE	DRIVER OWNER	NUM PATH	DEVICE FILE	FIRST VALID 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 component is constructed by using the bus/target/LUN address information. Because the bus/target/LUN addresses are different, the component is seen as two separate devices. This can cause data corruption problems because the operating system might use two different sets of device special files to access the disk (`dev/disk/dsk1` and `dev/disk/dsk2`).

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

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

# hwmgr show scsi -did 2 -full

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

      WWID:0910003c:"DEC      (C) DECZG41400123ZG41800340:d01t00002100000"
      WWID:ff10000e:"this is a test"

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

You repeat the operation on the other component path and the same name is given to the component at address 2/4/0. After you do this, hardware management uses your user-defined name to track the component and to recognize the alternate paths 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
HWID:  DEVICEID HOSTNAME  TYPE      DEVICE  DRIVER NUM  DEVICE FIRST
-----
25:   3          ftwod    disk     none    0          1    dsk1    [2/4/0]

      WWID:0910003c:"DEC      (C) DECZG41400123ZG41800340:d02t00004100000"
      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 disk fails, you might 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 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.

Do not use this procedure alone if a failed disk is managed by an application such as AdvFS or LSM. Before you can swap managed disks, you must put the disk management application into an appropriate state or remove the

disk from the management application. See the appropriate documentation, such as the *Logical Storage Manager* and *AdvFS Administration* guides.

---

**Note**

---

The replacement disk must be of the same type for the `redirect` operation to work.

---

The following example shows how you use the `redirect` option:

```
# /sbin/hwmgrr show scsi
      SCSI          DEVICE DRIVER NUM  DEVICE  FIRST
HWID: DEVICE- HOST-  TYPE  SUB-  OWNER  PATH  FILE  VALID
      ID      NAME      TYPE  TYPE
-----
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 component 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 component as described in the hardware manual.
2. Use the following command to find the new device:

```
# /sbin/hwmgrr 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 component 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/hwmgrr redirect scsi -src 3 -dest 36
```

#### 5.4.4.12 Using `hwmgrr` to Replace a Cluster Member's Boot Disk

On a single system, the `hwmgrr` command provides a `redirect` option which you use as part of the procedure to replace a failing disk. When you replace the failed disk, you use the `redirect` option to direct I/O from the failed component to the replacement device. This option redirects device special file names, cluster `dev_t` values, local `dev_t` values, logical ID, and HWID.



Only unique device identifiers (*did*) are accepted by the `redirect` option. In a cluster, device identifiers are not guaranteed to be unique and the command might fail as shown in the following example:

```
# hwmgr redirect scsi -src source_did -dest target_did
# "Error (95) Cannot start operation."
```

For the `redirect` operation to succeed, both or neither of the hardware identifiers must exist on each member of the cluster. Use the following procedure to ensure that the `redirect` operation works:

1. Verify whether the source and destination component exist. Use the following command on each member of the cluster:

```
# hwmgr show scsi -did device_identifier
      SCSI          DEVICE DEVICE  DRIVER NUM  DEVICE FIRST
HWID: DEVICEID HOST  TYPE    SUBTYPE OWNER  PATH FILE   VALID PATH
32:  did         rymoc  disk   none    2      1    dsk1   [0/1/0]
```

2. Follow this step only if the source component exists on other cluster members but the destination component does not.

Configure the destination component on those cluster members as follows:

```
# hwmgr scan scsi
```

---

#### Note

---

The bus scan is an asynchronous operation. The system prompt returns immediately but that does not mean that the scan is complete. On systems with many devices, the scan can take several minutes to complete.

---

3. Follow this step only if the destination component exists on other members of the system but the source component does not.

Delete the destination component from those cluster members as follows:

```
# hwmgr delete scsi did
```

4. You can now use the `redirect` option to direct I/O to the replacement drive.

#### 5.4.4.13 Viewing the Persistence Database for the name Subsystem

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 that you can manipulate by using other `hwmgr` commands.

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. You can also determine this by using the `view hierarchy` command option.
- **NAME** – The component name and the instance number, such as `pci0`, for personal computer interconnect (PCI) bus 0. Each additional PCI bus has a different instance number.
- **HOSTNAME** – The host on which the command is 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 is a bus, controller, or device.
- **PERSIST AT** – The logical address of the device, which might map to a physical location in the hardware. For example, the SCSI controller `scsi0` persists at slot 0 of the bus `isp0`.

#### 5.4.4.14 Deleting and Removing a Device from the name Persistence Database

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

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

The following example shows typical output from the `show name` command option on a small system. You specify the variable `name`, which is the component name shown in the output from the `show name` command option described in Section 5.4.4.13:

```
# 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
```

There are two SCSI adapters shown in the preceding output. If `scsi0` is the target of a `remove` operation then `scsi1` does not become `scsi0`. The location of the adapter persists at `aha0 slot 0` and the name `scsi1` is saved across boots.

To remove `scsi0` and rename `scsi1` you use the following commands:

```
# hwmgr remove name -entry scsi0
# hwmgr edit name -entry 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 disks (including CD-ROM devices). See Section 5.4 for a discussion of system components.

The system uses device special files 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 employed by 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 that make devices 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 need to perform manual operations on device special files only when there are problems with the system or when you need to support a device that the system cannot handle automatically. The following sections describe 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.

The following considerations apply in this release:

- The name of a device special file for a SCSI device has the format `/dev/disk/disk13a` for SCSI disks and `/dev/ntape/tape0_d0` for SCSI tapes. The name of 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 might also see these referred to as old (legacy) and new (current) device names in some scripts and commands. 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 is 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 nonclustered systems. Applications and commands either support all device names or display an error message informing you of the supported device name formats.

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 Commands

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 and you should be familiar with this concept before you proceed.
- Reference pages and commands:
  - See `dsfmgr(8)` for information on managing device special files and replaces the `MAKEDEV` command. (See `MAKEDEV(8)`.)
  - See `disklabel(8)` for information on maintaining disk pack labels.

- See `diskconfig(8)` for instructions on invoking the Disk Configuration GUI, a disk management tool that provides additional features over `disklabel`. 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 Configuration icon is located in the Configuration folder. Online help describes how to use this interface.

## 5.5.2 Device Special File Directories

To contain the device special files, a `/devices` directory exists under the root directory (`/`). 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 `/dev/disk` directory contains files for all supported disks, and the `/dev/ntape` directory contains device special files for nonrewind tape devices. In this release, only the subdirectories for certain classes are created. The available classes are defined in Appendix B. For all operations you must specify paths by using the `/dev` directory and not the `/devices` directory.

---

### Note

---

Some device special file directories are CDSLs, which enable devices to be available cluster-wide when a system is part of a cluster. You should be familiar with the file system hierarchy described in Chapter 6, in particular the implementation of CDSLs.

---

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 might 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 for disk and tape device special file names that use the legacy conventions are `/dev/rz14f` for a SCSI disk and `/dev/rmt0a` for a tape device. 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 continue to work as expected. However, features available with the current device naming convention might 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`) are created during the installation. If you determine that legacy device special file naming is required, you must create the legacy device names by using the appropriate commands described in `dsfmgr(8)`. Some devices do 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 a 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 addressed device. This suffix differs 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. 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
<code>tape0</code>	Default density
<code>tape0c</code>	Default density with compression
<code>tape0_d0</code>	Density associated with entry 0 in <code>/etc/DDR.dbase</code>
<code>tape0_d1</code>	Density associated with entry 1 in <code>/etc/DDR.dbase</code>

Using 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)	<code>_d0</code>
m (medium)	<code>_d2</code>
h (high)	<code>_d1</code>
a (alternate)	<code>_d3</code>

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 the correct 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 that act on device special files, be aware that any command or utility supplied with the operating system operates on current and legacy file names in one of the following ways:

- The utility accepts both forms of device name.
- Only the current device names are supported by the utility. If you use legacy device names, you cannot use the command.
- Only the legacy device names are supported by the utility. If you use current device names, you cannot use the command.



No device can use both forms of device names simultaneously. Test your shell scripts for compliance with the device naming methods. Refer to the individual reference pages or the online help for a command.

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. There is no relationship between the instance numbers. A device associated with legacy device special file `/dev/rz10b` might be associated with `/dev/disk/dsk2b` under the current system.

Using these names as examples, you can translate device names that appear in your scripts. You can also use the `dsfmgr(8)` command to convert device names.

**Table 5–3: Sample Device Name Translations**

<b>Legacy Device Special File Name</b>	<b>New Device Special File Name</b>
<code>/dev/rmt0a</code>	<code>/dev/tape/tape0</code>
<code>/dev/rmt1h</code>	<code>/dev/tape/tape1_d1</code>
<code>/dev/nrmt0a</code>	<code>/dev/ntape/tape0_d0</code>
<code>/dev/nrmt3m</code>	<code>/dev/ntape/tape3_d2</code>
<code>/dev/rz0a</code>	<code>/dev/disk/dsk0a</code>
<code>/dev/rz10g</code>	<code>/dev/disk/dsk10g</code>
<code>/dev/rrz0a</code>	<code>/dev/rdisk/dsk0a</code>
<code>/dev/rrz10b</code>	<code>/dev/rdisk/dsk10b</code>

### 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 you updated the operating system from a previous version by using the update installation procedure, both the current device special files and the legacy device files might exist. However, if you subsequently add new SCSI devices the `dsfmgr` command creates only the new device special files by default. When the system is rebooted, the `dsfmgr` command 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 is found and recognized automatically, added to the hardware management

databases, and its device special files created. On the first reboot after installation of the new device, the `dsfmgr` command is called automatically during the boot sequence to create the new device special files for that device.

To support applications that work only with legacy device names, you might need to manually create the legacy device special files, either for every existing device, or for recently-added devices only. Some recent devices that support features such as Fibre Channel can use only the current special device file naming convention.

The following sections describe some typical uses of the `dsfmgr` command. See `dsfmgr(8)` 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 `/sbin/dn_setup` script runs automatically at system start up to create device special file names. Normally, you do not need to use the `dn_setup` command. It is 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 might see device-related error messages at the system console during system start up. These messages might indicate that the system is 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 `dn_setup` script has the following functions. Generally, the `-sanity_check` option alone is useful to administrators. Use the remaining options under the guidance of technical support for debugging and problem solving. The options are as follows:

`-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 the required device special directories only.

-clean

Deletes everything in the device special directory tree and recreates the entire tree (including device special files).

-default\_config

Creates the class and category databases only.

-init

Removes all the default device special databases, files, and directories and recreates 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 by using the `dsfmgr` command. This information enables you to find out what devices are on a system, and obtain device identification attributes that you can use with other `dsfmgr` command options. For example, you can find a class of devices that 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`.

Use the following command to view the entries in the databases:

```
# /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 you can use with other `dsfmgr` commands. (See `dsfmgr(8)` 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. Use this information 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. Use this information 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 might be corrupted or device special files might be accidentally removed from the system. You might 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 might be a problem with the device special files, you can check the databases by using the `dsfmgr -v` (verify) command option.

---

#### Caution

---

If you see error messages at system start up that indicate a device naming problem, use the `verify` command option to enable you to proceed with the boot. Check your system configuration before

and after verifying the databases. The verification procedure fixes most errors and enables you to proceed, however it does 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 is 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, ensure that you are 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 attempt to configure the floppy disk device to use the `mtools` commands, and you find that you cannot access the device, use the following `dsfmgr` command to help diagnose the problem:

```
# /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
```

In the preceding output, the ERROR changes to a WARNING, indicating that the device special files for the floppy disk are created automatically. If you Repeat the `dsfmgr -v` command, no further errors are displayed.

#### 5.5.3.4 Deleting Device Special Files

If a device is permanently removed from the system, you can remove its device special file to reassign the file to another type of device. Use the `dsfmgr -D` command option to remove device special files, as shown in the following example:

```
# ls /dev/disk
cdrom0a  dsk0a    dsk0c    dsk0e    dsk0g    floppy0a
cdrom0c  dsk0b    dsk0d    dsk0f    dsk0h    floppy0c

# /sbin/dsfmgr -D /dev/disk/cdrom0*
-cdrom0a -cdrom0a -cdrom0c -cdrom0c
# ls /dev/disk
dsk0a    dsk0c    dsk0e    dsk0g    floppy0a
dsk0b    dsk0d    dsk0f    dsk0h    floppy0c
```

The output from the `ls` command shows that there are device special files for `cdrom0`. Running the `dsfmgr -D` command option 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 are located.

If device special files are deleted in error, and no hardware changes are made, recreate the files 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 might want to move (reassign) the device special files between devices by using the `dsfmgr -m` (move) command option. You can also exchange the device special files of one device for those of another device by using the `-e` option. 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 is recognized, mapped, and added to the device databases as described in Section 5.4. However, you might sometimes need to add devices that the system cannot detect and add to the system automatically. These devices might be old, or new prototypes, or they might not adhere closely to supported standards such as SCSI. In these cases, you must manually configure the device and its drivers in the kernel, by using the `ddr_config` command 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:

- The dynamic method uses the `ddr` command to rebuild the kernel and effect the disk configuration changes without shutting down the operating system.
- The static method requires that you use the `MAKEDEV` and `config` commands. You must also shut down the system and restart it to rebuild the kernel and effect the changes.

Use the `MAKEDEV` command or the `mknod` command to create device special files instead of using the `dsfmgr` command. The `kmknod` command creates device special files for third-party kernel layered products. See `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` command to manage the Dynamic Device Recognition (DDR) database for your system. These sections introduce DDR, then describe how you use the `ddr_config` command to:

- Add SCSI devices to the DDR database
- Convert a customized `cam_data.c` file

### 5.6.1.1 Understanding Dynamic Device Recognition

DDR 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 add new and changed SCSI devices to your system without rebooting 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.

---

You can use both methods 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 by 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. The system will



automatically recognize such devices and you can configure them by using the `hwmgrr` command.

### 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` command and does not produce an output database until all syntax errors are 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
```

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. (You specify the configuration file when you build your running kernel.) The procedure runs in multiuser mode and requires no input after it starts. Redirect output to a file 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 support both SVR4 and BSD ptys. To revert back to the original default behavior, create the BSD ptys by using the MAKEDEV command. See also SYSV\_PTY(8), pty(7), and MAKEDEV(8).

#### 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 might 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 is not 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 superuser (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 example shows how to add 400 pseudoterminals:

```
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, as follows:

```
./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. See `MAKEDEV(8)` for instructions on making a large number of pseudoterminals. (See 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/tty0` through `/dev/tty0f`. 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
```

See `securettys(4)` for more information.

#### 5.6.4.2 Adding Other Devices

When you add new SCSI devices to your system, the system automatically detects and configures them. It runs the appropriate `hwmgrr` and `dsfmgr` commands to register the devices, assign identifiers, and create device special files. However, you might need to manually create device names for other devices by using the `MAKEDEV` command. You might also need to recreate device special files that you 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. Before adding a device, read the owner's manual that came with your system processor and any documentation that came with the device itself. You might 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.

It is not necessary to use the `MAKEDEV` command if you simply want to create legacy `rz` or `tz` device special files in `/dev` such as `/dev/rz5`. The `dsfmgr` command 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 as follows:

```
# ./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. See 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. See Chapter 2 for information on booting your processor.

## 5.7 Using Device Commands and Utilities

The preceding sections described generic hardware management tools that you use to manage many aspects of all devices, such as the `hwmgr` command described in Section 5.4. The following sections describe hardware management tools that are targeted at a particular kind of device and perform a specific task. The topics covered in these sections are:

- Finding device utilities
- Using SCSI utilities
- Disk partitioning
- Copying disks
- Monitoring disks

### 5.7.1 Finding Device Utilities

Many of the device utilities are documented elsewhere in this guide or at other locations in the documentation set. For example, utilities that enable you to configure network devices are documented in detail in the *Network Administration: Services* guide. Table 5–4 provides references to utilities documented in the guides, including those listed in this chapter. Other utilities are documented 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	Advanced Configuration and Management	Section 5.7.2.1, scu
Disks	Partitioning	diskconfig, disklabel
	Copying	Section 5.7.5, dd
	Monitoring Usage	Section 5.7.6, 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

**Table 5–4: Device Utilities Documented in the Guides (cont.)**

Device	Task	Location
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 and 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).
	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)



**Table 5–5: Device Utilities Documented in the Reference Pages (cont.)**

<b>Device</b>	<b>Task</b>	<b>Location</b>
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 Configuration Utility, scu

The SCSI/CAM Utility Program, `scu`, provides commands for advanced maintenance and diagnostics of SCSI peripheral devices and the CAM I/O subsystem. For most daily operations, you use the `hwmgr` command. The `scu` program has an extensive help feature that describes its options and conventions. See `scu(8)` 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 `radisk(8)` 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
```

### 5.7.2.2 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 `diskconfig(8)` for information on invoking the Disk Configuration GUI (`diskconfig`). An online help volume describes how you use the graphical interface. See `disklabel(8)` for information on command options.

The Disk Configuration GUI provides a graphical interface to several disk maintenance tasks that you can perform manually, by using the following commands:

- `disklabel` – Use this command to install, examine, or modify the label on a disk drive or pack. The disk label contains information about the disk, such as type, physical parameters, and partitioning. See also the `/etc/disktab` file, described in `disklabel(4)`.
- `newfs` – This command creates a new UFS file system on the specified device. You cannot use the `newfs` command to create Advanced File System (AdvFS) domains. Instead, use the `mkfdmn` command, as described in `mkfdmn(8)`.
- `mkfdmn` and `mkfset` – Use these commands to create Advanced File System (AdvFS) domains and filesets.

An example of using manual methods is provided in Section 5.7.4.

Invoke the Disk Configuration interface 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

---

Disk Configuration displays appropriate warnings when you attempt to change partition sizes. However, you should plan the changes in advance to ensure that you do not overwrite any required data. Back up any data partitions before attempting this task.

---

A window titled Disk Configuration on *hostname* is displayed. This is the main window for 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 are 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, which displays disk information such as the device name, the total size of the disk, and usage information. This box enables you to assign or edit the disk label, and create an alias name for the device.
- The Selected Partition box, which displays dynamic sizes for the selected partition. These sizes are updated as you change the

partitions by using the bar-chart. You can 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 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:

1. Invoke the SysMan Menu, as described in Chapter 1
2. Expand the Storage options, and select Basic File System Utilities - Mount File Systems
3. In the Mount Operation window, select the option to mount a specific file system and press Next
4. In the Name and Mount Point window:
  - a. Type a mount point, such as `/usr/newusers`
  - b. 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 a single partition on the entire disk, use partition `c`.
- Unless it is mounted, you must specify the raw device for partition `a`, which begins at the start of the disk (sector 0), when you change the label. If partition `a` is mounted, you must then use partition `c` to change the label. Partition `c` must also begin at sector 0.

---

**Caution**

---

If partition `a` is mounted and you attempt to edit the disk label using device partition `a`, you cannot change the label. Furthermore, you do not receive any error messages indicating that the label is not written.

---

Before changing the size of a disk partition, review the current partition setup by viewing the disk label. The `disklabel` command allows you to view the partition sizes. The bottom, top, and size of the partitions are in 512-byte sectors.

To review the current disk partition setup, use the following `disklabel` command:

```
/sbin/disklabel -r device
```

Specify the device with its directory name (`/dev`) followed by the raw device name, drive number, and partition a or c. You can 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*)
```

Take care when you change partitions because you can overwrite data on the file systems or make the system inefficient. If the partition label becomes corrupted while you are changing the partition sizes, you can return to the default partition label by using the `disklabel` command with the `-w` option, as follows:

```
# disklabel -r -w /dev/rdisk/dsk1a rz26
```

The `disklabel` command allows you to change the partition label of an individual disk without rebuilding the kernel and rebooting the system. Use the following procedure:

1. Display disk space information about the file systems by using the `df` command.
2. View the `/etc/fstab` file to determine if any file systems are designated as swap space.

3. Examine the disk's label by using the `disklabel` command with the `-r` option. (See `rz(7)`, `ra(7)`, and `disktab(4)` for information on the default disk partitions.)
4. Back up the file systems.
5. Unmount the file systems on the disk whose label you want to change.
6. Calculate the new partition parameters. You can increase or decrease the size of a partition. 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:

```
# /sbin/disklabel -e disk
```

An editor, either the `vi` editor or that specified by the `EDITOR` environment variable, is invoked so you can edit the disk label, which is in the format displayed with the `disklabel -r` command.

The `-r` option writes the label directly to the disk and updates the system's in-memory copy, if possible. The `disk` parameter specifies the unmounted disk (for example, `dsk0` or `/dev/rdisk/dsk0a`).

After you quit the editor and save the changes, the following prompt is displayed:

```
write new label? [?]:
```

Enter `y` to write the new label or `n` to discard the changes.

8. Use the `disklabel` command with the `-r` option to view the new disk label.

#### 5.7.4.1 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 enables you to determine when a partition or an overlapping partition is in use.

If the partition is not in use, the command continues to execute. In addition to mounting or creating file systems, commands such as `mount`, `newfs`, `fsck`, `voldisk`, `mkfdmn`, `rmfdmn`, and `swapon` also modify the disk label, so that the `fstype` field specifies partition usage. For example, when you add a disk partition to an AdvFS domain, the `fstype` field is set to `AdvFS`.

If the partition is not available, these commands return an error message and ask if you want to continue, as shown in the following example:

```
# newfs /dev/disk/dsk8c
WARNING: disklabel reports that basename,partition currently
```



```
is being used as "4.2BSD" data. Do you want to
continue with the operation and possibly destroy
existing data? (y/n) [n]
```

Applications, as well as operating system commands, can modify the `fstype` of the disk label, to indicate that a partition is in use. See `check_usage(3)` and `set_usage(3)` for more information.

## 5.7.5 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 is not meant for copying multiple files, copy a disk or a partition only to a disk that you are using as a data disk, or to a disk that does not contain a file system. Use the `dump` and `restore` commands, as described in Chapter 9, to copy disks or partitions that contain a UFS file system. Use the `vdump` and `vrestore` commands, as described in *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. The target disk must be the same size as or larger than the original disk.

To determine if the target disk has a label, use the following `disklabel` command:

```
# /sbin/disklabel -r target_disk
```

You must specify the target device directory name (`/dev`) followed by the raw device name, drive number, and partition `c`. If the disk does not contain a label, the following message is displayed:

```
Bad pack magic number (label is damaged, or pack is unlabeled)
```

The following example shows a disk that already contains a label:

```
# disklabel -r /dev/rdisk/dsk1c
type: SCSI
disk: rz26
```

```

label:
flags:
bytes/sector: 512
sectors/track: 57
tracks/cylinder: 14
sectors/cylinder: 798
cylinders: 2570
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0          # milliseconds
track-to-track seek: 0 # milliseconds
drivedata: 0

8 partitions:
#   size  offset  fstype [fsize bsize  cpg]
a: 131072     0  unused 1024 8192 # (Cyl.   0 - 164*)
b: 262144 131072  unused 1024 8192 # (Cyl. 164*- 492*)
c: 2050860     0  unused 1024 8192 # (Cyl.   0 - 2569)
d: 552548 393216  unused 1024 8192 # (Cyl. 492*- 1185*)
e: 552548 945764  unused 1024 8192 # (Cyl. 1185*- 1877*)
f: 552548 1498312 unused 1024 8192 # (Cyl. 1877*- 2569*)
g: 819200 393216  unused 1024 8192 # (Cyl. 492*- 1519*)
h: 838444 1212416 unused 1024 8192 # (Cyl. 1519*- 2569*)

```

If the target disk already contains a label and you do not want to keep the label, you must clear the label by using the `disklabel -z` command. For example:

```
# disklabel -z /dev/rdisk/dsk1c
```

To copy the original disk to the target disk and keep the target disk label, use the `dd` command, specifying the device directory name (`/dev`) followed by the raw device name, drive number, and the original and target disk partitions. For example:

```
# dd if=/dev/rdisk/dsk0c of=/dev/rdisk/dsk1c \
skip=16 seek=16 bs=512k
```

## 5.7.6 Monitoring Disk Use

To ensure an adequate amount of free disk space, 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.

### 5.7.6.1 Checking Available Free Space

To ensure sufficient space for your configured file systems, use the `df` command regularly 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.

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 in use, the amount available (free), the percentage in use, 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.

See `df(1)` 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.

---

### 5.7.6.2 Checking Disk Use

If you determine that a file system has insufficient space available, you might want to find out who is using the space. You can do this with the `du` command or the `quot` command.

The `du` command returns 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 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 in use by the system

If you do not specify any options, an entry is generated for each directory. See `du(1)` for more information on command options.

The following example displays a summary of blocks that all main subdirectories in the `/usr/users` directory use:

```
# /usr/bin/du -s /usr/users/*
440    /usr/users/barnam
43     /usr/users/broland
747    /usr/users/frome
6804   /usr/users/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 file sizes only.

---

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 following example displays the number of blocks that each user owns and the number of files owned by each user in the `/dev/disk/dsk0h` file system:

```
# /usr/sbin/quot -f /dev/disk/dsk0h
```

---

**Note**

---

You must specify the character device special file to return the information, because when the device is mounted the block special device file is busy.

---

See `quot(8)` 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 might 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 and extend 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 ffm.

You might 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 might 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/usr/projects`, you add it as a new branch on the hierarchy, under the existing `/usr/usr` 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 might be the whole disk (partition `c`) or parts of the disk, such as partitions `a` through `h`. Depending on the size of the disk, the partitions vary in size, and are usually expressed in megabytes (MB). When you initially create a file system, you create it on a disk partition and thus assign a finite amount of size (disk space) to that file system. Increasing the size of a UFS file system might 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 might 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(4)` reference pages.
- Device Special Files – Refer to Chapter 5.
- Sockets – Refer to the `socket(2)` reference page, the *Network Administration: Connections* 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 might 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 an 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 incompatibility 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 `fsck` does 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` option, `-B`, is added. This option enables you to convert Version 4.0 file systems back to Version 3.0. When you use this option, `fsck` make the conversion only if no directory in the file system has more than 32k subdirectories and no file has more than 32k hard links.

The following important restrictions apply when using both Version 3.0 and Version 4.0 of UFS on systems that are running previous versions of the operating system (such as V4.0F):

- Do not run previous versions of `fsck` using the `-p` or `-y` options on a Version 4.0 file system unless you are certain that there are no directories that have more than 32k subdirectories. If you attempt to do this, any directories that have more than 32k subdirectories will be permanently 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 enter `y` in error, 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 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 are subject to a limit on the

number inodes, which in turn controls the maximum number of files that a file system can contain. The maximum number of inodes depends on the size of the file system.

The first inode (inode 1) on each file system is unnamed and unused. The second inode (inode 2) must correspond to the root directory for the file system. All other files in the file system are under the file system's root directory. After inode 2, you can assign any inode to any file. You can also assign any data block to any file. The inodes and blocks are not allocated in any particular order.

If an inode is assigned to a file, the inode can contain the following information:

- File type

The possible types are regular, device, named pipes, socket, and symbolic link files.

- File owner

The inode contains the user and group identification numbers that are associated with the owner of the file.

- Protection information

Protection information specifies read, write, and execute access for the file owner, members of the group associated with the file, and others. The protection information also includes other mode information specified by the `chmod` command.

- Link count

A directory entry (link) consists of a name and the inumber (inode number) that represents the file. The link count indicates the number of directory entries that refer to the file. A file is deleted if the link count is zero; the file's inode is returned to the list of free inodes, and its associated data blocks are returned to the free-block list.

- Size of the file in bytes

- Last file access date

- Last file modification date

- Last inode modification date

- Pointers to data blocks

These pointers indicate the actual location of the data blocks on the physical disk.

- Data blocks

Data blocks contain user data or system files.

## 6.1.5 Directories and File Types

The operating system views files as bit streams, allowing you to define and handle on-disk data, named pipes, UNIX domain sockets, and terminals as files. This object-type transparency provides a simple mechanism for defining and working with a wide variety of storage and communication facilities. The operating system handles the various levels of abstraction as it organizes and manages its internal activities.

While you notice only the external interface, you should understand the various file types recognized by the system. The system supports the following file types:

- Regular files contain data in the form of a program, a text file, or source code, for example.
- Directories are a type of regular file and contain the names of files or other directories.
- Character and block device special files identify physical and pseudodevices on the system.
- UNIX domain socket files provide a connection between network processes. The `socket` system call creates socket files.
- Named pipes are device files. Processes use named pipes to communicate with each other.
- Linked files point to target files or directories. A linked file contains the name of the target file. A

symbolically linked file and its target file can be located on the same file system or on different file systems. A file with a hard link and its target file must be located on the same file system.

## 6.1.6 Device Special Files

Device special files represent physical devices, pseudodevices, and named pipes. The `/dev` directory contains device special files. Device special files serve as the link between the system and the device drivers. Each device special file corresponds to a physical device (for example, a disk, tape, printer, or terminal) or a pseudodevice (for example, a network interface, a named pipe, or a UNIX domain socket). The driver handles all read and write operations and follows the required protocols for the device.

There are three types of device special files:

- Block device special files

Block device special files are used for devices whose driver handles I/O in large blocks and where the kernel handles I/O buffering. Physical



devices such as disks are defined as block device files. An example of the block device special files in the `/dev` directory follows:

```
brw----- 1 root system 8, 1 Jan 19 11:20 /dev/disk/dsk0a
brw----- 1 root system 8, 1 Jan 19 10:09 /dev/disk/dsk0b
```

- Character device special files

Character device special files are used for devices whose drivers handle their own I/O buffering. Disk, terminal, pseudoterminal, and tape drivers 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 that are not members of a cluster. You should be aware of these requirements as follows:

- The root (`/`), `/var`, and `/usr` file systems each have a `/cluster` subdirectory that is not used on a single system, but must not be deleted or the system cannot be added into a cluster at some future time.
- When systems are joined into clusters, they are designated as members of the cluster. There is a unique pathname to any file, including an identifier that is unique to the member system (member-specific). These pathnames are called context-dependent symbolic links (CDSLs). As the name implies, CDSLs are symbolic links with a variable element in the pathname. The variable element is different for each cluster member and provides the context when it is resolved by an application or command.

- Some important system files reside in target directories that have unique CDSLs pointing to the target location. This design ensures that shared (cluster-wide) files are kept separate from unshared (member-specific) files.
- Update installations might 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 might 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 might 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 `cdslinvcchk(8)` reference page contains a discussion of the `/usr/sbin/cdslinvcchk` 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 might 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 might need to re-create it.

### 6.2.2.1 Structure of a CDSL

CDSLs are simply the symbolic links described in 1n(1). The links contain a variable that identifies each system that is a cluster member. This variable is resolved at run time into a target. A CDSL is structured as follows:

```
/etc/rc.config -> /cluster/members/{memb}/etc/rc.config
```

Before support for clusters was introduced, the pathname for this file was `/etc/rc.config`. This file is now linked through a CDSL to a member-specific target, and the structure of the link can be interpreted as follows:

- The `/cluster` directory resides in the root directory and contains paths to the files that are either shared or (as in this example) member-specific.
- The `/cluster/members/` directory contains a directory for the local member identifier, `member0`, and a link to the variable path element `{memb}`. The directory `/cluster/members/member0` contains member-specific system directories such as `devices` and `etc`.
- The `{memb}` variable path element is used to identify individual members of a cluster. At run time, this variable is resolved to be `member`, appended with the value of the `sysconfigtab` variable `generic:memberid`. The default value for this variable is zero, and the value is unique for each member of a cluster.

The file `/.local..` in root is a link to `cluster/members/{memb}` and defines the system-specific files. Any system-specific file can be referenced or created through the `/.local..` path. A file created as `/.local../etc/[filename]` is not accessible through the path `/etc/[filename]` because `/etc` is a shared directory. The file is only accessible through `/.local../etc/[filename]` and `/cluster/members/{memb}/etc/[filename]`.

When a single system is not clustered with other systems the variable `generic:memberid` is 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/cdslinvchk` to check the CDSL inventory on a single system. Periodically, revise the inventory and check the CDSLs against it. See `cdslinvchk(8)` for information on `cdslinvchk`.

### 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 might 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 might be faced with administering a system that could be in an unknown state; that is,

devices might 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 might be available on the system, but without any device special files. Use the `hwmgrr` command to examine all devices that are physically known to the system and visible on a bus. For example:

```
# hwmgrr -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:  /dev/disk/dsk0c     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 might be in use as follows:

```
# disklabel -r /dev/rdisk/dsk0a

8 partitions:
#      size  offset  fstype  [fsize bsize cpgh] #NOTE: values not
      exact
a:   262144      0  4.2BSD 1024 8192  16  # (Cyl.  0 -78*)
b:  1048576 262144  swap                # (Cyl.  78*-390*)
c:  17773524      0  unused      0    0    # (Cyl.  0 -5289*)
d:  1048576 1310720  swap                # (Cyl. 390*-702*)
e:  9664482 2359296  AdvFS                # (Cyl. 702*-3578*)
f:  5749746 12023778  unused      0    0    # (Cyl.3578*-5289*)
g:  1433600  524288  unused      0    0    # (Cyl. 156*-582*)
h:  15815636 1957888  unused      0    0    # (Cyl. 582*-5289*)
```

2. From the `disklabel` command output, it appears that there are several unused partitions. However, you cannot use the `c` partition because it overlaps with the other partitions. Unless a custom `disklabel` has been created on the disk, only three possible tables of standard partitions are available for use, as shown in Table 6-1.



**Table 6–1: Disk Partition Tables**

Partition Table	Description
c	The entire disk is labeled as a single partition. Therefore, other partitions overlap c and you cannot use it.
a b g h	The disk is divided into four partitions. Partition a can be used as a boot partition. Partitions c, d, e, and f overlap and you cannot use them.
a b d e f	The disk is divided into five partitions. Partition a can be used as a boot partition. Partitions c, g, and h overlap you cannot use them.

The disk listed in the output from the `disklabel` command in step 1.c already uses partitions a, b, d, and e. Therefore it is labeled for five partitions, and the f partition is unused and available for use by the new file system.

**Note**

If a custom disk label has been applied to the disk and partitions are extended, you might 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:

<code>cdfs</code>	Specifies an ISO 9600 or HS formatted (CD-ROM) file system.
<code>nfs</code>	Specifies NFS.
<code>procfs</code>	Specifies a <code>/proc</code> file system, which is used for debugging.
<code>ufs</code>	Specifies a UFS file system or a swap partition.
<code>advfs</code>	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 <code>quotacheck</code> command and that file system quotas are enabled with the <code>quotaon</code> command.  By default, user and group quotas for a file system are contained in the <code>quota.user</code> and <code>quota.group</code> files, which are located in the directory specified by the mount point. For example, the quotas for the file system on which <code>/usr</code> is mounted are located in the <code>/usr</code> directory. You also can specify another file name and location. For example: <code>userquota=/var/quotas/tmp.user</code>
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 do the following:

- Check a file system by using the `fsck` command.

- Change partitions by using the `disklabel` command. (Take care with this operation. Changing partitions can destroy existing file systems on the disk.)

The `umount` command has the following syntax:

```
umount [- afv ] [- h host ] [- t type ] [ mount_point ]
```

If any user process (including a `cd` command) is in effect within the file system, you cannot unmount the file system. If the file system is in use when the command is invoked, the system returns the following error message and does not unmount the file system:

```
mount device busy
```

You cannot unmount the root file system with the `umount` command.

### 6.3.6 Extending the UFS File System

You can increase the capacity of a UFS file system up to the storage available on a single disk or Logical Storage Manager (LSM) volume. The process of increasing the capacity (or size) of an UFS file system is called extending the file system.

When the file system is on line (mounted) you can extend it by using `mount` command options. If preferred, you can perform this operation when the file system is off line (dismounted) by using the `extendfs` command. You can use this procedure as either a temporary or permanent solution if the system notifies you that a file system is full. File systems can be extended as frequently as required, up to the physical limits of the storage device.

You cannot reverse this procedure. The only way you can return a file system to its original volume is to back up the file system using the `dump` or a back up utility, and then restore the file system to an appropriately sized disk partition.

The prerequisites for extending a file system are as follows:

Identify file systems

Use the `more` command on the `/etc/fstab` file to identify file systems and the partitions on which they reside as follows:

```
# /usr/bin/more /etc/fstab
/dev/disk/dsk0a on / type ufs (rw)
/proc on /proc type procfs (rw)
.
.
.
/dev/disk/dsk15e on /databases type ufs (rw)
/dev/disk/dsk4g on /projects type ufs (rw)
```

## Verify the file system back up status

This procedure is nondestructive, and designed so that you can perform it quickly when needed. However, you might want to back up important data files.

## Determine the available disk storage capacity

A UFS file system exists on a single disk partition or LSM volume. To extend the file system, increase the disk space as follows:

- If LSM is not in use, increase the size of the disk partition by decreasing the size of an unused adjacent partition.
- If LSM is in use, use LSM commands to extend the volume. Refer to the *Logical Storage Manager* guide for information. The examples in this section refer to UFS file systems where LSM is not in use.

For example, you have a disk that is currently used as follows:

- The `a` partition is in use as a 500 MB tertiary swap partition.
- The `b` partition is in use as a 2 GB UFS file system dedicated to user files.
- The `g` and `h` partitions are unused and total 6 GB in additional disk storage capacity.

In the preceding example, you can extend the UFS file system on the `b` partition by an additional 6 GB. You need not take the entire 6 GB in a single extension; you can stage the extension to conserve disk space.

If you do not have adequate disk capacity to extend the file system, back it up and restore it to as new disk volume as described in Chapter 9.

## Reset the partition size.

Use the `disklabel` command to reset the size of the partition on which the file system resides. The following procedure describes the use of the `disklabel` command.

---

### Note

---

You cannot use the graphical disk configuration utility, `diskconfig`, to perform this operation. This is because the `diskconfig` utility will not bypass the partition checking and will disallow reconfiguration of any partitions that are in use.

---

1. Save the current disk label to a file so that you can edit the partitions. For example:

```
# /sbin/disklabel -r /dev/disk/dsk4 > d4label
```

2. Edit the saved label to increase the capacity of the partition in use, decreasing the unused partition by an identical amount.

```
:  
b: 10192000 1048576 4.2BSD 1024 8192 16 # (Cyl. 312*- 2750*)  
g: 7104147 8669377 unused 1024 8192 # (Cyl. 2580*- 5289*)
```

For example, to increase the size of partition `g` by 3,000,000 blocks, change the label as follows:

```
:  
b: 13192000 1048576 4.2BSD 1024 8192 16 # (Cyl. 312*- 2750*)  
g: 4104147 8669377 unused 1024 8192 # (Cyl. 2580*- 5289*)
```

Save the `disklabel` file and exit from the editor.

3. Write the label to the raw disk, specifying the edited file as follows:

```
# /sbin/disklabel -R /dev/rdisk/dsk4 d4label
```

Refer to the `disklabel(8)` reference page for more information.

After you create additional disk space, extend the file systems by using one of the methods described in the following sections.

### 6.3.6.1 Extending a Dismounted File System

Use the `extendfs` command to extend a file system that is off line and in use. You can extend the entire partition on a single operation, or extend the file system in stages. Use the `extendfs` command in either of the two following ways. These procedures assume that you completed the prerequisite step to increase the disk partition size, as described in Section 6.3.6.

- To extend the file system to the entire partition, use a command similar to the following:

```
# /sbin/extendfs dsk4  
Warning: 1324 sector(s) in last cylinder unallocated  
/dev/rdisk/dsk4h: 9057236 sectors in 2696 cylinders of 20  
tracks, 168 sectors  
4422.5MB in 169 cyl groups (16 c/g, 26.25MB/g, 6336 i/g)  
super-block backups (for fsck -b #) at:  
32, 53968, 107904, 161840, 215776, 269712, 323648, 377584,  
:  
:
```



Note that the output from this command is similar to the output from the `/sbin/newfs` command, used for creating new UFS file systems. Refer to the `newfs(8)` reference page for more information.

- To extend the file system to use only part of the available partition space, use the `-s` option as follows:

```
# /sbin/extendfs -s 500000
```

This example takes only 500,000 blocks of the available partition space, saving the remainder for future extensions. As shown in the preceding example, the output from the `/sbin/extendfs` command is similar to the output from the `/sbin/newfs` command.

### 6.3.6.2 Extending a Mounted File System

Use the `mount` command to extend a file system that is on line (mounted) and in use. This procedure assumes that you completed the prerequisite steps to identify the mounted partition and increase the disk partition size, as described in Section 6.3.6.

The syntax of the `mount` command is described in Section 6.3.4. To extend a file system, you use the `extend` option with the `-o` option and specify the mount point as follows:

```
# /sbin/mount -u -o extend /projects
extending file system, please wait.
```

The `mount` command does not display any completion or error output for this operation. The length of time for completion depends on the size of the partition, and might take several minutes to complete. You should therefore verify the operation using the `df` command as follows:

```
# /usr/bin/df /projects
```

The output from this command confirms if the operation was a success. If the operation was not a success, verify that the disk partition is correct and the mount point exists.

## 6.4 Administering UFS File Systems Using SysMan

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 you can use 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 might 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.  
To invoke this interface from the command line, use the `dxlsm` command. Refer to the `dxlsm(8X)` 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\_Fileystems 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\_Fileystems 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` or `autofs` utilities, which automatically mount exported networked file systems when a local user accesses (imports) them. Refer to the *Network Administration: Services* guide for information on NFS, `automount` and `autofs`. An NFS mount typically lists the exporting host system name, followed by the exported directory as follows:

```
19serv:/share/19serv/tools/tools    /tmp_mnt/19serv/tools
```

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 might 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 might 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 might not yet be visible to the system and you might 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 might 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 might 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. See *Network Administration: Services* and `nfs(4)` 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 `newfs(8)` for the memory file system.
- Advanced Mount options – Type any advanced mount options that you want for the file system. For example, the `dirty` option, which allows a file system to be mounted even if it was not dismounted cleanly, such as after a system crash. Refer to the `mount(8)` reference page for more information on the various options.

After 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 might have further options to share file systems with PC clients. Refer to the ASU *Concepts and Planning Guide*.

You might 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: Services* 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 you can launch from the command line or from the CDE Application Manager – DailyAdmin folder. See the File Sharing option in that folder. Online help is available for this interface. 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: Services* 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 might 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: Services* 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.

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 you can launch from the command line, or from the CDE Application Manager – DailyAdmin folder. See the File Sharing option in that folder. Online help is available for this interface. 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 as follows:
    - Read/Write – Allows users to both read data from and write data to the shared file system. Note that this might 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 as follows:
    - 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.
2. 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 are 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 might 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 might 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 might not always be possible for an unmount operation to complete. For example a user might 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 HB00931B93.
3. Advanced newfs options – Enter any option flags, such as `-b 64` for a 64 kilobyte block size.

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 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 might 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 might 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 might 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(1)`, and `repquota(8)` for more information on file system quotas.

## 6.6 Backing Up and Restoring File Systems

The principal backup and restore utilities for both AdvFS and UFS are the `vdump` and the `vrestore` utilities. These utilities are used for local operations on both AdvFS and UFS file systems. The utilities are described in `vdump(8)` and `vrestore(8)`. For remote backup and restore operations on both AdvFS and UFS file systems, the utilities are `rvdump` and `rvrestore`.

For administrators who want to back up only UFS, the traditional utilities are described in `dump(8)` and `restore(8)`.

Examples of backup and restore operations for AdvFS are described in *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 superbblock.

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

Use the following tools to help you resolve problems associated with UFS file systems:

- Using the UNIX Shell Option

The UNIX Shell Option is an installation option for experienced administrators and is available during either a textual or graphical installation of the operating system. For example, you might 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. Use the shell option for both AdvFS and UFS file system problems.

- Using the `/usr/field` directory and the `fsx` command

The `/usr/field` directory contains programs related to the field maintenance of the operating system. You can use the programs in this directory to monitor and exercise components of the operating system and system hardware.

The `fsx` utility exercises file systems. 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.

- Use the `dumpfs` utility to display information on UFS file systems. Refer to the `dumpfs(8)` reference page.
- Use the event manager, EVM (the Event Manager) to filter and display events that are related to file system problems. This utility is useful for setting up preventative maintenance and monitoring of file systems and storage devices. Refer to Chapter 13 for information.
- Use the SysMan Station and Insight Manager to provide graphical views of file systems and to monitor and troubleshoot file system problems, such as lack of disk space. Refer to Chapter 1 for information.



---

## Administering User Accounts and Groups

Assigning user accounts and organizing user accounts into related groups is the most common way that you provide system resources to users. This chapter describes user account and group administration, organized into the following sections:

- Section 7.1 describes the utilities that you can use to administer accounts and groups, and the user environments in which you can use these utilities. This section also provides pointers to related documentation and utilities.
- Section 7.2 is a quick start section, providing brief information on the utilities. You can use the online 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 system data files for passwords and groups and how to set 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 your local system configuration, the user environment, and your personal preferences, there are several methods and a number of different utilities that you can use to administer user accounts. The following sections introduce and describe these options and identify any restrictions or requirements for their use.

#### 7.1.1 Administrative Utilities

The operating system provides several different utilities that you can use to administer accounts. Not all are described in detail in this chapter.

However, the principles of use are the same for all utilities. Refer to the online help and reference pages for each utility for specific information on the options available.

The utilities are listed in Table 7–1. You must be root user on UNIX, or the Windows NT domain administrator to use these utilities.

**Table 7–1: Utilities for Administering Accounts and Groups**

Utility	User Environment Description
SysMan Menu, Accounts options. Manage local users and groups. Manage NIS users and groups.	<p>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 choose the creation or deletion of associated (synchronized) Windows NT domain accounts but does this automatically, depending on how the account defaults are configured (with <code>useradd</code> or <code>usermod</code>).</p> <p>The filter (search) features provided by SysMan Menu Accounts options make it the preferred method of managing a high volume of user accounts.</p>
Account Manager ( <code>dxaccounts</code> )	<p>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.</p>
<code>useradd</code> , <code>usermod</code> , <code>userdel</code>	<p>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.</p>
<code>groupadd</code> , <code>groupmod</code> , <code>groupdel</code>	<p>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.</p>

**Table 7–1: Utilities for Administering Accounts and Groups (cont.)**

Utility	User Environment Description
Advanced Server for UNIX (ASU). 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.
ASU 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 ASU User Manager for Domains utility.

You must install and configure the Advanced Server for UNIX (ASU) software to use the Microsoft Windows-based utilities. Using the ASU utilities is not explained in detail in this chapter, but is discussed only in the context of a UNIX server running the ASU software. Refer to the *ASU Installation and Administration Guide* for more information on installing and using ASU.

### 7.1.2 Notes and Restrictions on Using the Utilities

The following restrictions apply when using account management utilities, or when certain system features are enabled:

- **Configuring default UNIX account characteristics (properties)**  
To configure the default UNIX account and group characteristics, you can use only the UNIX command utilities or Account Manager `dxaccounts`. Refer to the *ASU Installation and Administration Guide* for more information on setting default values for PC accounts when ASU is in use.
- **Enhanced (C2) security**  
When enhanced security is enabled, it places restrictions on account creation and enables additional features such as:
  1. Enhanced password controls
  2. Options for enabling and disabling (or locking) accounts
  3. Options for deleting and retiring accountsRefer to the *Security* guide for more information.
- **Network Information Services (NIS)**  
NIS enables users to log in to any system on the local network that is running NIS. User data, such as account name and password is shared

between all NIS systems, and users 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 are enabled only when NIS is running. Refer to the *Network Administration: Services* guide for information on setting up the NIS environment.

- Multiple instances of account management utilities

When invoked, any account management utility creates a lock file, preventing other account management utilities (or two instances of the same utility) from accessing system files such as `/etc/passwd`. This lock file is located at `/etc/.AM_is_running`. Creation of the lock file prevents possible corruption of account data in the system files. Under certain circumstances, this lock file might not clear correctly and you must delete it manually. Before you remove a lock file, ensure that it does not relate to a legitimate instance of an account management tool.

The SysMan Menu Accounts options are designed to use deferred completion. This means that any data that you enter is stored and not written to a file until you confirm it. Therefore, while you can invoke a SysMan Menu Accounts option while another instance of an account management utility is running, you cannot click on Apply or OK to update the system file. When the other instance of an account management utility is closed, the lock file is removed and you can complete the transaction.

The Division of Privileges (DOP) and distributed administration features enables the root user to easily assign account management privileges to other users. However, only one account management utility can be used by one authorized user at any time.

### 7.1.3 Related Documentation

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: Services* 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 ASU documentation kit provides the *Concepts and Planning Guide, Installation and Administration Guide, and Release Notes*.
- Reference pages provide a definitive list of all options and switches supported by commands. The following pages are referenced in this chapter:
  - The command-line utilities are documented in `useradd(8)`, `usermod(8)`, `userdel(8)`, `groupadd(8)`, `groupmod(8)`, and `groupdel(8)`.
  - The SysMan utilities are documented in `sysman(8)` and `sysman_cli(8)`.
  - Invoking the Account Manager (`dxaccounts`) is documented in the `dxaccounts(8)` reference page.
  - The system files are documented in `passwd(4)`, `group(4)`, `shells(4)`, and `default(4)`.
  - Individual commands are documented in `passwd(1)`, `vipw(8)`, `grpck(8)`, and `pwck(8)`.
- 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.

#### 7.1.4 Related Utilities

The resources in the following list are also useful when administering accounts. These commands and utilities might be useful in correcting system problems when the graphical user environments are unavailable, such as after a system crash, or if you have access to only a character-cell terminal.

- 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` utility enables 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. You can also use the `vipw` utility 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 `cs`, `ksh`, and `sh` commands invoke and interpret the C, Korn, and POSIX shells.
- The `grpck` and `pwck` utilities enable you to check the integrity of the `group` and `passwd` files.
- The `quotaon` command enables you to turn quota information on and off.
- The `passwd`, `chfn`, and `chsh` commands provide the same functions as password options in command utilities, such as `usermod` and the `dxaccounts` Password option.

## 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 to configure the system defaults and use the advanced features of account and group administration utilities.

### 7.2.1 Creating Primary Accounts During System Setup

On the first root login after a full installation of the operating system, the System Setup utility is automatically displayed to guide you through the options for configuring your system. The Account Manager (`dxaccounts`) icon included in System Setup enables you to configure initial accounts. This icon invokes an X11-compliant graphical interface (GUI) that you can run under the Common Desktop Environment (CDE) or other X-windowing environments. See Section 7.5.2 for full information on using the Account Manager. When the Advanced Server for UNIX (ASU) is installed and configured, you can also use the Account Manager (`dxaccounts`) GUI to administer Windows NT domain accounts as described in Section 7.6.

### 7.2.2 Using the Account Manager (`dxaccounts`) GUI

The Account Manager (`dxaccounts`) provides features supported by the CDE environment, such as drag-and-drop and cut-and-paste, to quickly clone new accounts from existing accounts. You can invoke this GUI as follows:

- Use the following command from a terminal to invoke the GUI in any X11-compliant windowing environment:

```
# dxaccounts
```

- In CDE, open the Application Manager or the SysMan Applications pop-up menu from the Front Panel. Choose Daily Administration, and click on the Account Manager icon.

The `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 (`dxaccounts`) to configure default options for user accounts, such as the shell and the parent directory. See Section 7.4.2.6 for information.

### 7.2.3 Using the SysMan Menu Accounts Option

The SysMan Menu Accounts options provide the same functions as `dxaccounts`, but with limited support for the following features:

- Managing Windows NT domain accounts for PC clients
- Managing accounts under Enhanced (C2) security

Invoke the SysMan Menu Accounts options from the CDE Applications Manager, the CDE Front Panel (SysMan Applications menu), or from the command line as follows:

```
# sysman accounts
```

The Accounts options also let you add and modify accounts in NIS (Network Information Service) environments. You can add local users to any system without adding them to the NIS environment. Refer to the *Network Administration: Services* 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. Choose the Accounts option to expand the menu. The following menu options are displayed:
  - Manage local users
  - Manage local groups
  - Manage NIS users
  - Manage NIS groups
2. Move the pointer (or use the Tab key) to choose an option. Click on mouse button 1 (MB1) or the Enter key to invoke the utility.
3. The first window (or screen) of the utility is displayed, presenting you with the following options:

#### Add...

Use this option to create a new user account.

#### Modify...

Use this option to modify account details for an existing user account.

#### Delete...

Use this option to remove a user's account, and optionally to delete all their system resources.

#### Filter...

Use this option to filter (search) for a specific user or set of users. You can specify different search criteria such as the user's UID or account comment.

#### Options...

Use this option to define the number of accounts at which filtering starts automatically. You can also choose which user data is included in listings of user accounts.

Detailed use of these utilities is described in Section 7.4.1, and in the online help.

## 7.2.4 Using the Command-Line Utilities

The following command-line utilities are available for administering accounts and groups:

- `useradd`, `usermod`, and `userdel` – Use these commands to add, modify, and delete user accounts.
- `groupadd`, `groupmod`, and `groupdel` – 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 and Administration Guide* provided in the software kit.

## 7.3 Understanding User Accounts and Groups

The administration of user accounts and groups involves managing the contents of the system's password and group files. On standalone systems, the files you manage are `/etc/passwd`, which is documented in `passwd(1)`, and `/etc/group`, which is documented in `group(4)`.

On networked systems, typically, the Network Information Service (NIS) 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: Services* guide for more information.

If enhanced (C2) security is enabled on your system, you need to administer more than the `/etc/passwd` file for security. For example, the protected password database is used for security related information such as minimum password lengths and password expiration times. These tasks are documented in the *Security* manual.

### 7.3.1 System Files

The following system files may be updated when you perform account administration tasks and should be backed up regularly:

- The `/etc/group` file contains group data. Each row specifies one of the following: the group name; optional encrypted password; numerical group ID; and a list of all users who are members of the secondary group. For example:

```
system:*:0:root luis
daemon:*:1:daemon
uucp:*:2:uucp
mem:*:3:
kmem:*:3:root
```

```

bin:*:4:bin,adm
sec:*:5:
cron:*:14:
.
.
.
.users:*:15:billP carsonK raviL annieO
sysadmin:*:16:
tape:*:17:
.
.
.
.

```

- The `/etc/passwd` file consists of rows of one record (row) per user, containing seven fields of user data. See Section 7.3.3 for more information. Example entries are:

```

carsonK:6xl6duyF4JaEI:200:15:Kit Carson,3x192,1-6942,
:/usr/users/carsonK:/bin/sh
annieO:.murv3n1pg2Dg:200:15:Annie Olsen,3x782,1-6982,
:/usr/users/annieO:/bin/sh

```

The example lines are broken to fit the page, and appear as a single line in the file.

- 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.

## 7.3.2 Understanding Identifiers – UIDs and GIDs

Each user account is recognized by a unique number called a user identifier (UID). The system also recognizes each user group by a unique number called a group identifier (GID). The system uses these numbers to track user

file access permissions and group privileges and to collect user accounting statistics and information.

The maximum number of UIDs and GIDs is 4,294,967,294 (32 bits with 2 reserved values). The maximum number of users that can be logged on is determined by the available system resources, but is of course a much smaller figure. If you intend to use the full range of UIDs and GIDs, note that some older utilities and applications do not support the maximum number and you should take the following precautions:

- If you not running the latest versions of your end-user applications, ensure that they support maximum UIDs and GIDs. For example, the widely used Kerberos Version 4.0 does not support UIDs and GIDs beyond a certain range. If you currently use Kerberos Version 4.0, consider upgrading to Kerberos Version 5.0. Similarly, If you use PATHWORKS, consider upgrading to ASU Version 4.0 or higher.
- The System V file system (S5FS) does not support the maximum range of UIDs and GIDs. Any file system `syscall` that specifies UIDs and GIDs greater than 65,535 returns an `EINVAL` error. Users assigned a UID or GID greater than 65,535 cannot create or own files on a System V file system. Consider using the UFS or AdvFS as a solution.
- The behavior of certain commands and utilities 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 incorrectly copies quota files or other sparse files. To correctly copy quota files or other sparse files, use the `dd` command with the `conv=sparse` parameter:

```
# dd conv=sparse if=inputfile of= outputfile
```
  - If you back up a UFS file system that contains quota files or other sparse files using the `vdump` utility and restore it using the `vrestore` utility, the quota files or other sparse files are restored as follows:
    - The first page of a file on disk is restored as a fully populated page; that is, empty nonallocated disk blocks are zero filled.
    - Any additional pages on disk are restored sparse.

### 7.3.3 Understanding the Password File

The `passwd` file for a standalone system identifies each user (including root) on your system. Each `passwd` file entry is a single line that contains seven fields. The fields are separated by colons and the last field ends with a

newline character. The syntax of each entry and the meaning of each field is as follows:

```
username:password:user_id:group_id:user_info:login_directory:login_shell
```

<i>username</i>	The name for the user account. The <i>username</i> must be unique and consist of from one to eight alphanumeric characters.
<i>password</i>	You cannot enter a password directly. Enter an asterisk (*) in the <i>passwd</i> 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.
<i>user_id</i>	The UID for this account. This number must be unique for each user on the system. Reserve the UID 0 for root. Assign each UID in ascending order beginning with 100. Lower numbers are used for pseudousers such as <i>bin</i> or <i>daemon</i> . (See also the <i>/usr/include/limits.h</i> file.
<i>group_id</i>	The GID for this account, which is an integer. Refer to the <i>Technical Overview</i> for information on the limit. Reserve the GID 0 for the <i>system</i> group. Be sure to define the GID in the <i>group</i> file.
<i>user_info</i> (or GECOS data)	This field contains additional user information such as the full user name, office address, telephone extension, and home phone. The <i>finger</i> command reads the information in the <i>user_info</i> field. Users can change the contents of their <i>user_info</i> field with the <i>chfn</i> command. Refer the <i>finger(1)</i> and <i>chfn(1)</i> reference pages for more information.
<i>login_directory</i>	The absolute pathname of the directory where the user account is located immediately after login. The <i>login</i> 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 <i>passwd</i> file. Typically the user's name is used as the name of the login directory. Refer to the <i>chown(1)</i> ,

`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. 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 7.4.2.6 for an explanation of how to do this with Account Manager (`dxaccounts`).

When the `/etc/passwd` file is very large, a performance degradation can occur. If the number of `passwd` entries exceeds 30,000, `mkpasswd` sometimes fails to create a hashed (`ndbm`) database. Because the purpose of this database is to allow for efficient (fast) searches for password file information, failure to build it causes commands that rely on it to do a linear search of `/etc/passwd`. This results in a serious performance degradation for those commands.

If you use the `mkpasswd -s` option to avoid this type of failure, a potential database or binary compatibility problem might arise. If an application that accesses the password database created by `mkpasswd` is built statically (nonshared), that application cannot read from or write to the password database correctly. This causes the application to fail either by generating incorrect results or by possibly dumping core.

Any statically linked application can be affected if it directly or indirectly calls any of the `libc` `ndbm` routines documented in the `ndbm(3)` reference page and then accesses the password database. To remedy this situation, you must relink the application. To avoid this compatibility problem, do not use the `mkpasswd -s` option.

---

**Note**

---

In an NIS environment you can add a user account to either the local `passwd` file or the NIS distributed `passwd` file. Accounts added to the local `passwd` file are visible only to the system to which they are added. Accounts added to the NIS distributed

`passwd` file are visible to all NIS clients that have access to the distributed file. Refer to `nis_manual_setup(7)` for more information on adding users in a distributed environment.

---

### 7.3.4 Understanding the Group File

All users are members of at least one group. The `group` file identifies the group name for a user. There are two primary reasons to group user accounts:

- Several users work together on the same files and directories; grouping these users together simplifies file and directory access.
- Only certain users are permitted access to system files or directories; grouping them together simplifies the identification of privileged users.

The `group` file is used for the following purposes:

- To assign a name to a group identification number used in the `passwd` file
- To allow users to be members of more than one group by adding the user account to the corresponding group entries

Each entry in the `group` file is a single line that contains four fields. The fields are separated by colons, and the last field ends with a newline character. The syntax of each entry and the meaning of each field is as follows:

```
groupname: password: group_id: user1 [user2,...,userN]
```

<i>groupname</i>	The name of the group defined by this entry. The <i>groupname</i> consists of from one to eight alphanumeric characters and must be unique.
<i>password</i>	Place an asterisk (*) in this field. Entries for this field are currently ignored.
<i>group_id</i>	The group identification number (GID) for this group, which is an integer. Refer to the <i>Technical Overview</i> for information on the limits. Reserve the GID 0 for the system. The GID must be unique.
<i>user</i>	The user account belonging to this group, identified by the user name defined in the <code>passwd</code> file. If more than one user belongs to the group, the user accounts are separated by commas. The last user account ends with a newline character. A user can be a member of more than one group.

There is a limit to the number of groups that a user can be in, as documented in `group(4)`. The maximum line length is `LINE_MAX` as defined in the `/usr/include/limits.h` file. User accounts should be divided into a number of manageable groups.

You can also set defaults for certain GID values using the graphical or command-line utilities. See Section 7.4.2.6 for an explanation of how to do this with Account Manager (`dxaccounts`).

## 7.4 Administering User Accounts

The following sections describe how to:

- Administer user accounts using the SysMan Menu options. This method also allows you to add users in NIS (Network Information Service) environments. Invoking the SysMan Menu and selecting the Manage Local Users option is described in Section 7.2.3.
- Administer local and NIS users and associated Windows NT domain accounts using `dxaccounts`. Invoking `dxaccounts` is described in Section 7.2.2.

The process for using the `useradd` command-line utility is similar and is documented in the reference pages but does not support NIS accounts. Refer to the *Network Administration: Services* guide 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. To preserve the integrity of system files, avoid using manual methods of adding user accounts.

---

### 7.4.1 Using the SysMan Menu Accounts Options

The following sections describe how you create new accounts using SysMan Menu options. The following tasks are described:

- Gathering account information, described in Section 7.4.1.1.
- Account options are described in Section 7.4.1.2 and apply to both Local and NIS accounts.
- Filter options for searching accounts are described in Section 7.4.1.3 and apply to both local and NIS accounts.
- Creating or modifying local user accounts is described in Section 7.4.1.4.
- Deleting local user accounts is described in Section 7.4.1.5.

- Creating or modifying NIS user accounts is described in Section 7.4.1.6.
- Deleting NIS user accounts is described in Section 7.4.1.7.

For information on how you use the keyboard to enter information into fields on SysMan Menu utilities, invoke the online help.

#### **7.4.1.1 Gathering Account Information**

To prepare for administering accounts, gather the information on the worksheet provided in Table 7–2. If enhanced security is in use, the data items must comply with the minimum requirements (such as password length). Refer to the *Security* guide for more information. Items marked O in the table are optional.

Refer to Section 7.3.3 for an explanation of the passwd file data items.



**Table 7–2: Account Administration Worksheet**

<b>Field</b>	<b>O</b>	<b>Description</b>	<b>Data Item</b>
User Name			_____
Comments (gecos)	<input type="checkbox"/>	Full name	_____
	<input type="checkbox"/>	Location	_____
	<input type="checkbox"/>	Telephone	_____
User ID (UID)		Can be assigned automatically	_____
Password		Use mixed case or alphanumeric	_____
Primary Group		Can be assigned automatically	_____
Secondary Group(s)	<input type="checkbox"/>		_____
Shell	<input type="checkbox"/>	Can be chosen	_____
Home Directory		Can be created automatically	_____
Lock Account	<input type="checkbox"/>		_____
Local User	<input type="checkbox"/>		_____
NIS User	<input type="checkbox"/>		_____
Windows User	<input type="checkbox"/>	Shares needed	_____

An example of typical user data is provided in Table 7–3.

**Table 7–3: Account Administration Worksheet**

Field	O	Description	Data Item
User Name			carsonK
Comments (gecos)	<input type="checkbox"/>	Full name	Kit Carson
	<input type="checkbox"/>	Location	Office 3T-34
	<input type="checkbox"/>	Telephone	4-5132
User ID (UID)		Can be assigned automatically	Use next available
Password		Use mixed case or alphanumeric	Use site specific initial password
Primary Group		Can be assigned Automatically	Users
Secondary Group(s)	<input type="checkbox"/>		marsx, 25
Shell	<input type="checkbox"/>	Can be chosen	ksh
Home Directory		Can be created automatically	/usr/marsx/carsonK
Lock Account	<input type="checkbox"/>		no
Local User	<input type="checkbox"/>		no
NIS User	<input type="checkbox"/>		yes
Windows User	<input type="checkbox"/>		yes, share \\maul\astools

#### 7.4.1.2 Setting Filter and Display Options

Use the SysMan Manage local users Options... button to configure filtering (described in Section 7.4.1.3) and display options. To set options, invoke the SysMan Menu and choose the Manage Local Users option as described in Section 7.2.3.

When you click on the Options... button the SysMan Account Management: Program Options window is displayed and you can configure the following settings. Note that some option names are truncated here and appear as a descriptive line in the window:

On startup....

Use this option to set a trigger value for the filter feature. The default setting is 200 user accounts.

This feature is useful if you have many hundreds or thousands of user accounts. The more accounts that you have on your system, the

longer it takes any SysMan Accounts task to find and display all the accounts. Setting a trigger value causes the SysMan Accounts task to default to filter (search) mode on startup. This enables you to choose a specific account or group of accounts and to greatly reduce the search and display time.

For example, if you set a figure of 300 user accounts, SysMan Accounts defaults to filter mode only when you have more than 300 accounts.

#### UserName

This checkbox enables display of the user's account name in all account listings.

#### Userid (UID)

This checkbox enables display of the user identifier (UID) in all account listings.

#### Comments

This checkbox enables display of any account comments (such as location and telephone number) in all account listings.

Note that selecting checkboxes affects your filter options. You can filter accounts based only on the data displayed.

### 7.4.1.3 Using Filter Options

If you have a large number of accounts you can use the Filter... option to quickly find a particular account or group of accounts. You can invoke the filter automatically, depending on the settings in Options... (described in Section 7.4.1.2). Automatic invocation enables you to avoid a delay while the Account Manager finds and loads all the user account data. You can filter both local and NIS accounts using this feature.

To use the search and filter option, invoke the SysMan Menu and choose the Manage Local Users option as described in Section 7.2.3. Click on the Filter... option button to display a dialog window titled: Manage Local Users: Show. Using this window, you can perform simple and advanced searches.

To perform a simple search:

Enter a filter (a search string) or a set of filters. All simple searches are based on account names entered as follows:

- An individual user name such as `s_kahn`
- A wildcard pattern, such as `*khan` or `?_khan`

- A comma-separated list of user names or wildcard patterns, such as \*khan, kim, donny\_w

Any accounts matching the filter specification are listed in the Manage Local Users window, with the original filter string identified at the top of the window.

To perform an advanced search:

Click on the Advanced button to display the additional filter options. Activate a search option by clicking on the checkbox.

The filter options are:

User name or filter...

Enter a filter as described for the Simple Search option.

User ID range...

Enter a restricted range of UIDs such as 1-100 or an open-ended range such as 100- to find all accounts with a UID greater than 100.

Pattern in the “comments”...

Enter a search pattern to search on data entered in the Comment (GECOS data) field when the user’s account was created.

This might be a telephone number, a physical location, or other user-specific information. You can use the asterisk (\*) or question mark (?) wildcards to define a pattern. For example; *\*string\**, such as: *\*Sub\**.

LOCKED or UNLOCKED search criteria...

This option enables you to include (or exclude) locked or unlocked accounts. You can use this option to identify all currently locked accounts.

A warning dialog is displayed if you do not clear the contents of the Simple Search before invoking an Advanced Search. If you see this warning dialog, click on OK to accept the Advanced Search. This action supersedes any search criteria that you specified in the Simple Search.

#### 7.4.1.4 Creating or Modifying Local Accounts

To create a new account, invoke the SysMan Menu and choose the Manage local users option as described in Section 7.2.3. A table of users is displayed,

listing all the existing local user accounts. Use the following procedure to add a local user:

1. Choose the Add... option to display the Manage Local Users: Add a User window.
2. Complete the data fields using the information from the worksheet in Table 7–2.
3. If additional NIS options are required, choose Options and enter the appropriate NIS values. Then click on the OK button to return to the Add a User window.
4. click on the OK button to add the new user. You are prompted to correct any errors, such as mistyped password confirmations.
5. The Local Users window is displayed, showing a confirmation message. Click on the OK button to return to the SysMan Menu.

To modify an existing account, invoke the SysMan Menu and choose the Users option as described in Section 7.2.3. 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 choose the entry you want to modify.
2. Choose 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, choose Options and enter the appropriate NIS values. Then click on the OK button to return to the Add a User window.

To add or modify more than one account, click on the Apply button instead of the OK button. (All changes are deferred until you click on the OK button to exit.)

5. Click on the OK button to confirm the changes. You are prompted to correct any errors, such as mistyped password confirmations.
6. The Local Users window is displayed, showing a confirmation message. Click on the OK button to return to the SysMan Menu.

Online help provides explanations for the fields, and defines valid data.

#### **7.4.1.5 Deleting Local Accounts**

Before deleting accounts consider the following:

- As an alternative to deletion, you can use the Modify... option to lock an account. You can also transfer the account to another new user using the Modify... option to change some account details.
- You can invoke the `dxarchiver` utility before deleting the account to create a compressed archive file of the user's directories and files. See the `dxarchiver(8)` reference page for more information.

To delete an account, choose the Manager Local Users option as described in Section 7.2.3. 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 choose the user account that you want to delete.
2. Choose 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. Click on the OK button to delete the account. The list of local users is updated immediately.

#### 7.4.1.6 Creating or Modifying NIS Accounts

To create a new NIS account, invoke the SysMan Menu and choose the Manage NIS Users option as described in Section 7.2.3. 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. Choose 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. Click on the OK button to add the new user. You are prompted to correct any errors, such as mistyped password confirmations.
4. The Manager NIS Users window is displayed, showing a message confirming the successful addition. Click on the OK button to return to the SysMan Menu.

To modify an existing account, invoke the SysMan Menu and choose the Manage NIS Users option as described in Section 7.2.3. 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 choose the user account that you want to modify.

2. Choose the Modify... option to display the Manage NIS Users: Modify a User window.
3. Change the contents of data fields as required.
4. Click on the OK button to confirm the changes. You are prompted to correct any errors, such as mistyped password confirmations.  
To add more than one account, click on the Apply button instead of the OK button. All changes are deferred until you choose the OK button to exit.
5. The Local Users window is displayed, with a message confirming the successful addition. Click on the OK button to return to the SysMan Menu.

Online help provides explanations for the fields, and defines valid data.

#### 7.4.1.7 Deleting NIS Accounts

To delete NIS accounts, choose the Manage NIS Users option as described in Section 7.2.3. 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 choose the account that you want to delete.
2. Choose 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. Click on the OK button to delete the account. The list of NIS users is updated immediately.

#### 7.4.2 Using the Account Manager (dxaccounts)

Invoke the Account Manager (`dxaccounts`) as described in the quick start instructions in Section 7.2.2. 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. (Refer to the *Network Administration: Services* guide for more information on NIS.)

Most options require root privilege because they impact the user account databases. Options that do not affect the databases are available to all users. An example of such an option is Find, which you use to locate accounts.

When ASU is installed, additional options are displayed in the `dxaccounts` windows that enable you to administer accounts in Windows NT domains and create associated UNIX accounts simultaneously. Refer to the *Installation and Administration Guide* for more information on ASU.

If Enhanced (C2) security is enabled, additional options enable you to retire and disable accounts according to the security settings in force. Refer to the *Security* guide for more information.

#### 7.4.2.1 Adding and Modifying Accounts

You use the Account Manager on `<host>` dialog window to add or modify user accounts as follows:

- To create a new account, click on the Add button.
- To modify an existing account, double-click on the user's icon. If there are many accounts, you use the options described in Section 7.4.2.3 to find accounts.
- You can also copy (clone) a new account from an existing account, as described in Section 7.4.2.4.

Use the following procedure to add or modify accounts:

1. If the current view is not Local Users, pull down the View menu and choose the Local Users option.
2. Click on the Add button to display the Add/Modify Local User window and click on 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. Choose the next available UID, or enter a new UID.

If you modify a user's UID with the Account Manager (`dxaccounts`), the ownership of the user's files and subdirectories does not change and, under certain circumstances, the home directory ownership might not change either. For example, if you change the UID of user johndoe from 200 to 201, the files and subdirectories under his home directory still belong to UID 200. Furthermore, if johndoe does not own his home directory, the ownership of that directory does not change either. To avoid this problem, use the `chown` command to change the directory and files, if applicable.

5. Use the pull-down menu to choose the primary group, or clear the box and type a group name.



If secondary groups are required, click on the Secondary Groups... button. In the Secondary Groups window, double-click on any required local or NIS (if available) groups.

6. Choose 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. Click on 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. Click on the OK button to create the account and return to the Account Manager main window. You are prompted to correct any errors. The Current View is updated with an icon for the new user.

#### 7.4.2.2 Deleting Accounts

Invoke the `dxaccounts` utility as described in Section 7.2.2. The Account Manager on `<host>` window is displayed first.

1. Double-click on the icon of the account that you want to delete. If there are many accounts, use the options described in Section 7.4.2.3 to find accounts.
2. Click on the delete button. The Delete Local UNIX User window is displayed. You can remove the user's files and directories at this time. (You may want to archive these. See the `dxarchiver` option.)
3. Click on the OK button to confirm the deletion and return to the Account Manager on `<host>` window. This window is immediately updated, removing the deleted user account.

#### 7.4.2.3 Finding and Selecting Accounts

The `dxaccounts` utility provides a useful search feature that you can use to locate user accounts. You can also use this feature to choose groups of users to which you want to apply global changes, such as modifying the user shell or password.

Invoke the `dxaccounts` utility as described in Section 7.2.2. The Account Manager on `<host>` window is displayed first.

1. Click on the Find button.

2. Enter a search string (a text string) in one of the fields and click on the OK button.

The Find option enables you to locate and display all accounts where the data in the search field contains the search string. For example:

1. Enter the string `ad` in the Username field and click on the OK button.
2. The Selected Users window is displayed, stating that the following users matched the search criteria.
3. The matched users include `adm`, `admin`, `adamK`, and `wadmanB`. These user accounts are highlighted in the Current View.

After you select a group of user accounts, you can choose the modify (or delete) option to perform global operations on the selected users.

#### 7.4.2.4 Copying Accounts

You can use existing accounts as templates to create new accounts, enabling you to clone the account properties. You can create an exact duplicate of one or more accounts using the following procedure:

1. Click on an the icon for an existing user account to highlight it, or use the mouse to select a group of accounts.
2. Click on the Copy button to copy the account.
3. Click on the Paste button to create a clone account. The new icon label has the original name, appended with the string `_copy $n$` , where  $n$  represents the sequential number of the copy. You can make as many copies as required.
4. Choose each duplicate account in turn to rename it and to modify its properties as described in Section 7.4.2.1.
5. Make the minimum required modifications to the account as follows:
  - a. Enter the new user name
  - b. Change the UID or choose the next available UID
  - c. Change the password
6. Click on the OK button 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.

You can use the same procedure to clone groups.

When copying user accounts using cut and paste or drag and drop, the Allow Duplicate UIDs option in the General Preferences dialog box is honored. For example, when making a copy of user account that has a UID of 200,

if the Allow Duplicate UIDs check box is off (the default), a unique UID is automatically generated for the resulting copy. If the Allow Duplicate UIDs check box is on, then the copy has an identical UID. The same rule applies to copying groups.

Using MB1 to drag and drop user accounts, groups, or templates it results in a copy operation, not a move operation. This is different from the default CDE behavior, where using MB1 performs a drag and drop move operation and Shift-MB1 performs a copy operation. For example, if you use MB1 to drag a user account from the Local Users view and drop it in the NIS Users view, you create a copy of that user account in NIS. To avoid this problem, delete the original icon after the copy is complete.

#### 7.4.2.5 Using the Password Option

The `dxaccounts` utility provides a password option enabling you to change or remove passwords for a single user or a group of users. Use this option as follows:

1. Choose the user or users whose passwords you want to change. The Find option may be useful in selecting groups of users.
2. From the Edit menu, choose Password.
3. In the New Password window, enter and confirm the new password.  
You can click on the No Password button to remove the current passwords. (There are important system security implications when you choose this option.)
4. Click on the OK button to confirm the change and return to the Account Manager main window.

#### 7.4.2.6 Account Manager (`dxaccounts`) General Options

The Account Manager (`dxaccounts`) enables you to easily set defaults for newly created user accounts. You can also set account defaults using the command line (`useradd`) but you cannot use SysMan Menu Accounts options to set defaults. Use the following procedure to add or modify defaults:

1. From the Options menu, choose General..... The General Options window is displayed, enabling you to set the following defaults:
  - Duplicates Policy – These options enable you to allow duplicate User Identifiers (UID) and Group Identifiers (GID).
  - ID Ranges Policy – These options enable you to control the minimum, next, and maximum UID and GID.
  - Default Primary Group – This option enables you to set the default primary group to a group other than users.

- Default Primary Group – This option enables you to set the default home directory to a location other than `/usr/users`.
  - Default Shell for User – This option enables you to set the default login shell.
  - Default Primary Group – This option enables you to set the default skeleton directory path to a location other than `/usr/skel`.
  - Use Hashed Password Database – This option forces the creation of a hashed (encrypted) password database.
  - Require Password For New Accounts – This option forces the entry of a password each time an account is created.
  - Synchronize UNIX and Windows NT domain accounts – This option forces the automatic creation of an account when the UNIX account is created.
2. After you make the required changes, click on the OK button to update the defaults and return to the Account Manager main window.

## 7.5 Administering Groups

The following sections describe how to:

- Administer groups using the following SysMan Menu Accounts options:
  - Manage local groups
  - Manage NIS groups
- Administer groups using the Account Manager (`dxaccounts`).

You can also use the `groupadd`, `groupmod`, and `groupdel` command-line interfaces to administer groups. Refer to the documentation specified in Section 7.1.3 for more information on command-line options.

---

### Note

---

Avoid using the `addgroup` utility as it does not provide all the available options and is not sensitive to security settings.

To preserve system file integrity, avoid using manual methods of adding user accounts.

---

### 7.5.1 Using the SysMan Menu Accounts Group Options

The following sections describe how to administer groups using SysMan Menu options. The following tasks are described in this section:

- Creating a new local 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, the data items must comply with the minimum requirements. Refer to the *Security* guide for more information.

Refer to Section 7.3.4 for an explanation of the group file data items. In the SysMan Menu options, you can also specify default values for NIS groups. Refer to the *Network Administration: Services* guide for information on configuring NIS.

In Table 7–4 the data items marked O are optional. You must specify at least one user account.

**Table 7–4: Group Administration Worksheet**

Field	O	Description	Data Item
Group Name			_____
Password		Not currently used.	_____
Group Identifier (GID)		If unused, the next number is assigned automatically	_____
User			_____
User	O		_____
User	O		_____
User	O		_____
User	O		_____
User	O		_____

### 7.5.1.2 Creating or Modifying Groups

To create a new group, invoke the SysMan Menu and choose the Manage local groups option as described in Section 7.2.3. The Local Groups table is displayed, listing all the existing local groups. The process for adding NIS groups is identical, except that you choose the Manage NIS groups option.

Use the following procedure to create a group:

1. Choose the Add... option to display the Manage local groups: Add a Group window.
2. Complete the data fields using the information from the worksheet in Table 7–4.
3. On the Members panel, highlight the names of users who are the initial members of the new group. This action is optional.
4. Click on the OK button to add the new user. You are prompted to correct any errors.
5. The Local Groups table window is displayed, with a message confirming the successful addition. Click on the OK button to return to the SysMan Menu.

To modify an existing group, invoke the SysMan Menu and choose the Manage local groups option as described in Section 7.2.3. The Local Groups table is displayed, listing all the existing local groups. Use the following procedure to modify a group entry:

1. Scroll through the list of groups and choose the group that you want to modify.
2. Choose 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. Click on the OK button to confirm the changes.

To modify more than one group, click on the Apply button instead of the OK button. (All changes are deferred until you click on the OK button to exit.)

5. The Local Groups window is displayed, with a message confirming the successful modification. Click on the OK button to return to the SysMan Menu.

Online help provides explanations for the fields, and defines valid data.

## 7.5.2 Using the Account Manager (dxaccounts)

Invoke the Account Manager (`dxaccounts`) utility as described in Section 7.2.2. The Account Manager on `<host>` window is displayed first. Using the data from the worksheet in Table 7–4, use the procedures in the following sections to add, modify, and delete groups when using `dxaccounts`. The process for administering NIS groups is identical to the process for administering Local Groups, except that you must be authorized

to change the NIS databases. You can still use any options, such as Find, that do not change the databases.

If there are many groups on your system, use the Find option described in Section 7.5.2.4 to locate groups that you want to modify or delete.

### 7.5.2.1 Adding Groups

Add group 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. Choose the next available GID or enter a new GID.
5. Double click on any user name to add that user to the group. This action is optional.
6. Click on the OK button to add the group and return to the Account Manager on <host> window. This window is immediately updated with an icon for the new group.

An alternative method of adding a new group is to clone it from an existing group as follows:

1. Click on an existing group icon to highlight it.
2. Choose the Copy button to copy the group.
3. Choose the Paste button to create a new version of the group. The new icon label has the original name, appended with the string `_copy $n$` , where  $n$  represents the sequential number of the copy. You can make as many copies as required.
4. Double click on the newly copied icon to highlight it and display the Add/Modify Local UNIX group window. The Modify button is selected automatically.
5. Make any required modifications to the group as follows:
  - a. Enter the new group name
  - b. Change the GID, or choose the next available GID
  - c. Add or delete members
6. Click on the OK button 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 7.2.2. The Account Manager on `<host>` window is displayed first. Use the following procedure to modify a group:

1. Double-click on the group that you want to modify. The Add/Modify Local UNIX group window is displayed.
2. Make any required modifications to the group. For example:
  - Rename the group
  - Change the GID
  - Add or delete members
3. Click on the OK button to confirm the changes and return to the Account Manager on `<host>` window. This window is immediately updated with any changes for the group.

### 7.5.2.3 Deleting Groups

Invoke the `dxaccounts` utility as described in Section 7.2.2. The Account Manager on `<host>` window is displayed first. Use the following procedure to delete a group:

1. Click on the group that you want to delete.
2. Click on the delete button. You are prompted to confirm that you want to delete this group.
3. Click on the Yes button 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 (`dxaccounts`) enables you to locate groups and users who are members of groups.

Invoke the `dxaccounts` utility as described in Section 7.2.2. The Account Manager on `<host>` window is displayed first. To find a group:

1. Click on 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 GID (integer) – Any number entered is treated as a string. The Find option selects and displays all groups where the GID contains this string. For example, the string 20 is matched to groups 20 and 220.
- A user name (text string) – The Find option selects and displays all groups with users whose user name contains this string. For example, the string wal is matched to groups containing users named wallyB and cadwalZ.

## 7.6 Administering Windows Domain Accounts and Groups

When the Advanced Server for UNIX (ASU) is running, the account management utilities can be configured to support the creation and administration of Windows domain accounts. For information on installing and configuring ASU, refer to the ASU *Installation and Administration Guide*. When ASU is installed, you can use the account management utilities to perform certain operations on associated (synchronized) accounts. These are accounts for the same user that exist both in the Windows domain and the UNIX environment, and are referred to as synchronized accounts in the UNIX utilities. For specific information on Windows 2000, see Section 7.6.2.

To configure a UNIX system to create associated Windows NT domain and UNIX accounts, and to set the default account creation options, you must set the account environment variables using the `usermod` (or `useradd`) command as shown in Example 7-1.

---

### Note

---

When ASU is installed and configured, the creation of associated Windows NT domain and UNIX accounts is enabled by default. All account management utilities have their PC support features enabled. The value of the `Synchronized UNIX/PC Accts` environment variable is one (1), which indicates that the setting is on.

---

### Example 7-1: Changing the Default Environment Variables Using `usermod`

---

```
# usermod -D 1

Local                = 1
Distributed          = 0
Minimum User ID      = 12
Next User ID         = 200
Maximum User ID      = 4294967293
Duplicate User ID    = 0
Use Hashed Database  = 0
```

### Example 7-1: Changing the Default Environment Variables Using usermod (cont.)

---

```
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 environment variables.
- [2] The output from the usermod command is a list of default values for the environment variables. When you create an account, these values are assigned to the new account. For example, all new accounts are created in the base home directory of /usr/users.
- [3] This command specifies new default values for three environment variables that apply only to Windows NT domain accounts.
- [4] This (truncated) list shows the new default values for the environment variables, which are as follows:
  - pc\_synchronize=1 – Creates associated Windows NT domain and UNIX accounts if ASU is running
  - pc\_passwd\_uniqueness=1 – Forces validation of the password for uniqueness

### Example 7–1: Changing the Default Environment Variables Using `usermod` (cont.)

---

- `pc_max_passwd_age=60` – Specifies the maximum number of days that can elapse before a password must be changed by the user
- 

Use the `groupmod -D` command to set the default environment variables for creating new groups. You can also specify alternate values for the environment variables when you create a new account, overriding the defaults. For more information, refer to the `useradd(8)`, `usermod(8)`, and `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 environment variables for newly created accounts.

You cannot use ASU account management utilities to perform operations on UNIX-only accounts, or use UNIX utilities to perform operations on accounts that exist only in the Windows NT domain. The following sections provide information on how the UNIX and ASU account administration utilities behave when ASU is running and when you are administering synchronized accounts.

#### 7.6.1 Administering Synchronized Accounts

If you have set up ASU and configured the creation of synchronized accounts, certain features in the account administration utilities are enabled automatically. The following sections describe how those features appear in the different account management utilities.

A lock file prevents you from using two different utilities (or two instances of the same utility) at the same time. This scenario 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 informs you that the data files are locked.

When using the ASU utilities to add accounts, ASU detects the presence of the lock file, and is unable to create an associated UNIX account. It only creates a Windows NT domain account. No lock file error message is displayed, and you do not receive a confirmation that the associated account was not created. When using ASU tools, 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 user interfaces for SysMan Menu Accounts utilities do not show any changes when ASU is running. If synchronized accounts are enabled, there are no differences in the windows and screens. However the following changes in behavior should be noted:

- Add a user – You can choose from several `DOS----` groups when assigning the account holder to a group as part of account creation (the Primary Group option).

If the creation of associated Windows NT domain accounts is enabled as described in Example 7–1, the associated account is created automatically and you cannot override its creation.

- Delete a user – The associated Windows NT domain account is deleted automatically. You cannot override this deletion. If you want to retain the users' Windows NT domain account, do not perform this operation.
- Add/Modify a group – Several `DOS----` groups are included in the selection list of groups, showing the default Windows NT domain accounts, such as `lanman` and `lmsadmin`.

Refer to Chapter 1 for information on using the SysMan Menu.

### 7.6.1.2 Using the Account Manager (`dxaccounts`)

The Account Manager (`dxaccounts`) is an X11-compliant GUI and as such can be used only in an X-window user environment such as CDE. The `dxaccounts` main window provides an option to create **PC** (Windows NT domain) accounts. This option is grayed out and unusable unless ASU is running. When ASU is running, the following features are available:

- When creating an account in one user environment, such as the Windows NT domain, you can choose to create a synchronized account in the other user environment, such as UNIX.
- You can choose not to create an associated Windows NT domain account or UNIX account, even if creation is enabled by default as shown in Example 7–1.
- Additional options appear on the View menu, enabling you to display all Windows NT domain accounts and groups. When you choose these options, the PC (Windows NT domain) user and group accounts icons are displayed. You can add, modify, and delete PC accounts and groups as if they were UNIX accounts.
- From the Options menu, you can use the PC Defaults option to set characteristics that are inherited by any newly created account. You use

the General Options menu item to set account synchronization and to set characteristics for UNIX accounts.

- When removing accounts with the Delete button, you are prompted to choose the UNIX account, the PC account, or both.
- When using the View menu, Local Groups option, the PC groups (DOS----) are visible and you can perform administrative tasks on these groups.
- When using the View menu, PC Groups option, the PC domain groups are visible and you can perform administrative tasks on these groups.

You use the processes described in Section 7.5.2 to perform administrative operations on PC accounts and groups.

The advantage of using `dxaccounts` is that it is a native X11 application and can use the features of the windowing environment such as 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 runs only in an X-window user environment such as CDE.

### 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, when using the commands you can choose to override the default environment variables and specify customized values for new accounts.

When ASU is installed, the following account and group creation options become available for use:

- `useradd`, `usermod` – The following extended options are provided to set the default Windows NT domain account characteristics using the `-D` option. Also shown are the default values:
  - `pc_synchronize=` (value: 1, on) – Use this option to determine whether synchronized accounts are created by default when a new account is created either for the Windows NT domain or on a UNIX system. Synchronized accounts are not created if this value is zero.
  - `pc_min_password_age=` (value: 0, off) – Use this option to specify how many days must elapse before a password can be changed. The user is not allowed to change passwords more frequently than this.
  - `pc_max_password_age=` (value: 42 days) – Use this option to specify how many days can elapse before a password must be changed. The user must change passwords at least this frequently.

- `pc_passwd_uniqueness=` (value: 0, off) – Use this option to force 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 is logged out automatically after a certain time when the account expires.

You invoke these extended options with the `-D -x` options, as shown in Example 7–1. To override the default characteristic, you specify the extended option with the `-x` flag during an account administration operation, such as account creation:

```
# useradd -x pc_passwd_uniqueness=1 guest9
```

The following command options are not extended options and do not set default account characteristics. These account characteristics can also be created using the ASU utilities. Use these command options when adding a new account:

- `pc_username=`*name\_string*  
The user account name in the Windows NT domain. This can be identical to, or different from, the user's UNIX account name.
- `pc_unix_username=`*login\_name*  
The synchronized UNIX account name. If no name is entered, it is the same as the Windows NT domain account name.
- `pc_fullname=`*text\_string*  
The full name of the user or a description of the account.
- `pc_comment=`*text\_string*  
A brief description of the account that can be changed only by the administrator.
- `pc_usercomment=`*text\_string*  
A brief description of the account. This string can be changed by the user.
- `pc_homedir=`*pathname*  
The path to the user's home directory, specified as a Windows NT share format.
- `pc_primary_group=`*group*  
The primary group (Windows NT domain) to which the user belongs.
- `pc_secondary_groups=`*group,group...*  
The secondary Windows NT domains to which the user belongs. This value is specified as a comma-delimited list.
- `pc_logon_workstations=`*client\_name*

A list of client host systems from which the user can log on. This value is specified as a comma-delimited list. A null value (" ") means that the user can log on from all workstations.

- `pc_logon_script=pathname`  
The directory where the default logon script is located. (This directory is created during ASU configuration.)
- `pc_account_type=local/global`  
Specifies whether the account is a local or global account in the Windows NT domain.
- `pc_account_expiration=date_string`  
Specifies the date on which the account will expire and logins are prevented.
- `pc_logon_hours=Dd0000-0000,Dd0000-0000...`  
Specifies the days of the week and hours of the day during which logins will expire and logins are permitted or denied.
- `pc_user_profile_path=pathname`  
Specifies the pathname to the default user profile directory.
- `pc_disable_account=0/1`  
Specifies whether the account is initially locked, disabling logins.
- `pc_passwd0/1`  
A text string used as the initial account password. You must precede this option with the `-x` flag and you are prompted to enter a password, and then confirm the entry. The password 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 you can use with this command is Synchronized UNIX/PC Accts.  
Use this option to delete synchronized accounts, as follows:  

```
# userdel -r -x pc_synchronize=1 studentB
```
- `groupadd, groupmod`  
The following extended options can be used with the `-x` flag to administer groups in Windows NT domains:
  - `pc_group_description=string`  
Specifies a text string that provides a description of the group.

- `pc_group_members=user,user...`

Specifies a comma-delimited list of group members.

The advantage of using the command-line is that it offers complete control over administrative tasks, enabling you to specify any and all command options and override the default account environment variables.

Commands can be used as part of a shell script to customize and automate account creation. However, the command options can be lengthy, so it is often easier to set up an account using the graphical utilities.

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 must be installed on and can only be used from a system running Windows NT. It provides the same features as the `net` command-line options.

You can specify default environment variables for all newly created accounts. These environment variables are referred to as account policies in the Windows NT domain.. You cannot set the default environment variables for synchronized UNIX accounts when using the User Manager for Domains (`usrmgr.exe`).

Refer to the ASU *Installation and Administration Guide* and the User Manager for Domains online help for more information.

#### 7.6.1.5 Using ASU net Commands

ASU provides an extensive set of `net` commands that you enter on the UNIX command line or from a DOS window on a Windows NT server.

For example, the following command displays the help for `net user`, the command you can use to add, modify, or delete user accounts:

```
# net help user | more
```

The syntax of this command is:

```
NET USER [username [password | \*] [options]]
          username [password | \*] /ADD [options]
          username [/DELETE]
```

```
.
.
.
```

```
# net user josef /add
```



Enter the following command to display a list of `net` command options:

```
# net help view
```

See the *Installation and Administration Guide* and `net(1)` for more information on using `net` commands.

## 7.6.2 Windows 2000 Single Sign-On

If your local computing environment consists of UNIX servers and Windows 2000 client systems, and you have one or more domain controllers in the environment, you can configure the optional Windows 2000 Single Sign-On (SSO) software. The SSO software enables account holders in the Windows 2000 domain to access computing resources on the UNIX server without needing a separate UNIX account.

The SSO software modifies the Windows Active Directory and the associated Windows account management utilities. These modifications enable administrators in the Windows 2000 domain to record UNIX 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.

### 7.6.2.1 Single Sign-On Installation Requirements

Configuration and use of this feature has the following installation prerequisites:

- You must have root access to the UNIX system and be an administrator of every Windows 2000 domain controller on which the SSO software is to be installed. You must run an installation procedure on the UNIX system and at least one domain controller.
- The UNIX system cannot be running C2 level security. 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.

### 7.6.2.2 Installing the Single Sign-On Software

Install the software as follows:

1. Load the CD-ROM into the reader.
2. Create a mount point and mount the CD-ROM using commands similar to the following:

```
# mkdir /apcd
# mount -r /dev/disk/cdrom4c /apcd
```

3. Locate the installation kits and documentation as follows:

```
# ls /apcd/Windows2000_SSO
```

4. Use the `setld` command to install the software subset named `W2KSS0100`. The configuration script, `/usr/sbin/w2ksetup`, runs automatically when the installation is complete. Complete the configuration as described in the *Windows 2000 Single Sign-On Installation and Administration Guide*.

### 7.6.2.3 UNIX Requirements for Creating Single Sign-On Accounts

The following requirements for UNIX account characteristics apply to SSO accounts:

- You can 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 7.3.3 and subsequent sections for a description of UNIX account attributes.

Prepare the following account data for each user or group. If necessary, use the UNIX account management tools described in this chapter to ensure that the account data is of an appropriate format and is unique for each user:

### *Username*

In Windows 2000, the *Username* is the user logon name. For SSO it must meet two requirements; length and uniqueness. This also applies to group names.

Windows 2000 can support very long user names although in practice most users prefer short adaptations of their name and initials, which are easier to remember and type. The maximum length of the user name is determined by the current restriction to eight characters in the UNIX environment.

The actual name can be as short as the user's initials but must be unique on both systems for every user. If a user with only a UNIX account has the user name *chs*, you cannot assign that name to an SSO account.

### *Password*

Each user requires a password. You determine the length of the password by the current settings on the UNIX system. These settings can vary depending on the security mechanisms in force. 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 7.3.2 and Section 7.3.4 for a description of these identifiers.

### *User Comment*

This field enables you to enter a text description of the 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 7.3.1 for more information.

#### 7.6.2.4 Creating Single Sign-On Accounts and Groups

Using the information prepared in Section 7.6.2.3, create SSO accounts as follows:

1. Log in to your administrator's account on the Windows 2000 domain controller.
2. Invoke the Microsoft Management Console (MMC) interface and display the Active Directory Users and Computers Window.
3. Open the `Users` folder and either choose an existing user or open the Action menu and choose the New option then the User option.
4. Three dialog boxes 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 7.6.2.3, such as the UID and GID.

To create an SSO group use the same procedure, selecting the New and Group menu options in step 3.

#### 7.6.2.5 Single Sign-On System Files

When you install and configure the software, the following system files are created:

- The `ldapcd` daemon, which is the connection to the registry of account information on the domain server. If the daemon is 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.

---

## 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 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 might 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 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 *Installation Guide*. 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 receives all inbound remote print requests (on socket 515) instead of LPD. However, with this configuration, local `lpd` commands such as `lpr` do not work.

### 8.1.2 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.) Any user on the local system can access the printer.
- **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.

- **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 automatically displays a list of all the supported printers and enables you to configure them quickly.

---

**Note**

---

There are restrictions on using `printconfig` with older `/etc/printcap` files. See `printcap(4)` for information.

---

- The `lprsetup` utility. This utility provides backwards compatibility 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. If you select the command-line option when invoking `printconfig`, the `lprsetup` utility is 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

You might want to manage printers by editing the system files. For example, if you are working on printer driver development, it is necessary to create test entries in the printer configuration file

`/etc/lprsetup.dat`. Alternatively, you might 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 assists 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 provides information that you might need to provide to the configuration utility to use any special capabilities of the printer, such as tray selection.  
  
Usually, you see your printer included in the list of supported devices when you use an installation utility (or if you look in `/etc/lprsetup.dat`). If it is not listed there, and the manufacturer does not provide information for using the printer, use the generic settings provided by the configuration tools. Access to printer capabilities is often restricted when you use a generic configuration.
  - The *Network Administration: Connections* guide provides information connections used by 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)` describes a generic internationalized print filter for PostScript printers, which enables you to support local-language PostScript printing. The `pcfof(8)` 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. 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/sbin` directory contains the print filters for each known printer device. These filters are specified during printer installation, for example `/usr/sbin/ljn09of` 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 are 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/sbin/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 `lpc(8)` for information.
- The `pac` utility formats the data from printer accounting log files and displays it or stores it in a text file. See `pac(8)`. See `acct(8)` for information on accounting.
- CDE Application Manager — Desktop Apps contains the Print Manager graphical user interface, which enables you to perform the following tasks. See 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.

- 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 functions as described in the owner's manual. A good strategy for avoiding installation problems is to accept the default data presented by the configuration utility. 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 enter when using the `lprsetup` or `printconfig` utilities. The information required depends on whether the printer is remote, a direct connection, or a network connection via LAT or TCP/IP:

If your system is part of a network, you might need to consult your local network administrator or see the *Network Administration: Connections* and *Network Administration: Services* guides 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) – Determine this from `lprsetup.dat(8)` 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 depends 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: Connections* 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` supplies a default `/dev/lat` port or service name, LAT server node name, or LAT port name. (See the *Network Administration: Connections* guide for more information.)
  - If the connection type is TCP, you need the TCP address, in one of two formats. The owner’s manual of the printer specifies 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 might 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)` 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. You can sometimes increase the default rate 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 values for certain page layout characteristics, set the characteristics here. Supported layouts might 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 system name** — The host name for the remote system, such as `alfabet.ccc.nic.com`

- Remote printer name – The name of the printer on remote system, such as lp0, or a valid alias.
- Advanced options – These options are not required for a basic installation. Remote printing allows you to configure a few advanced options (or deconfigured, such as error logging). See the information in Section 8.2.1.1. Most advanced options are not passed on to the remote system.

### 8.2.1.3 Printer Data Worksheet

You can print the following table and use it as a worksheet.

Name / Format	Example/ Location
<b>Printer Type</b> (Alphanumeric string)	ln09 (lprsetup.dat)
<b>Alias</b> (Alphanumeric string)	alfie2 (User-supplied)
<b>Connection type</b> (software option)	TCP (Interface-supplied)
<b>Device pathname, direct connection</b> (file name)	/dev/tty00 (ports(8))
<b>Device pathname (Remote Connection)</b> (port or service name)	556 (lat reference page)
<b>Device pathname, network connection</b> (host/service name)	@alfie.nic.ccc.com/2501 (Printer Documentation, /etc/services file)
<b>Remote system name</b> (host name)	alfabet.ccc.nic.com (Network Administrator)
<b>Remote printer name</b> (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 incompatibilities in older `/etc/printcap` files that might 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. You can then use `printconfig` to modify the configuration later if required.

Invoke `printconfig` from CDE, or from the command line, as described in `printconfig(8)`.

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 is displayed (`lp0` if this is the first printer that you are configuring on this system). In this window you enter:

- The printer alias names, such as the name of the local host or perhaps something to help users to identify the printer type, such as `local_DL5100`.



- The connection type. Because you have connected the cable to a local serial port such as `COMM1`, you must choose the Direct option.
- Finally, you need to specify the Device pathname, which is the special device file that maps to the serial port. The device file `/dev/tty00` maps to the `COMM1` port.
- Choose Commit to write the options to the `/etc/printcap` file.

For a basic printer configuration, that is all you need to do. If you decide to use advanced options, such as setting print job limits, choose the Advanced option to display the “Printer Config: Local Printer Settings: Advanced” window. See Section 8.4.

After committing the configuration, you are returned to the Configuration on *host name* window, and the printer now appears in the list of configured printers. Use the Test option to print a test page to the printer. If you do not get any output, review the data carefully and 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 defaults to this printer. Generally, you 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’s capabilities 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. You can connect remote printers 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 system, list the local system (the client) in the `hosts.lpd` file or `hosts.equiv` file of the remote system (the host).

If your printer is 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: Connections* guide for information on how to enable remote LAT terminal server printing.

Refer to 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 is 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 system 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 committing the configuration, you are returned to the Configuration on *host name* window, and the printer now appears in the list of configured printers. Use the Test option to print a test page to the printer. If you do not get any output, review the data carefully and refer to the troubleshooting section.

## 8.2.4 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 you must register it with a TCP/IP

node name and node address. You can also use TCP/IP to communicate with print server boxes or terminal servers.

With TCP/IP printing, the local host manages print jobs in the same way as it does 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`).

#### 8.2.4.1 Setting Up TCP/IP Printing Using `printconf`

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 8.2.4. Gather the information as described in Section 8.2.

Invoke `printconf` 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 is 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`.

You can also specify this value as `@nodename/servicename`, where `servicename` is defined in `/etc/services` and associated with the printer's TCP/IP port.

- The connection type, which is TCP.
- Finally, you need to specify the device pathname as a port number or service, which in this case is the network address of the printer, such as `@alfie.nic.ccc.com/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 committing the configuration, you are returned to the Configuration on *host name* window, and the printer now appears in the list of configured printers. Use the Test option to print a test page to the printer. If you do not get any output, review the data carefully and refer to the troubleshooting section.

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 defaults to this printer. Generally, you 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.

#### 8.2.4.2 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 listens for connection requests. You can either specify a name that is defined in the `/etc/services` file, or directly specify the port number assigned to the printer. If you opt to create a service name, you need the socket number in Step 2b when you edit the `/etc/services` file. Table 8–1 lists the socket numbers for three printers made by Compaq and one made by Hewlett Packard.

**Table 8–1: 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 might 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 that you 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. These variables establish settings that are relevant to the serial port driver but are ignored by the network socket driver.

- b. Configure the services database. You must register a service name and `tcp` port number (socket number) in the `/etc/services` database file. Enter the socket number that you determined when you configured the printer in step 1 and associate it with a service name of your choice. For example, to configure the services database for a DEClaser 3500, you add the following line to the `/etc/services` file:

```
declaser3500    10001/tcp
```

The user-defined string `declaser3500` represents the service; it is the same string that you entered as the *servicename* in the `/etc/printcap` file in step 2a. Save the changes to the `/etc/services` file.

- c. Configure the remote hosts database. The *nodename* value that you specified as part of the `lp` variable value in the `/etc/printcap` file must be known by your local host's network management services; therefore, you must enter the *nodename* and its network address in the `/etc/hosts` database file. If you are running a BIND server for remote host names, you do not necessarily need to add the printer's node name to the `/etc/hosts` file, though if there is ever a problem with the BIND server, an entry in `/etc/hosts` is a useful fallback.

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.

## 8.2.5 Configuring PC Print Queues

If the Advanced Server for UNIX (ASU) is installed and running, you can configure client PC printer queues. ASU also offers features for configuring and managing print queues. See *Installation and Administration Guide* for more information on using ASU. You must have at least one printer configured in your `/etc/printcap` file before you can create a printer share queue for PC clients.

Invoke `printconfig` from the CDE Application Manager — Configuration window, or from the command line, as described in `printconfig(8)`. 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 is 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 might need to perform other ASU tasks to make the queue available to PC systems. Consult the ASU documentation 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 8–2.

**Table 8–2: lprsetup Options**

Command	Description
add	Adds a printer
modify	Modifies an existing printer's characteristics
delete	Removes an existing printer from your configuration
exit	Exits from the <code>lprsetup</code> program
view	Displays the current <code>/etc/printcap</code> file entry for the printer you are configuring
quit	Exits from the <code>lprsetup</code> program
help	Displays online help about the <code>lprsetup</code> 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. See `printcap(4)` for information on the supported symbols.

---

The following example shows how to use the `lprsetup` command to set up an LN03R printer on the local system. Some tables are truncated to shorten the example:

```
# /usr/sbin/lprsetup
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_l2cpi   hp61_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    lg104plus ln20_a4
DEClaser2200 (ln06) generic_text   ibmpro   la75    lg105plus ln40
DEClaser2250 (ln06r) generic_text_a4 la100    la84    lg108plus ln40_a4
DEClaser2300 (ln05ja) hp1120c      la120    la86    lg109plus 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  wpsosof
ep1050+            hp680c_a4    la380cb  lg08plus ln17     xf
escp_a4_10cpi      hp61         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/lplacct]? Return
Set spooler directory 'sd' [/usr/spool/lpd/lpd1] ? Return
Set printer error log file 'lf' [/usr/adm/lpd/lplerr] ? 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. See `printcap(4)` 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 mj 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/lplacct
br INT 9600
ct STR dev
fc INT 0177777
fs INT 03
if STR /usr/lbin/ln03rof
lf STR /usr/adm/lpd/lplerr
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 are inserted directly above the `printcap` entry in the `/etc/printcap` file.

Adding comments to `printcap` file for new printer, type '?' for help.  
Do you want to add comments to the `printcap` file [n] ? : **y**

```
# 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**

See `lprsetup(8)` 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 are directed to the Advanced Printing Software inbound gateway. The inbound gateway submits the job to an Advanced Printing Software spooler.
- `rp` – Specify the name of the Advanced Printing Software logical printer.

See the *System Administration and Operation Guide* 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

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. Set the protection and ownership of this file appropriately.

Ensure that the `/etc/inittab` file does not invoke the `getty` process on serial lines that have printers attached. If you use `printconfig`, this is done for you.

### 8.3.2 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)` 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. Several printers can share an accounting file. If you have such 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 management 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. See 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 require superuser (root) privileges.

---

#### 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 8–3.

**Table 8–3: lpc Command Arguments**

<b>lpc Argument</b>	<b>Description</b>
<code>help [argument]</code>	Prints a one-line description of the specified <code>lpc</code> command argument. If an <i>argument</i> variable is not specified, the list of arguments is displayed.
<code>? [argument]</code>	Same as the <code>help</code> argument.
<code>abort</code>	Terminates an active <code>lpd</code> daemon and then disables printing. This prevents the <code>lpr</code> or <code>lp</code> command from starting a new <code>lpd</code> daemon.
<code>clean</code>	Removes any temporary files, data files, and control files (for example, files that do not form a complete printer job) from the specified print spooling directory.
<code>disable</code>	Turns off the specified print spooling queue. This prevents the <code>lpr</code> or <code>lp</code> command from entering new jobs in the queue.
<code>down message...</code>	Turns off the specified print queue, disables printing, and enters the specified message in the printer status file. You do not need to quote the message because remaining arguments are treated the same as <code>echo</code> . You can use the <code>down</code> argument to take down a printer and inform users. If a printer is down, the <code>lpq</code> command indicates that the printer is down.
<code>enable</code>	Enables spooling for the specified printers. This enables the <code>lpr</code> or the <code>lp</code> command to enter print jobs in the spooling queue.
<code>exit</code>	Exits from <code>lpc</code> .
<code>quit</code>	Exits from <code>lpc</code> .
<code>restart</code>	Attempts to start a new <code>lpd</code> 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 <code>lpq</code> 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 <code>restart</code> argument.
<code>start</code>	Enables printing and starts a spooling daemon for the specified printer.
<code>status [printer]</code>	Displays the status of the specified printer daemon and queue. The <code>status</code> 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 <code>lpd</code> daemon. If a printer name is not supplied, information about all printer daemons and queues is displayed.
<code>stop</code>	Stops a spooling daemon after the current job is complete and disables printing.

**Table 8–3: lpc Command Arguments (cont.)**

<b>lpc Argument</b>	<b>Description</b>
<code>topq printer</code>	Puts print jobs in the queue in the specified order. You can specify the print jobs by also specifying a <i>request_ID</i> variable or a <i>username</i> variable.
<code>up</code>	Enables all printing and starts a new printer daemon. Cancels the <code>down</code> argument.

The following example shows that the `lpd` daemon is active on the printer named `tester` and there is one entry in the queue:

```
# /usr/sbin/lpc
lpc> status tester
tester:
    printer is on device '/dev/tty02' speed 9600
    queuing is enabled
    printing is enabled
    1 entry in spool area
lpc>
```

See `lpc(8)` for more information.

### 8.3.5 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 system 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**

---

User `adm` owns the `/var/adm/lpd` directory and it belongs to the `adm` group. User `adm` also owns printer accounting files, which have a protection mode of `644` and belong to the `system` group.

---

In the printer's `/etc/printcap` entry, the `af` parameter specifies the name of the accounting file. The accounting process uses this file to record the number of pages printed by each user on each printer. The name of the accounting file is unique for each printer on your system. Use the `pac` utility to display information in the printer accounting files. User `daemon` owns the accounting file, and it is a member of the `daemon` group. The correct file ownership is set automatically if you use the `printconfig` utility to specify the printer accounting file. The `af` parameter is not applicable for remote printer entries.

Accounting is accomplished through programs called print filters. The `printconfig` utility suggests a default print filter. 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, ensure that the file names are unique. An unlimited number of printers can share an accounting file but you cannot specify an accounting file for remote printers. The print daemon owns the accounting files. If you specify an accounting file, intermediate directories are automatically created as needed.

---

**Note**

---

Printer accounting does not work for PostScript files.

---

## 8.4 Reference Information

The following sections contain information about the information you need to configure a printer. They also provided information on the line printer daemon (`lpd`) and the print system files. If you use `printconfig` as described in Section 8.2.2, it creates the system files automatically.

You can also create and modify the files manually. If you create files manually, you must manually change the `/etc/printcap` file so the changes can take effect.

### 8.4.1 The /etc/printcap File

The `lpd` daemon uses information in the `/etc/printcap` database file to print requests. Each entry in the `/etc/printcap` file describes a printer. You specify printer characteristics using two-letter abbreviations called print symbols. The print symbols are described in this section and in `printcap(4)`. The `lprsetup` utility modifies the `/etc/printcap` file.

The 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, separating each field on the first line by using a vertical bar (|). The first line must end with a colon (:).

---

**Note**

---

A local printer entry in the `/etc/printcap` file should have the default printer reference name `lp0` so that print jobs can have a destination when printer reference names are not specified in print commands.

---

The remaining lines of each printer entry contain the descriptive symbols and values that define the printer's configuration. Symbols are two-character mnemonics and you can specify them by using an equal sign (=) and alphabetic characters or with a number sign (#) and a numeric value. Some symbol names have Boolean equivalents, which do not use parameters. You can specify the symbols on one line or on individual lines, but you must separate them with colons (:).

To make the `/etc/printcap` file easy to read, you can place a colon (:) at the beginning of a line and a backslash (\) at the end of a line to separate the symbols.

The `printcap(4)` 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 (`lpN`) is the other system-assigned synonym. If the default number is 1, the two assigned names are 1 and `lp1`. You can queue jobs to this printer using either of the following commands:

```
# lpr -P1
# lpr -Plp1
```

You 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 has a printer number of 0. This is recognized as your system's default printer and it has an additional name of `lp`.

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 `lprsetup.dat(4)`.

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, see 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/lpn` (for example: `lp1`, `lp2`, `lp3`), and the device special files for serial line printers are named `/dev/ttynn` (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 8-4.

**Table 8-4: Communication Ports and Printer Device Special Files**

Device Special File	Communication Type	Connector Label
<code>/dev/lp0</code>	parallel	printer, or <code>lp</code>
<code>/dev/tty00</code>	serial	COMM1 or 1
<code>/dev/tty01</code>	serial	COMM2 or 2

Note that if only one 9-pin serial connector is provided on a system, it is not always labeled as such. Some systems also use graphical icons instead of labels. See the owner's manual for the system for more information.

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.

### 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
- `tcp` for TCP/IP devices

### 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 is unique. All `printcap` file entries must specify a spooling directory, both local and remote. When the spooling directory is created with `printconfig`, intermediate directories are created as necessary.

Each printer must have its own spooling directory located under the `/usr/spool` directory. The spooling directory acts as a printer's spooling queue; it contains the files that are queued for printing on that printer. A printer spooling directory should have the same name as the printer reference name and is located on the system attached to the printer. The printer reference name is the name that you specify to print on a particular printer.

If you are using `lprsetup`, the utility supplies the default value `/usr/spool/lpdn`. The `n` variable specifies the printer number. For example, the default name of the spooling directory for a second line printer is `/usr/spool/lpd2`. 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 system 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 might 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 are removed or have 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 does 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 print files that users have submitted and contain printing instructions for these files. Each line in a control file begins with a key character that indicates what to do with the remainder of the line. The key characters and their meanings are described in detail in `lpd(8)`.

Data files, whose names begin with `df`, are also located in the spooling directory. Data files contain text formatted 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 users submit jobs to a printer named `milhaus`, 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. Set the spooling directory permission mode to `775`. Set the directory's group and ownership mode to the name `daemon`. For example:

```

# cd /var/spool/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/sbin` directory. Printing cannot take place unless the `lpd` daemon is running. The `lpd` daemon has many functions:

- Handles printer spooling conjunction with the `lpr` and `lprm` commands. Spooling is the mechanism by which a file is placed in a queue until the printer can print the file.
- Scans the `/etc/printcap` file to determine printer characteristics.
- Uses specific print filters for print requests. Print filters translate an input format into a printer-specific output format.
- After a system reboot, prints any files that were not printed when the system stopped operating.

When you use the `lpr` command, it copies files into the spooling directory and activates the `lpd` daemon. Requests are printed in the order in which they enter the queue. A copy of the file remains in the queue until the printer is ready to print it; then the `lpd` daemon removes the file from the spooling queue after it sends the job to the printer.

After you install and boot your system, the `lpd` daemon is usually started by the `/sbin/init.d/lpd` utility. You can start and stop the `lpd` daemon with the following commands:

```
/usr/sbin/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 -- lpg: cannot open printer description file
Apr 15 16:36:46 cymro lpd[1130]: ERROR -- exiting ...
#
```

Monitor log files regularly for errors and deleted (or archive) the logs to prevent them from filling up the available disk space. You can establish a regular clean up procedure using the `cron` utility. You 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)` 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. Local printers can share an error log file, 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, use the `ln03rof` filter to print files with the LN03R printer,

Input filters process job data being sent to the printer. Output filters process banner page data generated by the `lpd`. You can specify many filters as either input or output filters, and each operates differently depending on whether they are called as input or output filters. These filters are specified in the `/etc/printcap` file as follows:

```
if=/usr/lbin/ln03rof  
of=/usr/lbin/ln03rof
```

Input filters are also responsible for printer accounting, which enables you to keep a record of printer usage for text jobs (not for PostScript printing). For information on configuring printer accounting refer to Chapter 10.

See `lpd(8)` for more information on using filter capabilities.

The `lprsetup.dat(4)` lists the available print filters located in the `/usr/lbin` directory. For printers not listed in the reference page, see the printer documentation or contact the printer manufacturer for filter information.

#### 8.4.6 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 does not work with PostScript-only printers.

PCF files are text files. You can use any text editor to modify an existing file to customize printing behavior or create a new file for an unsupported



printer. PCF files provided in Tru64 UNIX are replaced during an installation update, so you should take care that you preserve any customizations in backups. Using a file name prefix for new or modified PCF files prevents potential file name conflicts. For example, copy file names before customizing as follows:

```
# cp 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.

### 8.4.7 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 might 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. Clear all bits by using `fc#0177777` before you call the `fs` symbol. To set or clear any groups of bits, specify the octal sum of the combined bits for the number of flag bits.

The following is an example of flag bit specifications:

```
fc#0177777  
fs#0141
```

In this example, `fc#0177777` clears all bits and the `fs` symbol is set to `0141` specifying the `OPOST`, `ONLRET`, and `OFILL` flag bits.

Table 8-5 lists each flag bit name, its octal value, and its description.

**Table 8–5: Flag Bits**

<b>Flag</b>	<b>Octal Value</b>	<b>Description</b>
OPOST	0000001	Enable output processing
ONLCR	0000002	Map NL to CR-NL
OLCUC	0000004	Map lower case to upper case
OCRNL	0000010	Map CR to NL
ONOCR	0000020	No CR output at column 0
ONLRET	0000040	NL performs CR function
OFILL	0000100	Use fill characters for delay
OFDEL	0000200	Fill is DEL, else NUL
NLDLY	0001400	Newline delay
NL0	0000000	
NL1	0000400	
NL2	0001000	
NL3	0001400	
TABDLY	0006000	Horizontal tab delay
TAB0	0000000	
TAB1	0002000	
TAB2	0004000	
TAB4	0006000	
CRDLY	0030000	Carriage Return delay
CR0	0000000	
CR1	0010000	
CR2	0020000	
CR3	0030000	
FFDLY	0040000	Form feed delay
FF0	0000000	
FF1	0040000	
BSDLY	0100000	Backspace delay
BS0	0000000	
BS1	0100000	
OXTABS	1000000	Expand tabs to spaces

See `tty(7)` for detailed information on flag bits.

## 8.4.8 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.

See `tty(7)` 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 8–6 lists a description of each mode bit.

**Table 8–6: Mode Bits**

Mode	Octal Value	Description
ECHOKE	0000001	Echoes KILL by erasing the line
ECHOE	0000002	Visually erase characters
ECHOK	0000004	Echoes NL after KILL
ECHO	0000010	Enable echoing
ECHONL	0000020	Echoes NL even if ECHO is off
ECHOPRT	0000040	Echo erased chars between and /
ECHOCTL	0000100	Echo control characters as ^(char)
ISIG	0000200	Enable special chars INTR, QUIT and SUSP
ICANON	0000400	Enable canonical input
ALTWERASE	0001000	Use alternate word erase algorithm
IEXTEN	0002000	Enable FLUSHO and LNEXT
XCASE	0040000	Canonical upper/lower presentation

## 8.4.9 Remote Printer Characteristics

For users to access a printer from a remote system, both the local system and the remote system require printer information in their `/etc/printcap`

files. On the local system attached to the printer, security is controlled by the entries in `/etc/hosts.lpd` or `/etc/hosts.equiv`.

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 system that is attached to the printer. If you define the value as false, remote users can access the local printer if the local printer is listed in the `/etc/hosts` file. Refer to Section 8.4.1 for an example of an `/etc/printcap` file.

On the remote system, you must specify the `rm`, `rp`, `lp`, and `sd` symbols.

The `rm` symbol specifies the name of the system attached to the printer. For example:

```
rm=deccom
```

The `rp` symbol specifies the printer spool name on the remote system. For example:

```
rp=ln03lab
```

For remote printers, specify the `lp` symbol without a value:

```
lp=
```

The `sd` symbol specifies the spooling directory. For example:

```
sd=/usr/spool/lpd
```

#### 8.4.10 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 `p1` 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:

```
p1#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.

### 8.4.11 Generic Print Filters

Tru64 UNIX provides print filters for generic and for local-language use.

The `pcfof` filter is a generic print filter that you can use 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. See `pcfof(8)` for a list of options.

The `wpsof` filter uses settings in a printer customization file (PCF) to find the font glyphs for local language characters and then embeds the font data in the PostScript file. The filter uses PostScript outline fonts, if installed on the local system, or bitmap fonts, which the filter obtains through a font server. This feature supports print jobs that contain multinational character sets and you do not need to send such jobs to special printers. See `wpsof(8)` for a list of options. See the *Writing Software for the International Market* and the `i18n_printing(8)` for additional information on local-language printer support.

## 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 might cause reliability problems. You might need to configure network cards in the printer in order to identify the router. See printer documentation for information.
- Printing non-PostScript files with some PostScript and non-PostScript filters might yield unexpected results. Table 8-7 lists the filters which might cause problems.

**Table 8-7: Non-PostScript and PostScript Filters**

Filter Name	Filter Type
lpf	Non-PostScript
la75of	Non-PostScript
la324of	Non-PostScript
lqf	Non-PostScript
hplaserof	PostScript

To provide expected behavior with older printers, these non-PostScript filters maintain a dependence on the serial port driver to automatically supply carriage returns after line feeds when you specify the (octal) 020 bit to the `fs` control variable in the `/etc/printcap` file.

Because this control bit is not interpreted by the network socket driver, the formatting behavior supplied by the serial port driver is absent. Therefore, non-PostScript files that are not preformatted for the printer might not print as in serial-port-connected configurations. In particular, this might affect ASCII text files that do not contain embedded carriage-returns.

- Most printers using the `lpf`, `la75of`, `la324of`, and `lqf` non-PostScript filters do not provide network interface card support. However, users who use serial-and-parallel-port to network-port converters might still have problems. (An example is the Compaq RapidPrint network interface box, which allows printers to behave like TCP/IP printers.)
- The `hplaser4ps` PostScript filter works for PostScript files and for preformatted non-Postscript files (like PCL files), but it might produce unexpected results for files that are not preformatted (such as ASCII text without embedded carriage-returns).
- Some filters designed to work with character-set printing (such as ASCII) might 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. See `lpr(1)` for more information on how to invoke these features.

The `lpctest` command writes a ripple test pattern to the standard output, or you can direct the output to a printer. A pattern that contains all 96 printable ASCII characters in each column is printed using 96 lines. In the pattern, each printed character is displaced rightward one character column on each successive line. This test is also useful for ascertaining the number of lines per page and the default page parameters. You can use

the ripple test pattern to test printers, terminals, and drive terminal ports during debugging.

The `lptest` command has the following syntax:

```
/usr/sbin/lptest [length [count]]
```

Use the `lptest` command if you need quick output of random data. For example:

```
# /usr/sbin/lptest | lpr -P3r44
```

See `lptest(8)` for more information.

If a problem occurs on an existing printer or when adding a printer to a system, diagnose the problem as follows:

- See the error log files specified in Section 8.4.4.
- 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. 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, use 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/sbin/lpd` causes the daemon to log requests from the network. This flag is useful for debugging problems with remote printers. See `lpd(8)` for information.

- 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 (see 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 exist and that ownership and access are correct (see 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. See the *Network Administration: Connections* guide for information on network troubleshooting.



---

## Administering the Archiving Services

One of the more common tasks of a system administrator is helping users recover lost or corrupted files. To perform that task effectively, you must set up procedures for backing up files at frequent and regular intervals. This chapter describes how you use resident commands and utilities to back up (archive) and restore files and directories.

Design and implement a disaster recovery plan that describes how you intend to restore your entire operating system and user files to normal operations in the event of a catastrophic failure. This chapter does not describe the disaster recovery process, because it is often very specific to site operations and business requirements. However, backup operations are an important component of such a plan.

The following topics are included in this chapter:

- 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 might 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 might 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 might 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, 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 during a backup. If a user accesses a file during a backup is in process, the backup might not record changes in the file. To ensure a completely accurate back up of a UFS file system, you might 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), 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. (Using LSM requires spare disk capacity and might be unsuitable for small systems with few disks.)
- If you want to back up and restore a root volume to a different system, consider using configuration cloning. This feature is described in the *Installation Guide — Advanced Topics*. Configuration cloning enables you to recreate a customized operating system on another processor in the event of a disaster, or to recreate an environment on one or more systems.
- This chapter describes only those backup and archiving utilities that are provided in the base operating system when installed.

The *Associated Products* CD-ROM might include additional backup applications (which might require additional licenses). Refer to the *Installation Guide* for information. 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 by using the `pax`, `tar`, and `cpio` commands or the associated graphical user interface, `dxarchiver`
- Performing a full UFS backup using the `dump` utility
- Performing an incremental backup
- Performing a remote backup
- Using backup tools
- Restoring data
  - Restoring files from small archives
  - Restoring a file system from a dump
  - Restoring a dumped file system on a new partition
  - Restoring files
  - Restoring files interactively
  - Performing remote restorations
  - Restoring standalone systems from bootable tape

## 9.2 Backing Up Data and System Files

For basic backup, you can use the `dump` and `restore` commands. See `dump(8)` for full details of all command options that are supported. The operating system also provides graphical and command line tools for archiving and for creating a bootable tape of the standalone system (SAS).

Prevention of data loss is an important part of any backup and recovery strategy. There are many tools for system monitoring that you can configure to help prevent situations that might result in data loss. For example, some systems support environmental monitoring, and there are tools to test and exercise peripherals. There are also the event and error logging systems that you can configure to monitor the system for priority events such as a backup failure. See Chapter 13 for information on using EVM (event management) to set up the event reporting strategy for your system and site. You can also use EVM to report on the success of your backups, ensuring that you do not miss a scheduled backup event.

It is important that all the files on your system, user files and system files, are protected from loss. Back up your entire system, including the system software. Many system files are static; that is, after you install them they remain unchanged. Therefore, you do not need to back up system files as frequently as data files. Incremental backups are also possible, and you might consider implementing them if your data changes significantly in a short period.

Each file system backup is a single process. To ease the backup process, organize your file systems so that dynamic files are on file systems that you back up regularly and static (system or program) files are on file systems that you back up occasionally. You might find that you have dynamic files on file systems that you back up occasionally. If this happens and you need to back them up regularly, just prior to performing a backup, copy the frequently changing files to systems that you back up regularly. This allows you to back up those files without backing up an entire file system. You can also write shell scripts to automate these tasks and use the `cron` command to automate the schedule. See `cron(8)` for more information on scheduling tasks.

### 9.3 Choosing a Backup Schedule

To decide how often to back up each file system, consider the balance between the potential loss of user time and data and the time it takes you to perform backups. Ask yourself the question, “How much information can I afford to lose?” The answer to this question helps you determine your minimum backup interval. On most systems the backup interval is daily, but you can choose any other interval.

It is not necessary to back up all the files in a file system at each backup. Back up only those files that changed since the previous backup; this is called an incremental backup. Using the `dump` and `restore` commands, you can perform up to nine levels of incremental backups. For example, while a level 0 dump backs up an entire file system, a level 1 dump backs up only those files changed since the last level 0 dump, and a level 7 dump backs up only those files changed since the last lower level dump.

To integrate incremental backups into your file backup schedule, you need to balance the time and tape space required for backup against the amount of time it could take you to restore the system in the event of a system failure. For example, you could schedule backup levels following the 10-day sequence:

```
[0 1 2 3 4 5 6 7 8 9]
```

On the first day you save an entire file system (level 0). On the second day you save changes since the first backup and so on until the eleventh day when you restart the sequence. This makes the amount of time spent and data saved on each backup relatively small each day except the first; however, if a system failure on the tenth day requires that you restore the entire system, you must restore all ten tapes.

Most systems follow some variant of the common Tower of Hanoi backup schedule. Once monthly you make a level 0 dump to tape of all the file systems that you backup regularly. Once weekly, you make a level 1 dump to start a daily sequence of:

[...3 2 5 4 7 6 9 8 9 9 ...]

If you do backups only once a day on the weekdays, you end up with a monthly backup schedule as follows:

[0 1 3 2 5 4 1 3 2 5 4 ...]

This schedule, although slightly complex, requires that you restore at most four tapes at any point in the month if a system failure corrupts files. Of course, doing a level 0 dump daily requires that you restore at most one tape at any point, but requires a large amount of time and tape storage for each backup. On most days in the Tower of Hanoi schedule, you require very little time and tape storage space for the backup.

## 9.4 Backup Methods

Depending on your needs and your local system configuration, there are several options for backing up data, as follows:

- You can run the following command-line interfaces from a terminal:
  - `dump`, `rdump`, `restore`, and `rrestore`
  - `tar`, `pax`, and `cpio`

Use these to create quick file archives or to create scripts that you run with the `cron` scheduler.

- The bootable tape feature, `bttape`, is a SysMan Menu application that you can invoke from the command line, the SysMan Menu, or from CDE. Depending on how you invoke it, it either runs the command-line interface or a graphical interface that is appropriate to your windowing environment. (See Chapter 1 for more information.) The commands are `btcreate` and `btextract`.

Use the bootable tape feature to create a bootable tape for recovery and to back up critical system data and customized system files. This feature also enables you to use any terminal and a number of windowing environments, and is therefore recommended for remote operations.

- From the CDE folder Application Manager – System Admin, open the Storage Management folder and click on the Bootable Tape icon. This action invokes the graphical 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.

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 might need to prepare for a backup. Also included is a list of utilities that can assist you in preparing for a backup, and a list of prerequisite tasks.

Chapter 6 contains information on the UFS file system. 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` command in Chapter 3 for information on scheduling regular backups. The following documentation contains other information that you might 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 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 your options, and defines what data you can enter 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 `restore` commands create or use the following files:
  - `/etc/dumpdates` – Contains a list of file systems that were backed up, the date that each file system was backed up, and the backup level
  - `/tmp/rstdir*` – Lists directories stored on the default tape
  - `/tmp/rstmode*` – Records the owner, permission mode, and timestamps for stored directories
  - `./restoresymtab` – Holds information required during incremental `restore` or `rrestore` operations
- The bootable tape feature creates or uses the following files:
  - `/var/adm/btcreate.log` – Provides a log of the `btcreate` process
  - `/usr/lib/sabt/sbin/custom_install.sh` – Specifies which files are added to the miniroot
  - `/usr/lib/sabt/etc/addlist` – A data file that specifies which files and directories are added to the miniroot file system that is created on the bootable tape
  - `/usr/lib/sabt/etc/fslist` – A data file that specifies which file systems are backed up
  - `/usr/run/bttape.pid` – A lock file that prevents multiple instances of the `btcreate` utility

### 9.5.2 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 option 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 alerts 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` GUI, which you can invoke from CDE Application Manager – Configuration folder, or from the SysMan Menu. This interface provides size information in megabytes, bytes, and blocks. (The `disklabel` command provides a command-line disk configuration interface).

### 9.5.3 Prerequisite Tasks

The following prerequisite tasks apply to all the backup methods:

- Become familiar with using the interfaces and the sources of information about the commands. Ensure that this information is available to you when the system is down. You must often perform recovery operations in single-user mode, and reference pages might not be available.
- Ensure that all the required products or utilities are installed and configured (if necessary). The simplest way to do this is to 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 you want to back up. For example, you can use the following commands:

```
# df /usr
```

```
Filesystem          512-blocks    Used   Available Capacity  Mounted on
/devices/disk/dsk0g  1498886      688192    660804    52%    /usr
```



```
# du -s -x /usr/users
1835      /usr/users
```

You can 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 might need some temporary scratch disk space if assembling different directories into a single volume before archiving (although you can do this 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 prompts you for the following file names. (You can accept the default or provide another file name):
  - `/usr/lib/sabt/etc/fslist` – A data file that specifies which files and directories are added (appended) to the miniroot
  - `/usr/lib/sabt/etc/addlist` – A data file that specifies which file systems are backed up

The Archiver requires the following files:

- One or more source files or directories. In CDE, directories are identified as folders, and you can drag and drop them into the Archiver window from File View windows instead of 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`. (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
tape0c	/dev/tape/tape0c	Default density rewind tape (with compression)
tape0_d0	/dev/ntape/tape0_d0	Nonrewind tape device 0. The _d0 suffix specifies the density

Device names are located in the /dev directory under the /disk, /rdisk, /tape, or /ntape subdirectories. 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. See `tz(7)` for information on tape density options, and how you select them by specifying different device names.

- Full backups might 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. If you use the `dump` command to back up files in a file system, an inode is attached to each file. If the system or user activity changes a file after the inode data is recorded, but before the file is backed up, it might corrupt the backup.

To shut down the system, unmount a file system, and check the integrity of a file system:

1. Shut down the system by using the SysMan Menu General Tasks option, or with the `/usr/sbin/shutdown` command. For example, to shut down the system in 5 minutes and give users periodic warning messages, enter:

```
# /usr/sbin/shutdown +5 'System going down to perform backups'
```

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
```

The root file system remains mounted.

3. Use the `fsck` command to ensure the integrity of the file system.

For example, use the following command to check a file system on the `c` partition (the whole disk):

```
# /sbin/fsck -o /dev/disk/dsk0c
```

## 9.6 Using the dump Command

The `dump` command copies all designated file systems or individual files and directories changed after a specified date to a file, pipe, magnetic tape, disk, or diskette. 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

Set up a schedule for performing a full backup of each file system on your entire system, including all the system software. A conservative schedule for full system backups is to do one with each normal level 0 dump (using Tower of Hanoi, once a month), but you can set any schedule you like within the reliability of your storage media, which is about two years for magnetic tapes. To back up your file system, use the `dump` command. See `dump(8)` for a description of the command options that you use to specify the characteristics of the tape device, such as block size, tape storage density, and tape length. Specify the file system with a full pathname when you use the `dump` command. The `dump` command can back up only a single file system at a time, but there might be several `dump` processes simultaneously writing files to different tape devices.

The following list describes the most commonly used options to the `dump` command:

<code>-integer</code>	Specifies the dump level as an integer (0-9). A dump level of 0 causes a full dump of the specified file system. All other dump levels cause an incremental backup. That is, only files that have changed since the last dump of a lower dump level are backed
-----------------------	--

up. The `/etc/dumpdates` file contains a record of when the `dump` command was used on each file system at each dump level. The `-u` option to the `dump` command updates the `dumpdates` file.

- `-f dump_file` Writes the dump to the device specified by `dump_file` instead of to the default device, `/dev/tape/tape0_d0`. If you specify the `dump_file` as a dash (`-`), the `dump` command writes to the standard output.
- `-u` Updates the `/etc/dumpdates` file with the time of the dump and the dump level for the file system in the backup. You use this file during incremental dumps (by using the dump level option) to determine which files have changed since a particular dump level. You can edit the `/etc/dumpdates` file to change any record or fields, if necessary. See `dump(8)`, which describes the format of this file.

To back up your entire file system to the default backup device, use the `dump` command for each file system on your machine. The `dump -0u` command option causes a level 0 dump and updates the `/etc/dumpdates` file with the time and date of the backup for each file system. This creates an initial point on which to base all future incremental backups until the next full or level 0 dump. Each file system must be backed up individually.

For example, if you want to perform a level 0 dump of the root, `/usr`, and `/projects` file system partitions, follow these steps:

1. To back up the root file system, 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 might require additional media management to cross-reference dump files with tapes, especially when a single dump file spans media. Exercise care when labeling this type of backup media.

## 9.6.2 Performing an Incremental Backup

Set up a routine as part of your backup schedule to make it easier to remember which backup to do each day. Include a mechanism for logging your backups and their dump level and for listing the tapes on which they are made. Because of the chance of system corruption, do not keep this information on the local computer system.

After you establish a system for making incremental backups, the procedure is simple. Assume you use the following backup schedule to do a daily backup of `/usr`:

```
0 1 9 9 9 1 9 9 9 9 ...
```

On Monday, perform a level 0 dump:

```
# dump -0u /usr
```

On Tuesday, perform a level 1 dump:

```
# dump -1u /usr
```

The level 1 dump backs up all the files that changed since Monday. On Wednesday through Friday 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. See `dump(8)` for a description of the options to the `rdump` command.

The `rdump` command updates the `/etc/dumpdates` file on the local machine in the same way as does the `dump` command. The `rdump` command starts a remote server, `/usr/sbin/rmt`, on the remote machine to access the storage medium. (This server process is transparent. See `rmt(8)` for more information.)

To back up the `/projects` file system from `bhost1` onto a tape drive on `bhost2` with the attached backup device `/dev/rmt0h`, enter the following command from `bhost1`. The name of `bhost1` must be defined in the `/.rhosts` file of `bhost2` to allow access.

```
# rdump -0uf bhost2:/dev/tape/tape0_d0 /projects
```

## 9.6.4 Using Backup Scripts

You can automate the backup process by using shell scripts. The `cron` daemon can execute these shell scripts late in the evening when there is less chance of the `dump` commands making errors 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. After the shell procedures are finished, remove the backup tape and archive it.

Backup shell scripts are best used when the dump is small enough to fit on a single tape. You must specify the no-rewind device and the `-N` option to the `dump` command to inhibit the tape from automatically going off line when each dump is completed. After the `dump` command reaches the end of the tape, it takes the tape off line and you must replace the tape.

## 9.7 Restoring Data

Occasionally, you need to retrieve files from your backup tapes, and possibly need to restore entire file systems at some time. If you have set up a good backup procedure, then restoring files or full file systems is a simple task.

If a serious problem occurs, you might have to restore your entire system. Before restoring, determine what caused the problem with the system.

After determining the cause of the problem, reinstall your system from the initial boot tapes. The installation instructions that came with your system explain this procedure.

After your system is up and running, restore the system to the state it was in just prior to the system crash. If you are using AdvFS, use the `vrestore` command. Refer to *AdvFS Administration* for information on restoring the AdvFS file system. If you used the `vdump` command to back up a UFS file system, you can also use the `vrestore` command to recover it. However, if you used the `dump` command you must use the `restore` command to recover files. Because the `dump` command saves only a single file system at a time, you must execute the `restore` command for each file system you want to restore. See `restore(8)` for information on the command syntax.

### 9.7.1 Restoring a File System

This section describes a general procedure for restoring a file system, such as after a disk failure or other loss of data. To restore individual files, go to Section 9.7.2.

To restore a file system, create a new file system and restore the files from the dump files by using the following commands:

- `newfs` – Creates a new UFS file system. See `newfs(8)`.
- `mount` – Mounts the file system, making it available for general use. See `mount(8)`.
- `cd` – Changes your current working directory. See `cd(1)`.
- `restore` – Restores archived files from a backup media to a disk. See `restore(8)`.

Refer to the *AdvFS Administration* guide for information on restoring an AdvFS file system.

If the disk does not have a label, write the label by using the `disklabel` command before you create the new file system. See `disklabel(8)`.

Writing a label with customized partition table settings might affect the entire disk. Use the following command to write the default disk partition table:

```
# /sbin/disklabel -rw dsk1
```

Invoke the editing option of the `disklabel` command to use the customized partition table settings. Refer to Chapter 6 for more information. You can also use the `diskconfig` Disk Configuration interface. See `diskconfig(8)`.

The following example shows the commands you use to restore a file system called `/usr/projects` from a tape:

```
# disklabel -rw dsk1
# newfs /dev/rdisk/dsk1c
# mount /dev/rdisk/dsk1c /usr/projects
# cd /usr/projects
# restore -Yrf /dev/tape/tape0_d0
```

## 9.7.2 Restoring Files Manually

If users lose data files, they ask their system administrator to restore those files. Users might also ask you to restore an earlier version of a file. Whatever the reason for a file restoration, you must determine which tape contains the correct version of the file. Enquire when the file was lost and when it was last modified, you can use your backup log to determine which tape contains the most recent version of the wanted file.

Use the `-t` option with the `restore` command to determine whether a file is on the selected tape. The `-t` option creates a list of files and directories on the tape. For example, to list the contents of the `working` subdirectory of the `/usr` file system on a particular backup tape, load the tape and enter:

```
# restore -t ./working
```

To create a list of the entire contents of a backup tape, load the backup tape and enter:

```
# restore -t
```

Make a listing of each backup tape after you create it. This verifies a successful backup and gives you a place to look up what files are on the tape.

After determining the location of the file, create a new directory for the file. If you restore the file into an existing directory and the file already exists, the restored file overwrites the existing file.

For example, to restore the `working/old.file` file from a `/usr` file system backup tape into your current directory, load the backup tape and enter:

```
# restore -x ./working/old.file
```

To restore the entire contents of the `working` subdirectory from the same tape, enter:

```
# restore -x ./working
```

If your dump media contains multiple dump images, you need to know the sequence of the dump images in order to restore a file from one of the images. To examine the contents of the first dump image on the media, load the tape and enter:

```
# restore -ts 1
```



The `-s` option followed by the number 1 specifies the first dump image.

For example, to restore the `working/old.file` file from a `/usr` file system, which is the third dump image on the backup tape into your current directory, load the backup tape and enter:

```
# restore -xs 3 ./working/old.file
```

### 9.7.3 Restoring Files Interactively

To ease the task of restoring multiple files, use the `-i` option to the `restore` command. This option starts an interactive restore session. The interactive mode has commands similar to shell commands.

To begin an interactive restore session, enter:

```
# restore -i
```

The system responds with the following prompt:

```
restore >
```

The following command-line options are available in the interactive restore mode:

- |                                      |   |
|--------------------------------------|---|
| <code>ls [ <i>directory</i> ]</code> | Lists files in the current or specified directory. Directory entries end with a slash (/). Entries that are marked for reading begin with an asterisk (*).  |
| <code>cd [ <i>directory</i> ]</code> | Changes the current directory to the directory specified by the <i>directory</i> argument.  |
| <code>pwd</code>                     | Lists the pathname of the current directory.  |
| <code>add [ <i>files</i> ]</code>    | Adds the files in the current directory or the files specified by the <i>files</i> argument to the list of files recovered from the tape. Files are marked with an asterisk (*) if they are identified as “to be read” by the <code>add</code> command. You see this asterisk when you use the <code>ls</code> command to list files. |
| <code>delete [ <i>files</i> ]</code> | Deletes all the files in the current directory or the files specified by the <i>files</i> argument from the list of files recovered from the tape.  |
| <code>extract</code>                 | Restores from the tape the files that are marked “to be read” into the current working directory. The <code>extract</code> 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.

<code>setmodes</code>	Sets owner, access modes, and file creation times for all directories added to the files-to-read list; no files are recovered from the tape. Use this command to clean up files after a <code>restore</code> command is prematurely aborted.
<code>verbose</code>	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 <code>-v</code> command line option to the <code>restore</code> command.
<code>help</code>	Lists a summary of the interactive commands.
<code>?</code>	Lists a summary of the interactive commands.
<code>what</code>	Lists the tape header information.
<code>quit</code>	Quits the interactive restore session.
<code>xit</code>	Exits from the interactive restore session. The <code>xit</code> command is the same as the <code>quit</code> command.

To interactively restore the `./working/file1` and `./working/file2` files from a backup tape, load the tape and enter:

```
# restore -i
```

After you switch to interactive mode, follow these steps to add the files to the list of files that you want to extract:

1. Change to the working directory:  

```
restore > cd working
```
2. 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 (.) current directory. For this example, reply with n.

```
set owner/mode for '.'? [yn] n
```

6. After 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

You use the `rrestore` command to restore files to local directories from a remote tape device. The `rrestore` command requires the `-f` option to specify the machine name and its backup device. See `rmt(8)` for more information and Section 9.7 for a description of the options to the `rrestore` command.

You must specify the name of the remote system where the backup device is attached, and the name of the backup device on that remote system in the format `system:device`

To restore the `./working/file1` file onto the local directory on `system1` from a backup tape mounted on `system2` where the backup device `/dev/rmt0h` is attached, enter the following command from `system1`. The

name `system1` must be in the `/.rhosts` file of `system2` to allow access from `system1` to `system2`.

```
# rrestore -xf system2:/dev/tape/tape0_d0 ./working/file1
```

The `rrestore` command starts a remote server, `/usr/sbin/rmt`, on the remote system to access the storage medium.

### 9.7.5 Restoring or Duplicating a System (Root) Disk

In previous versions of the operating system, device names were assigned based on the physical location of the drive, according to the SCSI bus target. In Version 5.0 and higher, device names are assigned logically and stored in a database. They have no relationship to the bus address of the device. The device database must be recovered and possibly updated to successfully restore the root file system or if you want to move the root disk to a disk with larger capacity. You might also need to install devices (such as a tape device) to the device database when you restore the device from tape backup media.

After you reboot the system during the restoration, you might see the following message:

```
Unable to save existing hardware configuration.  
New configuration will be used
```

This message indicates that the device database is not recoverable and you must restore it.

The following procedure is a generic method for recovering or duplicating (cloning) a root disk. It covers the following possible scenarios:

- The disk and the root partition are not damaged but you want to replace it with a different disk, possibly a different model with larger capacity.
- The disk drive on which the root partition is located is damaged and you must:
  - Install a new disk drive, possibly of a different type and capacity
  - Choose an alternate disk drive that is already installed and available for use
- The root (`/`), and possibly the `/usr` or `/var` file systems are corrupt, but the disk drive on which they are located is fully functional.

---

#### Note

---

This procedure does not specifically address recovery methods from network backups and it does not address recovery of an AdvFS file system. See *AdvFS Administration*.

---

Depending on your knowledge of your system, you might not need to read all the following sections:

- Section 9.7.5.1 – Describes how you prepare for a recovery
- Section 9.7.5.2 – Describes the requirements for recovery
- Section 9.7.5.3 – Describes the recovery (or duplication) procedure
- Section 9.7.5.3 – Describes alternate procedures

### 9.7.5.1 Preparing for Recovery or Duplication

Depending on how your system is set up, and your level of system knowledge, you might need the following:

- A replacement root disk drive. This procedure assumes that if the original root disk is unusable, you have either installed a new replacement disk, or you have decided to use an alternate disk that is already installed in your system. Install the drive as described in the owner's manual for the drive. The operating system automatically detects the drive.

There are steps in the procedure that assist you in identifying the new or alternate drive.

- Firmware update.

Verify that your system's firmware is current. You can obtain information and download kits from the *Firmware Updates* web page at the following URL:

<http://ftp.digital.com/pub/DEC/Alpha/firmware>

(You can also go to the Compaq Web site at <http://www.compaq.com> and select the Support option to search for information.)

- Information about console commands.

You use Alpha System Reference Manual (SRM) console commands at the system console prompt (>>>) to perform some tasks. These commands are documented in the owner's manual for your AlphaServer system.

If you cannot find the printed document, it is usually shipped as a printable file on a CD-ROM supplied with the system. If the CD-ROM is unavailable, you can often find the documentation at the *Alpha Systems Technology* web page at the following URL:

<http://www.compaq.com/alphaserver/technology/index.html>

- The procedure instructions are typical for newer processors. If your system is older, you must consult the owner's manual and *Installation Guide* for your version of the operating system to obtain the actual commands and procedures.

The status of your system must be as described in Table 9–1:

**Table 9–1: Recovery Preparation**

Requirement	Description
A Full and Recent Backup	You need a full backup of all operating system file sets that are on the root volume. This might include root (/), /usr, and /var.
System Configuration	This procedure applies to all configurations where there is a single disk drive used for the root partition which might also contain the /usr and /var file systems. You need a functional disk drive to contain the restored root volume. This disk must have a minimum storage space as defined in the operating system limits for the restored release. The restore device (typically a tape drive) must be local and not a remote backup device.
Logical Storage Manager	If you are using the Logical Storage manager (LSM), refer to the <i>Logical Storage Manager</i> guide for information on recovering the root volume.
User Interface	This procedure requires a console login.
Impact on System Availability	Except on clustered systems, loss of the root disk invariably involves one or more shutdowns and reboots of the system. This procedure is intended to help you restore full operation as quickly as possible. The time required for duplicating or recovering a disk depends on the disk size.
Privileges	You must be a root user with physical access to the system's storage array and backup devices

### 9.7.5.2 Determining the Restoration Requirements

You might need the following resources to complete the restoration of your root disk. If are very familiar with your system's configuration, or if you have a recovery plan which records all the information you need to perform a recovery, you do not need to read this section. You might need the following items:

- Distribution Media for the Operating System.

You use the installation shell to restore the root disk. The installation shell is a compact version of the operating system from which you can execute commands, such as `mount`. The shell is packaged with the operating system software as part of the distribution kit.

Your local site might also provide a Remote Installation Service (RIS) server from which you can boot your system across the network. If RIS services are available in place of CD-ROM media, follow your site-specific procedures and consult the *Installation Guide*.

- CD-ROM Drive Name or Network Device Name.

To restore the root volume, you boot your system from a CD-ROM drive or a network device. You might need to refer to the owner's manual for your system to find the correct commands. Typically, you determine the CD-ROM device name at the console prompt as follows:

```
>>> show device | grep -E 'RR|CD'
dka400.4.0.5.0 DKA400 RRD47 1206
```

You typically determine the network device name as follows:

```
>>> show config | more
```

After you enter the preceding command, the complete system configuration is displayed one page at a time. Scroll down to the section headed `Slot Option` and locate the network device. Network devices are typically named `ew*` or `ei*`, where `*` is a letter. For example:

```
11 DE500-BA Network Con ewa0.0.0.0.11.0 08-00-99-1Z-67-BB
```

For information on booting your system from a RIS server, refer to the *Installation Guide — Advanced Topics* guide. For a full discussion of the System Reference Manual (SRM) console device naming conventions, see the owner's information for your system.

- Boot Device Name

Determine the default boot device name as follows:

```
>>> show bootdef_dev
bootdef_dev dka0.0.0.5.0
```

In this example the default boot device is `dka0`.

---

**Note**

---

If you are using Fibre Channel the name of the boot device is as you defined it during configuration of the storage devices.

---

If the current root device is usable and you are restoring to the same device, you use the device name later in the restore procedure. If you intend to install a new disk or use an alternate, you must specify the name of the disk. You determine the alternate by translating its `b/t/l` into the boot device name during the restore procedure.

- Backup Media and the Restore Device

Depending on what file systems were on your original root disk, you might need full and current backup tapes for the root (`/`), `/usr` and `/var` file systems.

In cases where you are duplicating (cloning) a disk, such as to increase the disk space available by using a disk with larger capacity, you can opt to back up directly from the source disk to the target disk.

- **Disk Label for the Target Disk**

If the original drive is usable, you can choose to restore the root file system to the same drive. If the drive is damaged, you must select an alternate drive or install a new drive. The alternate or new drive must have enough storage capacity to hold the restored file systems and it must be partitioned to hold the restored file systems.

If the original root drive held custom partitions, restore the custom disk label or at least plan to select partitions that can adequately contain the restored file systems (and possibly allow for future expansion of those file systems). Depending on what data is stored on the original disk, you might need to plan for the following partitions:

- The 256MB a partition to hold the root (/) file system.
- If the /usr file system is on the root drive and you also need to restore it, you need a partition at least large enough to contain the restored file system. (Consider expansion requirements, if appropriate.)
- If the /var file system is on the root drive and you also need to restore it, you need a partition at least large enough to contain the restored file system unless it is included in /usr on the original root drive.
- If primary or tertiary swap partitions were on the original root drive, you must recreate these partitions on the replacement drive.

You can restore other file systems as required, or restore them to different devices and then remount them by updating the /etc/fstab file after the restoration is complete.

See `disklabel(8)` for more information on creating a disk label with custom partitions.

### 9.7.5.3 Applying the Procedure

Some steps in the procedure are dependent on your system's original configuration. Ignore these steps if they do not apply to your configuration. The optional steps are marked [Configuration Dependent].

In the procedure, you always proceed to the next step unless redirected.

1. Boot the system from the operating system distribution media by using one of the following methods:
  - Insert the distribution CD-ROM that contains the operating system version that you wish to restore and boot the operating system



specifying the CD-ROM reader device name that you determined previously. For example:

```
>>> boot dka400
```

- Boot from your local RIS server. For example:

```
>>> boot ewa0
```

2. [Configuration Dependent] If you are already using the character-cell installation procedure, go to step 3, otherwise complete the following task.

If your system has a graphics console, the installation defaults to graphical mode. Wait until the installation procedure displays a dialog box titled Installation Welcome.

Pull down the File menu and select the Exit option to invoke character-cell mode.

3. Verify the status of the backup device and the target disk (the restored root disk) by using the following command:

```
# hwmgr view devices
```

The hwmgr command displays a list of all devices currently recognized by the system as shown in the following example:

HWID:	Device Name	Mfg	Model	Location
4:	/dev/kevm			
28:	/dev/disk/floppy0c		3.5in floppy	fdi0-unit-0
31:	/dev/disk/dsk0c	DEC	RZ26L (C)	DEC bus-0-targ-0-lun-0
32:	/dev/disk/dsk1c	DEC	RZ26 (C)	DEC bus-0-targ-1-lun-0
33:	/dev/disk/dsk2c	COMPAQ	HB00931B93	bus-0-targ-3-lun-0
34:	/dev/disk/cdrom0c	DEC	RRD45 (C)	DEC bus-0-targ-4-lun-0
35:	/dev/disk/dsk3c	COMPAQ	HB00931B93	bus-0-targ-5-lun-0
37:		DEC	TLZ06 (C)	DEC bus-0-targ-6-lun-0

Locate and write down the following data:

- The device name for the target disk.

This is important if you installed a new replacement disk. The device name is the entry under the Device Name column, such as /dev/disk/dsk2. Ignore the partition suffix (c).

If there is no entry for a newly installed target disk you cannot proceed. You must shut down the system, verify the disk's physical installation, and restart the recovery procedure.

- The device name for the backup device.

The device name appears in the Device Name column. For example, if you are restoring the root disk from the default tape device TLZ06 the Device Name column must contain a device special file name such as /dev/ntape/tape0.

There might be no device special file name in the `Device Name` column for the backup device, as shown in the preceding example. In this case, go to 4 to install the tape device.

4. Install the backup device by using the following command:

```
# dn_setup -install_tape
```

To verify the installation and determine the device name (such as `tape0_d0`), repeat the `hwmgr` command in step 3.

5. [Configuration Dependent] If the original file system format is unknown, you can now ascertain it and verify that you have a readable backup tape as follows:

- a. Load the backup (dump) media into the device.
- b. Invoke the interactive mode of the `restore` command, specifying the backup device name that you determined in step 4. For example:

```
# restore -i -f /dev/ntape/tape0_d0
```

- c. If the backup is good, a prompt for interactive restoration is displayed. Enter the `what` command to display the header and record the information.

6. Create and apply a disk label by using the following information:

- a. The partition plan that you created during recovery planning, including any swap space requirements. (See Section 9.7.5.2.)
- b. The new root device name determined in step 3.

Specify the a partition and label the drive as a bootable device. For example:

```
# disklabel -wr /dev/disk/dsk2a
```

7. Create your UFS target file systems as follows:

You must create file systems on the new root drive for each file system that you need to restore. For example, to create the new root and `/usr` file systems on partitions `a` and `g`, use commands similar to the following:

```
# newfs /dev/disk/dsk2a
# newfs /dev/disk/dsk2g
```

8. Mount the replacement disk on the temporary mount point `/mnt` according to the type of file system. For example:

```
# mount /dev/disk/dsk2a /mnt
```

9. Use the `vrestore` or `restore` command to restore files. For example:

```
# cd /mnt
# vrestore -x device
```

10. Shut down and halt the system by using the following command:

```
# shutdown -h now
```

11. Boot the system to single-user mode, specifying the restored root drive as the boot device. For example:

```
>>> boot dka2 -flags s
```

If you are using an alternate drive, or if you installed a new drive, you might need to translate the system device name to the appropriate boot device name. In step 3, you used the `hwmgr` command to determine the device database entry for the new device. For example:

```
33: /dev/disk/dsk2c    COMPAQ    HB00931B93    bus-0-targ-3-lun-0
```

Use the following command to display the devices:

```
>>> show device
```

Map the value of the *b/t/l* (in this case 0.3.0) to the alternate or new device and identify its boot device name, such as `dka300`.

12. If the boot is successful, run the following script to update the device database:

```
# /sbin/mountroot
```

While the `dsfmgr` command attempts to update the device database, some error or warning messages might be displayed. You can ignore the messages.

13. [Configuration Dependent] If you installed a new drive for root, or you specified an alternate device, you need to rename devices. Using the device name information that you determined in step 3, rename the devices as follows:

- If you remove the old root disk and replace it with a new device, use the `dsfmgr` command with the `-m` option to move the device names. For example:

```
# dsfmgr -m dsk20 0
```

- If the old root disk is still connected to the system, use the `dsfmgr` command with the `-e` option to exchange the device names. For example:

```
# dsfmgr -e dsk20 0
```

14. Using the interactive mode of the `vrestore` command, load the backup media into the restore device and restore the device directories. This

step ensures that all appropriate devices, including any custom device drivers are recreated:

- Delete the existing directories as follows:

```
# rm -rf /cluster/members/member0/dev*
```

- If you used the dump command, restore the directories as follows:

```
# mkdir -p /usr/tmp /usr/var/tmp
# restore -i -f /dev/ntape/tape0_d0
restore > add /cluster/members/member0/dev
restore > add /cluster/members/member0/devices
restore > extract
```

15. Use the `dsfmgr` command to verify the device databases and device special file names as follows:

```
# dsfmgr -v
```

16. This is the end of the procedure. Assuming that you have restored all the required file systems, including `/usr` and (if necessary) `/var`, you can now shut down the system, redefine the boot device, and reboot the system to multiuser mode as follows:

```
# shutdown -h now
>>> set bootdef_dev dka300
>>> boot
```

You can verify success by checking the boot process for any error messages relating to devices. If you determine that the procedure is not successful, your only option is to reinstall the operating system from the distribution media and recreate your customized environment.

#### 9.7.5.4 Using Alternative root Disk Duplication Methods

Other methods of restoring or duplicating the root disk depend on whether or not you configured your system in anticipation of such problems. The following methods are available:

- Methods of duplicating the root disk, such as mirroring, which are available when LSM and AdvFS are in use. Refer to the *AdvFS Administration* and *Logical Storage Manager* guides for more information.

Refer to the *Logical Storage Manager* guide for information on setting up a recoverable root volume.

- Bootable tape provides a method of restoring customized systems that is faster than a conventional backup tape. This feature is available only on certain configurations. Refer to Chapter 9 for more information.

- Backing up your customized system files can assist recovery if you have to reinstall the operating system. If you used the update installation feature (`updateinstall`) for the previous installation, you can create an archive of customized system files. Refer to the *Installation Guide* for more information.

### 9.7.6 Restoring the /usr and /var File System

You might need to restore the root file system as described in Section 9.7.5 before you can restore the `/usr` file system. If the `/var` directory is on a file system other than `/usr`, repeat the steps in this section for restoring the `/var` file system.

The procedure in this section requires that you have access to the most recent dump files of your `/usr` file systems. The following steps show how you restore from a level 0 dump of files, by using the text-based (or character cell) interface to perform the task:

1. Label the disk if required by using the `disklabel` command as follows:

```
# disklabel -rw /dev/disk/dsk0
```

---

#### Note

---

The options used with the `disklabel` command in this procedure will write a default disk partition table to the disk. If the disk you are restoring has a customized partition table, invoke the editing option of the `disklabel` command or restore the partition table from your protofile label. See Chapter 6 and `disklabel(8)` for more information.

---

2. Create a new file system by using the `newfs` command. For example:

```
# newfs /dev/rdisk/dsk1c
```

3. If necessary, create mount points by using the `mkdir` command. For example:

```
# mkdir /usr
```

4. Mount the file system by using the `mount` command, For example, to mount the file system created in the previous step, enter:

```
# mount /dev/disk/dsk1c /usr
```

5. Restore the file system:

- If you are restoring dump files from a local file system, change to the restore directory, insert the medium containing the dump file, and enter the `restore` command. For a tape, you might enter the following commands:

```
# cd /mnt
# restore -Yrf /dev/tape/tape0_d0
```

- If you are restoring dump files from a remote system, change to the restore directory and use the `rsh` command. You might need to specify the following command options:
  - `remote_hostname` – The host name of the remote system that contains the dump file
  - `dumpfile` – The full pathname of the dump file on the remote system.
  - `blocksize` – The block size is required to read data from a tape.

Read the dump file using the same block size that you specified when you wrote the tape. The default dump record size is 10 KB.

For example, to restore a dump file on a TLZ06 from the remote system `remotesystem` that you wrote by using the default block size, enter the following:

```
# cd /mnt
# rsh remotesystem "dd if=/dev/tape/tape0_d0 bs=10k" \
| restore -Yrf -
```

## 9.8 Using the Command-Line Utilities, tar, pax, and cpio

The `tar`, `pax`, and `cpio` command-line utilities provide a method of quickly creating an archive from the command line or for writing scripts to back up files. The disadvantage is that you might have to type long command strings, and backing up or restoring large volumes of files and directories is not easy when using these interfaces. You might use these utilities make a small archive of files for distribution to other users, such as a program, its sources, and associated documentation.

The following examples demonstrate how you can create or restore typical archive files by using the command-line utilities.

### Using tar to Create an Archive

The `tar` command saves and restores multiple files on a single device such as a disk or tape.

To use the `tar` to create an archive on tape drive `/dev/tape/tape12_d0`, enter a command such as the following

```
# tar cvfb /dev/tape12 -e ./netscape -C /usr/glenn
```

The resulting archive contains all files and directories in the `/usr/glenn` directory, except for file `./netscape`. See `tar(1)` for more information.

### Using pax to Create an Archive

The `pax` command extracts, writes, and lists members of archive files. It also copies files and directory hierarchies.

To use the `pax` command to create an archive of the current directory to device `/dev/tape/tape0_d0`, enter:

```
# pax -w -f /dev/tape/tape0
```

The following command reads the archive `a.pax`, extracting all files rooted in the `/usr` directory, relative to the current directory:

```
# pax -r -s ',^//*usr//*,,' -f a.pax
```

See `pax(1)` for more information.

### Using cpio to Create an Archive

The `cpio` command copies files between archive storage and the file system. It is used to save and restore data from traditional format `cpio` archives.

To use the `cpio` command to create an archive to tape device `/dev/tape/tape12_d0`, enter:

```
# cpio -ov < file-list -O/dev/tape12_d0
```

See `cpio(1)` for more information.

## 9.9 Using dxarchiver

The Archiver, `dxarchiver`, is a graphical user interface 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

Because the `dxarchiver` GUI is a CDE application, you can drag and drop files and directories (folders) to assemble an archive set, without having to type long commands.

It is assumed that you gathered the information Section 9.5.3, 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 the `/usr/bin/X11/dxarchiver` GUI from a terminal command line, or open the CDE Application Group: `System_Admin`. Then open System Admin Subgroup: `Daily Admin` and click on the Archiver icon.
2. Select the Archive Type: `tar`, `cpio`, or `pax`. Not all command-line options might be available from the graphical interface.
3. Select any Archive Options. You can only append to an existing archive, and you cannot further compress an existing archive that was compressed on creation. Specify either an absolute or a relative pathname as the method of storing the directories. (An absolute pathname is the full path, beginning at the root directory such as `/usr/users`. A relative pathname begins at the current directory, for example dot (`.`) or `users/chan`.)

During future recovery of these files, you can write them to a temporary location only if you specified a relative path during the original archiving process. Otherwise, files are restored to their original locations. (Potentially overwriting the existing version of the file unless you rename it.)

4. Specify the source, the files, and directories to archive. You can type pathnames or you can open a File Manager view and drag files and directories (CDE folders) to the Source Container box within the Archiver window. If you type pathnames, use the OK button to add them to the container.
5. After all required files are specified, choose the Archive... option and the Archiver: Archive window is displayed.
6. Enter a destination path, such as:
  - `/dev/tape/tape0_d0` for the default tape device.
  - `/usr/backup/myback_991803` for a disk archive. You do not need to enter a file name extension; the Archiver adds an identifier such as `.Z`.

After you choose OK, the destination is displayed under the Destination Container box.

7. Press Create Archive. A window titled Archiver working is displayed, flashing a green button to indicate that the archive is being written. The files being archived are displayed in the Destination Container.
8. After the archive is complete, you can optionally print a copy of the files list to keep as a record with the tape.



9. Choose 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 is read and the contents displayed in the Show Contents Window.

To extract an archive, you must specify a destination on a target device such as a disk. If you are not recovering a damaged file system on a complete disk partition, you might consider using a temporary location rather than overwriting existing directories. You can then restore individual files and directories as needed. You can also restore selected files from the archive as follows:

1. Invoke the `/usr/bin/X11/dxarchiver` GUI from a terminal command line, or open the CDE Application Group: System\_Admin. Then open System Admin Subgroup: Daily Admin and click on the Archiver icon.
2. Choose Show Contents... to select individual files and directories. The tape or archive file is read and the contents displayed in the Archiver Show Contents window. Select individual files or directories as follows:
  - In the Archiver Show Contents window, 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.
  - After you select all the files that you want, choose OK in the Archiver Show Contents window. The files are displayed in the Source Container box in the Archiver main window. You can use the Edit menu to make additional changes to selections. For example, highlight an entry in the source container and choose Edit: Clear Selected Source to delete it.
3. Choose the Extract... option to display the Archiver Extract window.
4. Enter a destination directory. This directory can be the same as the archive, assuming that files can be overwritten. Alternatively, give the path to a temporary location. This path must be to an existing directory, or you must open a terminal and create it with the `mkdir` command. Alternatively, create a folder by using the New Folder option in CDE File Manager. The destination is displayed under the Destination Container box.
5. Choose Extract Contents to begin the extraction. A window titled Archiver Working is displayed, flashing a green button to indicate that the archive is being extracted. The files being recovered are displayed in the Destination Container.
6. After the archive is complete, you can optionally print a copy of the files list, to keep as a record.

7. Choose Cancel to return to the Archiver main window. Before exiting, use the File Manager or a terminal window to ensure that the files were recovered as expected and that the file contents are not corrupted. 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 required file systems to the tape for data or programs that you might need on the recovered system.

To create the SAS kernel, you must use the SysMan Menu Create a Bootable Tape option or the `btcreate` command-line utility. After you create the kernel, you can restore the customized image by using the `btextract` utility. You can invoke only a single instance of the `btcreate` utility. The `/usr/run/bttape.pid` locking file prevents multiple instances of the utility.

The following sections provide an overview of the `bttape` interfaces, SysMan Menu task, and the `btcreate` and `btextract` command-line utilities.

### 9.10.1 Tape Device Requirements

If you use QIC tape drives to create bootable tapes, you must use only high-density tapes of 320 or more megabytes. The QIC-24, QIC-120, and QIC-150 format tapes of fixed-512 blocks do not work. Tapes with a variable block size, such as the QIC-320 and QIC-525, work with bootable tape. Using an improperly configured QIC tape drive to create a bootable tape results in an I/O error, a write error, or permission denied error. Therefore, you must take one of the following actions:

- Configure the drive at installation time
- Rebuild the kernel if the drive was attached to the system after the installation

A QIC tape created with the `btcreate` command might fail with the following error message when booted:

```
failed to send Read to mka... Be sure that the tape is
write protected before booting.
```

If you are creating a bootable tape with a file system that extends to multiple tapes, the `/sbin/dump` command displays a message indicating that you must change the tape. If you do not change the tape promptly, warning messages repeat periodically until you change the tape.

The behavior of the open call to a tape device has changed. You can no longer use write mode to open a write-protected tape. An attempt to open the tape fails, returning the following message:

```
EACCES (permission denied)
```

If an application is written so that it attempts to open the tape device with `O_RDWR` permission when the intention is only to read the tape, the open attempt fails. Change any applications or scripts to open the device with `O_RDONLY` permission. Use the following command to obtain the previous behavior of the open call for applications that cannot be changed:

```
# sysconfig -r cam_tape open_behaviour=0
```

## 9.10.2 Using the `btcreate` Utility

To build a bootable SAS kernel on UFS or AdvFS file systems only, you must use the `btcreate` utility. The following sections provide an overview of the information you must have to create the SAS kernel on tape.

The `btcreate` command provides both a noninteractive and interactive user interface. Both require that you have superuser (root) privileges.

### 9.10.2.1 Gathering Information

To prepare to use the `btcreate` command, you must have the following information:

- Name of the kernel configuration file in the `/usr/sys/conf` directory. The default is the same as the system (HOST) name in capital letters.
- Name of the disk partition (for example, `disk2e`) where the miniroot file system is to reside. Minimum size needed on the disk is 38000 blocks (512 bytes per block). This disk partition must not be in a mounted state when `btcreate` is executed.
- Name of the tape device, for example `/dev/tape/tape0_d0`, where the SAS kernel and file systems are to reside.
- Device name, mount point, and type of each file system (UFS or AdvFS) that you want to back up to the tape device. The following examples show valid UFS and AdvFS entries:

UFS:

```
/dev/dsk1a /      ufs
/dev/dsk1g /usr   ufs
/dev/vol/rootdg/rootvol /ufs
```

**AdvFS:**

```
root_domain#root /      advfs
usr_domain#usr   /usr   advfs
```

---

**Note**

---

Do not select swap partitions for file system backups.

---

For UFS file systems that are stored on LSM volumes, the `vdump` and `vrestore` utilities are used during bootable tape creation.

- An *addlist\_file* file, which lists the files or directories you want to include on the miniroot file system.
- An *fslist\_file* file, which specifies the file systems to back up.
- A `/usr/lib/sabt/sbin/custom_install.sh` script, if you want to customize the restored system image. Write the script using the Bourne shell language (`sh1`) because it is the only shell provided on the miniroot file system. The `btcreate` command copies the `custom_install.sh` file onto tape and places it in the `sbin` directory on the miniroot file system. The `btextract` command invokes the `custom_install.sh` script before exiting.

The following additional features might be useful in planning your bootable tape layout:

- Use the `-d` option to specify the location where `btcreate` creates its temporary files. If you do not specify a location, 156000 blocks (512 bytes per block) of disk space in the `/usr` file system is required.
- You can label disks using your own custom `disklabel` script, which must meet the following requirements:
  - It must be located in the `/usr/lib/sabt/etc` directory.
  - It must be named `custom_disklabel_file`.

If a custom `disklabel` script is not present, the `btextract` command labels the disks in the usual manner. See `disklabel(8)` for more information.

### 9.10.2.2 Creating the SAS Kernel

To create the SAS kernel, the `btcreate` command copies the `/usr/sys/conf/YOUR_SYSTEM_NAME` configuration file to

`/usr/sys/conf/YOUR_SYSTEM_NAME.BOOTABLE` and modifies it as follows:

```
config      vmunix      root  on md
pseudo-device      memd  38000
```

These modifications configure a memory file system of 38000 blocks. The memory file system and the disk partition where the miniroot file system reside are equivalent in size.

After modifying the configuration file, the `btcreate` command executes the `doconfig` command and moves the bootable kernel to the `/usr/sys/bin` directory. For information on the command syntax and options, see `btcreate(8)`.

### 9.10.3 Using the `btextract` Utility

The `btextract` command is a shell script that restores file systems from tapes that contain a SAS kernel created by using the `btcreate` utility. You can perform a default restoration or an advanced restoration of the system.

During a default restoration, you can choose to duplicate the customized system on more than one system of the same hardware platform type. You cannot specify which disk partitions to use for the restore operation. Instead, the `btextract` command restores file systems using the disk partition information gathered during the `btcreate` session and all existing information is overwritten. If you are performing an advanced restoration, you can optionally specify which disk partition to use. However you can duplicate the customized system only on a system of the same hardware platform type.

To use the `btextract` command, place the system in a halt state, initialize the system, then boot from the tape as follows:

```
>>> init
>>> show dev
>>> boot -fl "nc" MKA500
```

In this example, the `show dev` command provides the device name under `BOOTDEV` and `MKA500` is the `BOOTDEV`.

After the initial boot completes, the shell invokes the `btextract` utility. If you created a `/usr/lib/sabt/sbin/custom_install.sh` script during the `btcreate` session, the `btextract` command invokes the `custom_install.sh` script before exiting. You can also create a `custom_prerestore` answer file to automate the recovery procedure. See `btcreate(8)` for more information.

After the `btextract` command completes its task, you must shut down and reboot the system from the restored disk as follows:

```
# shutdown -h now
>>> boot DKA100
```

In this example, DKA100 is the BOOTDEV.

For more information and examples, see `btextextract(8)`.

### 9.10.4 Using the SysMan Menu `boot_tape` Option

The following steps describe the basic process for creating a bootable tape. It assumes that you have gathered the necessary device data as described in Section 9.10.2.1, and the tape device is ready to save.

1. Invoke the Create a Bootable Tape task from the SysMan Menu, or enter the following command at the prompt:

```
# sysman boot_tape
```

2. A window titled Bootable Tape Creation on *hostname* is displayed. Complete the fields or choose options as follows:
  - In the Kernel Name field, the default kernel name for the host is displayed. This is usually the same as the local host name. However, you can enter any name for the saved kernel.
  - The Miniroot File System field provides the following options:
    - The option to create the miniroot as a memory file system (mfs) or a Disk Partition. Click on the button for the option of your choice.
    - The option to specify a disk partition name such as `disk0b` with the Specify Disk Partition/mfs... button. (This button displays a dialog box in which you enter the disk partition name.)
  - The Tape Device field contains the name of the default tape device, usually `tape0_d1`. This is the name of the device on which the SAS kernel is saved, but you can specify any other supported device.
  - The Customizing the Miniroot File System field displays the default file location for the `addlist` file. This is a data file that contains a list of additional files that you want to include, such as commands or utilities. You cannot exceed 360 KB of data in the mfs. This list is stored in the `/usr/lib/sabt/etc/addlist` file by default but you can choose to create your own location.

To create a new append file, or modify an existing append file:

- a. Click on the Create/Modify Miniroot Append File button to display the Create/Modify window.
- b. Click on the Add button 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 is located, such as `/sbin`. Click on the OK button to return to the Create/Modify window.

- c. The Contents of file: box contains a list of the files to be appended. Click on the OK button 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. Click on the Create/Modify File Systems Backup File... button to display the Create/Modify window.
  - b. Click on the Add button 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`. Click on the OK button to return to the Create/Modify window.
  - c. The file systems to be backed up are listed in the Contents of file: box. Click on the OK button to return to the main window for Bootable Tape Creation.
3. After completing the required fields, you are ready to create the tape. In the main window for Bootable Tape Creation, click on the OK button to proceed. A message window is displayed to indicate that the task has started. The creation of the tape can take twenty or more minutes, depending on the speed of the devices used.

If the task cannot be completed, a message is displayed informing you that the error log is located in `/var/adm/btcreate.log`.
4. After the tape is successfully written, a message is displayed confirming the success and the location of the log file, `/var/adm/btcreate.log`.

Print `btextract(8)`  
and store it with the tape for future reference.
5. Use the instructions in Section 9.10.3 and `btextract(8)` to restore the bootable SAS kernel. Consider running a test recovery to ensure that any future recovery works as intended.





# 10

---

## Administering the System Accounting Services

This chapter describes how to set up and use the system accounting services. 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:

<b>File</b>	<b>Description</b>
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
<code>ac</code>	Command	Displays connect session records.
<code>acctcms</code>	Command	Formats the binary command usage summary files.
<code>acctcom</code>	Command	Displays process accounting record summaries from the default <code>pacct</code> database file or a specified file.
<code>acctcon1</code>	Command	Summarizes the records in the <code>wtmp</code> file in ASCII format.
<code>acctcon2</code>	Command	Summarizes the contents of the files formatted by the <code>acctcon1</code> command.
<code>acctdisk</code>	Command	Performs comprehensive disk usage accounting.
<code>acctdusg</code>	Command	Performs disk block usage accounting.
<code>acctmerg</code>	Command	Merges accounting record files.
<code>accton</code>	Command	Turns on process accounting.
<code>acctprc1</code>	Command	Displays records of <code>acct</code> type structure by user identification number and login name.
<code>acctprc2</code>	Command	Displays records of <code>acct</code> type structure by user identification number and full name.
<code>acctwtmp</code>	Command	Writes records to the <code>/var/adm/wtmp</code> file.
<code>chargefee</code>	Script	Writes a charge-fee record to the <code>/fee</code> database file.

**Table 10–1: Accounting Commands and Shell Scripts (cont.)**

<b>Name</b>	<b>Type</b>	<b>Description</b>
ckpacct	Script	Checks the size of the <code>/var/adm/acct/pacct</code> 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 <code>/var/adm/nite/dacct</code> disk usage accounting database file.
fwtmp	Command	Displays the <code>/var/adm/wtmp</code> 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 <code>/var/adm/acct/sum/loginlog</code> file.
monacct	Script	Creates monthly summary accounting report files.
nulladm	Script	Creates files that are owned by the <code>adm</code> user and group and that have 664 permission.
pac	Command	Displays printer accounting records.
prctmp	Script	Displays the <code>/var/adm/acct/nite/ctmp</code> connect session record file.
prdaily	Script	Collects and displays daily accounting records from various files.
printpw	Command	Displays the contents of the <code>/etc/passwd</code> file.
prtacct	Script	Formats in ASCII and displays a <code>tacct</code> daily accounting file.
remove	Script	Removes any <code>/var/adm/acct/sum/wtmp*</code> , <code>/var/adm/acct/sum/acct/pacct*</code> , and <code>/var/adm/acct/nite/lock*</code> 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 <code>/var/adm/wtmp</code> 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 <code>dodisk</code> shell script.
fee	ASCII	Contains output from the <code>chargefee</code> shell script.

**Table 10–2: Database Files in the /var/adm Directory (cont.)**

<b>Name</b>	<b>Type</b>	<b>Description</b>
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.
pacctn	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 pacctn file is created, the value <i>n</i> is incremented by one.
qacct	Binary	Contains queuing (printer) system accounting records. This file is used by the runacct shell script.
savacct	Binary	Specifies the file used by the sa command to store system process accounting summary records.
Spacctn.mmdd	Binary	Specifies the pacctn files produced by the runacct shell script for the month and day specified by <i>mm</i> and <i>dd</i> , 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**

<b>Name</b>	<b>Type</b>	<b>Description</b>
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.
activemmdd	ASCII	Specifies the daily runacct shell script error file for the month and day specified by <i>mm</i> and <i>dd</i> , respectively. This file is similar to the active file.

**Table 10–3: Daily Files in the /var/adm/acct/nite Directory (cont.)**

<b>Name</b>	<b>Type</b>	<b>Description</b>
cklock	ASCII	Specifies the file the <code>ckpacct</code> shell script uses to ensure that more than one <code>runacct</code> shell script is not called during any 24-hour period. This file is removed each day if the <code>runacct</code> shell script has completed.
cms	ASCII	Specifies the active total daily command summary file. This file is the ASCII version of the <code>/var/adm/acct/sum/cms</code> file. This file is created by the <code>acctcms</code> command, which is called by the <code>runacct</code> shell script to rewrite the <code>/var/adm/acct/sum/cms</code> file records. The <code>monacct</code> shell script initializes this file.
ctacct.mmdd	Binary	Specifies the connect accounting records in <code>tacct.h</code> format that are obtained from the connect session accounting records for the month and day specified by <code>mm</code> and <code>dd</code> , respectively. This file is temporary and is deleted after the <code>daytacct</code> file records are written for each accounting period.
ctmp	ASCII	Specifies the temporary login/logout record file. This file contains the output of the <code>acctcon1</code> accounting command, which is called by the <code>runacct</code> shell script to rewrite the <code>wtmp</code> file records.
daycms	ASCII	Specifies the daily command summary file. This file is the ASCII version of the <code>/var/adm/acct/sum/daycms</code> binary file. The <code>runacct</code> shell script calls the <code>prdaily</code> shell script, which invokes the <code>acctcms</code> command to create the file.
daytacct	Binary	Contains the total accounting records in <code>tacct.h</code> format for the previous day.
dacct	Binary	Contains the weekly total disk usage accounting records when the <code>acctdisk</code> command is called by the <code>dodisk</code> shell script.
lastdate	ASCII	Specifies the last day that the <code>runacct</code> 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.)**

<b>Name</b>	<b>Type</b>	<b>Description</b>
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.
ptacctn.mmd	Binary	Specifies the additional daily pacctn files for the month and day specified by <i>mm</i> and <i>dd</i> , 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.mmd	Binary	Specifies the fixed daily login/logout accounting database file for the month and day specified by <i>mm</i> and <i>dd</i> , 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.
wtmperrormmd	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 <i>mm</i> and <i>dd</i> , respectively.

**Table 10–4: Summary Files in the /var/adm/acct/sum Directory**

<b>Name</b>	<b>Type</b>	<b>Description</b>
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 <code>runacct</code> 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 $mm$ $dd$	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 <code>runacct</code> shell script.
tacct $mm$ $dd$	Binary	Specifies the total accounting file for the month and day specified by $mm$ and $dd$ , respectively.
tacctprev	Binary	Specifies the previous day's <code>tacct</code> file. This file is the <code>tacct</code> 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 <code>monacct</code> 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` queuing accounting file and the `pacct` process accounting database file must exist on the system for accounting to function. These files are preinstalled as blank files with path names that are context-dependent symbolic links (CDSLs). When you use the `ls -l` directory display command, the links resolve to the following paths:

```
/usr/var/cluster/members/member0/adm/acct/fee  
/usr/var/cluster/members/member0/adm/acct/pacct
```

```
/usr/var/cluster/members/member0/adm/acct/qacct
```

If the original files do not exist (or have been 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:

<code>ckpacct</code>	This shell script checks the size of the <code>pacct</code> process accounting database file and ensures that it does not become too large.
<code>runacct</code>	This shell script includes other accounting shell scripts and commands and creates daily and monthly accounting files. You can modify the <code>runacct</code> shell

script to remove the commands for the accounting features that you do not want.

<code>monacct</code>	This shell script creates monthly summary accounting files. You can modify the <code>monacct</code> shell script to remove the commands for the accounting features that you do not want.
<code>ac</code>	This command displays connect-time records. You can direct the output to a file. You can also add this command to the <code>runacct</code> shell script.
<code>pac</code>	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
<code>ac</code>	This command displays connect session records for the entire system and for each user.
<code>acctcon1</code>	This command summarizes connect session records and displays those records in ASCII format, using one line for each connect session.
<code>acctcon2</code>	This command uses the output of the <code>acctcon1</code> command to produce an accounting record file of the total connect session in ASCII format.
<code>acctwtmp</code>	This command enables you to write records to the <code>wtmp</code> file by entering them from the keyboard.
<code>fwtmp</code>	This command displays records from files with the <code>utmp.h</code> file structure.
<code>last</code>	This command displays login information.
<code>lastlogin</code>	This shell script updates the <code>/var/adm/acct/sum/loginlog</code> file to show the last date that each user logged in.
<code>prctmp</code>	This shell script displays the contents of the session-record file (usually <code>/var/adm/acct/nite/ctmp</code> ) that the <code>acctcon1</code> command created.
<code>wtmpfix</code>	This command corrects the <code>wtmp</code> connect session records that are affected by a date modification and validates login names written to the login name field in the <code>wtmp</code> 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.mmd`

- `/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	<code>ut_user</code>	The user login name, which must have exactly <code>sizeof(ut_user)</code> characters.
2	<code>ut_id</code>	The <code>inittab</code> ID, which must have exactly <code>sizeof(ut_id)</code> characters.
3	<code>ut_line</code>	A memory location, where information used to describe the type of record (for example, the device name) is stored. It must have exactly <code>sizeof(ut_line)</code> characters.
4	<code>ut_pid</code>	The process identification number.
5	<code>ut_type</code>	The type of entry, which can specify several symbolic constant values. The symbolic constants are defined in the <code>/usr/include/utmp.h</code> header file.
6	<code>ut_exit.e_termination</code>	The process termination status.
7	<code>ut_exit.e_exit</code>	The process exit status.
8	<code>ut_time</code>	The starting time (in seconds).
9	<code>ut_host</code>	The host name, which must have exactly <code>sizeof(ut_host)</code> 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/rprtmmdd` 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/rprtmmdd` 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  tty1  shout  Fri Jan 21 10:56  still logged in
root  tty1  raven  Fri Jan 21 08:59 - 09:00  (00:00)
root  tty0  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 `pacctn` 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 <code>/etc/passwd</code> file.
2	LOGNAME	Specifies the user login name, which is obtained from the <code>/etc/passwd</code> file.
3	PRI_CPU	Specifies the prime time CPU run time, which is the total time (in seconds) that prime time CPU run time was charged to the user.
4	NPRI_CPU	Specifies the nonprime time CPU run time, which is the total time (in seconds) that nonprime time CPU run time was charged to the user.
5	PRI_MEM	Specifies the prime time memory kcore minutes, which is the total CPU time (in minutes) multiplied by the mean size of the memory used.
6	NPRI_MEM	Specifies the nonprime time memory kcore minutes, which is the total CPU time (in minutes) multiplied by the mean size of the memory used.
7	PRI_RD/WR	Specifies the total number of characters transferred during prime time operation.
8	NPRI_RD/WR	Specifies the total number of characters transferred during nonprime time operation.
9	PRI_BLKIO	Specifies the total number of I/O blocks transferred during prime time read and write operations. The number of bytes in an I/O block depends on how it was implemented.
10	NPRI_BLKIO	Specifies the total number of I/O blocks transferred during nonprime time read and write operations. The number of bytes in an I/O block depends on how it was implemented.
11	PRI_CONNECT	Specifies the total number of prime time seconds that a connection existed.
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 <code>/usr/sbin/acct/charge-fee</code> 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 `pacctn` 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 `pacctn` 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      ttyt1      11:59:00 11:59:00   0.77    0.01    0.00
ls              root      ttyt1      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      ttyt1      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.

- ❏ Shows information about disk I/O operations, either the average number of I/O operations or the total number of I/O operations.
- ❏ Shows information about kiloblocks (number of blocks multiplied by 1024) used or the memory time integral.
- ❏ 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  ssh
   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
    ❏         ❏         ❏
```

The additional columns show the following information:

- ❏ Indicates the number of times each command was executed with respect to the total number of times all commands were executed.
- ❏ 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.
- ❏ 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      tty0      0.01 secs Fri Jan 21 11:34
sed      S  root      tty0      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 described 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 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
<code>runacct</code>	This shell script creates the daily and summary files in the <code>/var/adm/acct/nite</code> and <code>/var/adm/acct/sum</code> directories.
<code>acctmerg</code>	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.
<code>prtacct</code>	This shell script formats and displays accounting files that have the <code>/usr/include/sys/acct.h</code> header file structure. Each record includes information about the user identification number, connect time, process time, disk usage, and printer usage.
<code>prdaily</code>	This shell script creates an ASCII file that contains the accounting data from the previous day. When this script is invoked from the <code>runacct</code> script, it creates the <code>/var/adm/acct/sum/rprtmmdd</code> file.
<code>monacct</code>	This shell script creates cumulative process and total accounting files in the <code>/var/adm/acct/fiscal</code> 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 <code>acctcon1</code> 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 <code>acctmerg</code> command to create the binary total accounting file.
FEES	Uses the <code>acctmerg</code> command to merge records from the <code>fee</code> file into the binary total accounting file.
DISK	Uses the <code>acctmerg</code> command to merge disk-usage records into the binary total accounting file.
QUEUEACCT	Uses the <code>acctmerg</code> 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 <code>acctmerg</code> 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/activemmdd` 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?.mddd already exists run setup manually
```

You must invoke the `runacct` shell script manually from the `MERGE` state. The question mark (?) specifies a single-character wildcard. The actual file name includes a version number and a date such as `var/adm/Spacct1.1101`.

```
ERROR: wtmpfix errors see nite/wtmperror
```

An unrepairable `wtmp` file was found during the `WTMPFIX` state. Use the `fwtmp` command to correct the file.

ERROR: invalid state, check /usr/var/adm/nite/active

During processing, the runacct shell script may have detected a corrupted active file. 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/tacctmmd` file. This example assumes that the `/var/adm/sum/tacctmmd` file needs to be fixed.
3. To obtain an ASCII version of the `/var/adm/sum/tacctmmd` 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/tacctmmd` 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/rprtmmdd` 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/tacctmmd`
- `/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[ mmd]] | [-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:

<code>tacctmm</code>	Specifies the binary total accounting file for the month preceding the month specified by the <i>mm</i> variable.
<code>cmsmm</code>	Specifies the binary cumulative command summary file for the month preceding the month specified by the <i>mm</i> variable.
<code>fiscrptmm</code>	Specifies the ASCII total monthly accounting report file. This file has a format that is similar to the <code>/var/adm/acct/sum/rprtmmdd</code> report file and is created from the following files: <ul style="list-style-type: none"> <li>• <code>/var/adm/acct/fiscal/tacctmm</code></li> <li>• <code>/var/adm/acct/fiscal/cmsmm</code></li> <li>• <code>/var/adm/acct/sum/loginlog</code></li> </ul>



---

## 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 be 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. 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_check` utility 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_check` utility.

### 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 by using the `runsyscheck` script with the `cron` utility as described in Section 11.2.2. In the event of a system problem, the regularly-updated report is useful when analyzing and troubleshooting the problem.

---

#### Note

---

Crash dump data 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, 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 can impact 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
```

See `vmstat(1)` or more information.

iostat

The `iostat` command reports input and output information for terminals and disks and the percentage of time the CPU has spent performing various operations. An example of `iostat` output is:

```
bigrig> iostat
      tty          floppy0          dsk0          cpu
  tin tout    bps    tps    bps    tps  us ni sy id
    0   1     0     0     3     0   0  0  1 98
```

See `iostat(1)` for more information.

who

The `who` command reports 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
```

See `who(1)` and `users(1)` for more information.

uptime

The `uptime` command reports how long the system has been running. See `uptime(1)` for more information.

Refer also to the `netstat` command and the *Network Administration: Connections* 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 is 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 invokes 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: Connections* 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 report output might 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 (`advfsd`), 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 by using the basic `binlogd` and `syslogd` event

channels. This chapter also describes how you access system log files, where events and errors are recorded.

- 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: Connections* 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 describes 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 `collect(8)`. Important features provided by the `collect` utility are:

- Controlling the duration of, and rate at which data is sampled. Sorting the output according to processor usage.
- Extracting a time slice of data from a data record file. For example, if you want to look at certain system parameters during the busiest time of use, you can extract that data from the data file by using the `-C` option.

- Specifying a particular device using its device special file name. For example the following command identifies that data is collected from the named devices:

```
# collect -sd -Ddsk1,dsk10
```

- Specifying a particular subsystem such as the CPU or the network. For example, the following command specifies that data is collected only for the CPUs, and a sample of data is shown:

```
# collect -e cf
CPU SUMMARY
USER SYS IDLE WAIT INTR SYSC CS RUNQ AVG5 AVG30 AVG60 FORK VFORK
 13 16 71 0 149 492 725 0 0.13 0.05 0.01 0.30 0.00
SINGLE CPU STATISTICS
CPU USER SYS IDLE WAIT
 0 13 16 71 0
```

- Recording and preserving a series of data files by using the `-H` (history) option. Compressing data files for economical storage.
- Specifying specific users, groups, and processes for which data is to be sampled.
- Using the `-p` option, 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 by using the page locking function `plock()`, and cannot be swapped out by the system. It also raises its priority by using the priority function `nice()`. If required, page locking can be disabled by using the `-ol` command option and the priority setting can be disabled by using the `-on` command option. However, using `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` utility 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_check` utility typically varies between 0.5MB and 3MB in size and it can take from 30 minutes to an hour to complete the check. See `sys_check(8)` 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.

Before running MPH, review the following information:

- The Standard Programmer Commands (Software Development) OSFPGMR400 subset must be installed. Use the `setld -i` command to verify that the subset is installed.
- The MPH software kit is contained in the mandatory base software subset OSFHWBASE400. This subset is installed automatically during the operating system installation. Full documentation is located in `/usr/field/mph/unix_installation_guide.ps`. A text file is also supplied.
- The disk space requirement for the MPH software subset is approximately 100 blocks.

To configure MPH on your system, you must be the root user and 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 `mph(1)` 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.

These components are described in the following sections.

### 11.3.1 Loadable Kernel Module

The loadable kernel module and its associated APIs contain the parameters needed to monitor and return status on your system's threshold levels. The kernel module exports server management attributes as described in Section 11.3.1.1 through the kernel configuration manager (CFG) interface only. It works across all platforms that support server management, and provides compatibility for other server management systems under development.



The loadable kernel module does not include platform-specific code (such as the location of status registers). It is transparent to the kernel module which options are supported by a platform. That is, the kernel module and platform are designed to return valid data if an option is supported, a fixed constant for unsupported options, or null.

### 11.3.1.1 Specifying Loadable Kernel Attributes

The loadable kernel module exports the parameters listed in Table 11–1 to the kernel configuration manager (CFG).

**Table 11–1: Parameters Defined in the Kernel Module**

Parameter	Purpose
<code>env_current_temp</code>	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 is not exceeded, a value of -1 is returned. If a system does not support temperature readings and a temperature threshold is exceeded, a value of -2 is returned.
<code>env_high_temp_thresh</code>	Provides a system-specific operating temperature threshold. The value returned is a hardcoded, platform-specific temperature in Celsius.
<code>env_fan_status</code>	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.
<code>env_ps_status</code>	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.
<code>env_supported</code>	Indicates whether or not the platform supports server management and environmental monitoring.

### 11.3.1.2 Obtaining Platform-Specific Functions

The loadable kernel module must return environmental status based on the platform being queried. To obtain environmental status, the `get_info()` function is used. Calls to the `get_info()` function are filtered through the `platform_callsw[]` table.

The `get_info()` function obtains dynamic environmental data by using the function types described in Table 11–2.

**Table 11–2: `get_info()` Function Types**

Function Type	Use of Function
<code>GET_SYS_TEMP</code>	Reads the system's internal temperature on platforms that have a KCRCM module configured.
<code>GET_FAN_STATUS</code>	Reads fan status from error registers.
<code>GET_PS_STATUS</code>	Reads redundant power supply status from error registers.

The `get_info()` function obtains static data by using the `HIGH_TEMP_THRESH` function type, which reads the platform-specific upper threshold operational temperature.

### 11.3.1.3 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
<code>svrThSensorReading</code>	<code>env_current_temp</code>
<code>svrThSensorStatus</code>	<code>env_current_temp</code>
<code>svrThSensorHighThresh</code>	<code>env_high_temp_thresh</code>
<code>svrPowerSupplyStatus</code>	<code>env_ps_temp</code>
<code>svrFanStatus</code>	<code>env_fan_status</code>

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. See `envmond(8)` and `envconfig(8)` for more information.

### 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 is 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 are started. You can customize the `envmond` daemon by 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 is broadcast.
- 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 must not alter any data in this file other than 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 provide 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 `/var/adm/binary.errlog` system error log file. This file also contains information on errors reported by the device drivers or by the system.

The log files provide a record of the diagnostics. However, after reading a log file, 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 `intro(2)` and `sigvec(2)`.

### 11.4.3 Exercising a File System

Use the `fsx` command to exercise the local file systems. The `fsx` command exercises the specified local file system by initiating multiple processes, each of which creates, writes, closes, opens, reads, validates, and unlinks a test file of random data.

---

#### Note

---

Do not test NFS file systems with the `fsx` command.

---

The `fsx` command has the following syntax:

```
fsx [-fpath] [-h] [-ofile] [-pnum] [-tmin]
```

Refer to `fsx(8)` for a description of the command options.

The following example of the `fsx` command tests the `/usr` file system with five `fsxr` processes running for 60 minutes in the background:

```
# fsx -p5 -f/usr -t60 &
```

### 11.4.4 Exercising System Memory

Use the `memx` command to exercise the system memory. The `memx` command exercises the system memory by initiating multiple processes. By default, the size of each process is defined as the total system memory in bytes divided by 20. The minimum allowable number of bytes per process is 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`

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 is 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] [-msize] [-ofile] [-pnum] [-tmin]
```

See `memx(8)` for a description of the command options.

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 `shmxsb`. The `shmx` command writes and reads the `shmxsb` data in the segments, and the `shmxsb` process writes and reads the `shmx` data in the segments.

Using `shmx`, you can test the number and the size of memory segments and `shmxsb` 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] [-ofile] [-v] [-ttime] [-msize] [-sn]
```

See `shmx(8)` for a description of the command options.

The following example tests the default number of memory segments, each with a default segment size:

```
# shmx &
```

The following example runs three memory segments of 100,000 bytes for 180 minutes:

```
# shmx -t180 -m100000 -s3 &
```

### 11.4.6 Exercising 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 whether the test continues.

---

You can use the `diskx` command options to specify the tests that you want performed and to specify the parameters for the tests.

The `diskx` command has the following syntax:

```
diskx [options] [parameters] -f devname
```

See `diskx(8)` for a description of the options.

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 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.

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]
```

See tapex(8) for a description of the command options.

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`.

You cannot test pseudodevice lines or `lta` device lines. Pseudodevices have `p`, `q`, `r`, `s`, `t`, `u`, `v`, `w`, `x`, `y`, or `z` as the first character after `tty`, for example, `ttyp3`.

The `cmx` command has the following syntax:

```
/usr/field/cmx [-h] [-o file] [-t min] [-l line]
```

See `cmx(8)` for a description of the command options.

The following example exercises communication lines `tty22` and `tty34` for 45 minutes in the background:

```
# cmx -l 22 34 -t45 &
```

The following example exercises lines `tty00` through `tty07` until you 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 channels. 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.

### 12.1 Understanding the Basic Event-Logging Facilities

The operating system uses three mechanisms to log system events:

- The system event-logging facility, also known as `syslogd`. Refer to `syslogd(8)` for information on the initialization options and `syslog.conf(4)` for information on configuration options. See `syslog.auth(4)` for information on remote logging.
- The binary event-logging facility also known as `binlogd`. Refer to `binlogd(8)` for information on the initialization options and `binlog.conf(4)`. See `binlog.auth(4)` for information on remote logging.
- The Event Manager (EVM) provides an integrated approach to administering system events and errors. See `EVM(5)` for an introduction to EVM, and see Chapter 13 for information on configuring and using EVM.

You can review events detected and recorded by `syslogd` and `binlogd` 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.

System events are often returned in a binary format. To render such events in a readable text format you must use a translation tool such as:

- The service tools provided with your service contract contain event analysis tools such as Compaq Analyze. Refer to <http://www.support.compaq.com/svctools/webes/index.html> for more information. Recent processor models produce `binlogd` events using a header format that differs from the format produced by earlier platforms. The newer format events are known as Common Event Header (CEH) events. If your system does not produce CEH events you cannot use Compaq Analyze to translate them, and you must install the `DECevent` formatter utility, `/usr/sbin/dia`.
- A limited use license for `DECevent` is provided in the distribution kit as described in the *Installation Guide*. Refer to the *DECevent Translation and Reporting Utility* guide and `dia(8)` for more information.
- The `uerf` command utility. See `uerf(8)`.

---

**Note**

---

The `uerf` command utility does not support CEH events and will be retired in a future release. You should migrate your event management procedures to EVM as soon as possible.

---

The log files created by the event-logging facilities are protected and owned by `root`, and belong to the `adm` group. You must have the proper authority to examine the files.

The following sections describe the event-logging facilities.

### 12.1.1 System Event Logging

The primary event-logging facility uses the `syslog` function to log system-wide events in ASCII format. The `syslog` function uses the `syslogd` daemon to collect the messages that are logged by the various kernel, command, utility, and application programs. The `syslogd` daemon logs the messages to a local file or forwards the messages to a remote system, as specified in the `/etc/syslog.conf` file.

When you install the operating system, the `/etc/syslog.conf` file is created and specifies the default event-logging configuration. The `/etc/syslog.conf` file specifies the file names that are the destination for the event messages, which are in ASCII format. Section 12.2.1.1 discusses the `/etc/syslog.conf` file. Refer also to `syslog.conf(4)`.

The `/etc/syslog.auth` file specifies which remote hosts are allowed to forward `syslog` messages to the local host. For system security, only

messages coming from remote hosts listed in this file are logged by the `syslogd` daemon. If the `/etc/syslog.auth` file is not present, then event forwarding from all remote hosts is enabled.

The `/etc/syslog_evm.conf` file specifies which `syslogd` messages are forwarded from the `syslogd` daemon to EVM, in the form of EVM events. Those `syslogd` messages are posted to the EVM daemon, `evmd`, by `syslogd` if the `syslogd` forwarding function is turned on with the `-e` option. Event forwarding is always on by default. Use the `-E` option to turn it off if required. Events are posted with the EVM name of `sys.unix.syslog.facility`.

Refer to `syslog.auth(4)` and `syslog_evm.conf(4)` for more information.

## 12.1.2 Binary Event Logging

The binary event-logging facility detects hardware and software events in the kernel and logs the detailed information in binary format records. Some events that are logged by the binary event-logging facility are also logged by the `syslog` function in a less detailed message.

The binary event-logging facility uses the `binlogd` daemon to collect various event-log records. The `binlogd` daemon logs these records to a local file or forwards them to a remote system, as specified in the `/etc/binlog.conf` default configuration file, which is created when you install your system. Section 12.2.1.3 discusses the `/etc/binlog.conf` file.

You use `DECEvent` (or `Compaq Analyze`) to translate binary events to ASCII reports from entries in the system's binary event log files. Invoke `DECEvent` by entering the `dia` command at the command line. Entering the command without any options causes `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 all events are displayed or you press `Ctrl/c`.

### Example 12–1: Sample Translated Event

---

```

**** V3.3 ***** ENTRY 4
***** [1]
Logging OS                               2 [OS] [2]
System Architecture                       2.
Alpha Event sequence number              440.
Timestamp of occurrence                   22-MAR-2001 18:24:31 [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 [4]                               1. O/S claims event is valid

```

### Example 12–1: Sample Translated Event (cont.)

---

```
Event severity          5. Low Priority
Entry type              301. Shutdown ASCII
Message Type            -1.    - (minor class)
SWI Minor class         9.
ASCII Message SWI Minor sub class 2. Shutdown ASCII Message
System halted by root: System going down @ 6:24PM on 22 Mar
Please log off in good time 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, an informational message that the system was shutting down.
- 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)

Compaq Analyze is a rules-based hardware fault management diagnostic application that provides error event analysis and translation. The multi-event correlation analysis feature of Compaq Analyze provides you with the capability to analyze events stored in the binary system event log or other specified binary log files. When Compaq Analyze is installed, you can launch its GUI interface directly from the SysMan Station by clicking on the Host Icon and selecting Compaq Analyze from the Tools menu.

## 12.2 Configuring Event Logging

You can change the default configuration by modifying the configuration files as described in this section. For example, you can change the configuration so that only important, system-critical events are logged and informational events are ignored. You can choose to concentrate on certain subsystems, such as mail or print services, and control how and where event messages are logged. The optimum method of monitoring system events is to use

Event Manager (EVM), as described in Chapter 13. EVM enables you to consolidate and filter events.

To enable system and binary event-logging, the special files must exist and the event-logging daemons must be running. Refer to Section 12.2.3 and Section 12.2.4 for more information.

The file `/var/adm/syslog.dated` and other files in `/var/adm` directory are context-dependent symbolic links (CDSLs), which facilitate joining single systems into clusters. The CDSL for the `syslog` directory is `/var/cluster/members/member0/adm/syslog.dated`. Take care not to break symbolic links when working with these files. 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, you can 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 following data:

- The facility, which is the source of a message or the part of the system that generates a message
- The priority, which is the message's level of severity
- The destination for messages.

The following sections describe how to edit the configuration files.

### 12.2.1.1 Editing the `syslog.conf` File

If you want the `syslogd` daemon to use a configuration file other than the default, you must specify the file name 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:

- ❶ Specifies the facility, which is the part of the system generating the message.
- ❷ 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.
- ❸ Specifies the destination where the messages are logged. This might be a log file or a device such as `/dev/console`.

The `syslogd` daemon ignores blank lines and lines that begin with a number sign (`#`). 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 tab characters or spaces.

You can specify more than one facility and its severity level by separating them with semicolons. In the preceding example, messages from the `auth` facility of `crit` severity level and higher and messages from the `syslog` facility of `debug` severity level and higher are logged to the `/var/adm/syslog.dated/syslog.log` file.

You can specify more than one facility by separating them with commas. In the preceding example, messages from the `mail` and `lpr` facilities of `debug` severity level and higher are logged to the `/var/adm/syslog.dated/misc.log` file.

You can specify the following facilities:

Facility	Description
<code>kern</code>	Messages generated by the kernel. These messages cannot be generated by any user process.
<code>user</code>	Messages generated by user processes. This is the default facility.
<code>mail</code>	Messages generated by the mail system.
<code>daemon</code>	Messages generated by the system daemons.
<code>auth</code>	Messages generated by the authorization system (for example: <code>login</code> , <code>su</code> , and <code>getty</code> ).
<code>lpr</code>	Messages generated by the line printer spooling system (for example: <code>lpr</code> , <code>lpc</code> , and <code>lpd</code> ).



Facility	Description
local0	Reserved for local use, along with local1 to local7.
mark	Receives a message of priority <code>info</code> every 20 minutes, unless a different interval is specified with the <code>syslogd -m</code> option.
msgbuf	Kernel syslog message buffer recovered from a system crash. The <code>savecore</code> command and the <code>syslogd</code> daemon use the <code>msgbuf</code> 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	An error message.
warning or warn	A warning message.
notice	A Condition that is not an error conditions, but is handled as a special case.
info	An informational message.
debug	A message containing information that is used to debug a program.
none	A way to disable 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: <code>kern.log</code> , <code>mail.log</code> , or <code>lpr.log</code> ).
Host name preceded by an at sign (@)	Forwards messages to the <code>syslogd</code> daemon on the specified host. Messages will not be forwarded if the <code>-R</code> option is specified when the <code>syslogd</code> daemon is started. See Section 12.2.2 for more information.

Destination	Description
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 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 specified in the `/etc/syslog.conf` file.

The `syslogd` daemon automatically creates a new *date* directory every 24 hours, when you boot the system, or when the `syslogd` daemon is restarted or reconfigured. You can get the latest logs from the `/var/adm/syslog.dated/current` directory. The `current` directory is a symbolic link to the latest *date* directory.

For example, to create a daily log file of all mail messages of level `info` or higher, edit the `/etc/syslog.conf` file and include a line similar to the following:

```
mail.info /var/adm/syslog.dated/mail.log
```

If you specify the previous line in the `/etc/syslog.conf`, the `syslogd` daemon creates the following daily directory and file:

```
/var/adm/syslog.dated/11-Jan-12:10/mail.log
```

### 12.2.1.2 Configuring syslog to Use EVM

By default, `syslogd` is configured with the `-e` option to forward events to EVM. (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 forwards all emergency events to EVM 3
kern.info+ 4
user.notice+
mail.notice+
daemon.notice+
auth.notice+
syslog.notice+
```

---

1 The first part of each line item specifies which facility generated the message, such as `kern` for kernel. An asterisk (\*) indicates that all facilities are selected. In this case, `*.emerg` ensures that all messages of emergency priority are forwarded to EVM.

You can choose which events are forwarded by creating an entry for a facility, or removing an existing entry. Entries are based on the keywords in the facility table in Section 12.2.1.1.

2 The second part of each item specifies the priority of messages, based on the keywords in the severity level table in Section 12.2.1.1.

3 You can add comments, preceded by a number sign (#). However, you cannot mix forwarding entries and comments in the same line

4 The plus sign (+) appended to a priority indicates that the specified priority and all higher priority messages are forwarded. If you want to choose individual severity levels for a facility (such as warning, critical and emergency, create a line for each priority.

Events are posted with the EVM name of `sys.unix.syslog.facility`.

For more information, refer to `syslog_evm.conf(4)` and Chapter 13.

#### 12.2.1.3 Editing the `binlog.conf` File

If you want the `binlogd` daemon to use a configuration file other than the default, specify the file name with the `binlogd -f config_file` command. The `binlogd` daemon forwards all events to EVM. You can filter and select `binlog` events using EVM utilities, as described in Chapter 13.

You can forward `binlogd` events to a remote host. Refer to the `binlogd(8)` for information on the remote logging options. The `-R` and `-r` options are important because you use them to control the creation of an inet port for remote access.

The following is an example of a `/etc/binlog.conf` file:

```
#
# binlogd configuration file
#
# format of a line:  event_code.priority      destination
#
# where:
# event_code - see codes in binlog.h and man page, * = all events
# priority   - severe, high, low, * = all priorities
# destination - local file pathname or remote system hostname
#
#
*. *    /usr/adm/binary.errlog
dumpfile /usr/adm/crash/binlogdumpfile
102.high /usr/adm/disk.errlog
```

Each entry in the `/etc/binlog.conf` file, except the `dumpfile` event class entry, contains three fields:

- 1 Specifies the event class code that indicates the part of the system generating the event.
- 2 Specifies the severity level of the event. Do not specify a severity level if you specify `dumpfile` for an event class.
- 3 Specifies the destination where the binary event records are logged.

The `binlogd` daemon ignores blank lines and lines that begin with a number sign (`#`). You can specify `#` as the first character in a line to include comments in the file or to disable an entry.

The event class and severity level are separated from the destination by one or more tab characters or spaces.

You can specify the following event class codes:

Class Code	General
*	Specifies all event classes.
dumpfile	Specifies the recovery of the kernel binary event log buffer from a crash dump. A severity level cannot be specified.

<b>Class Code</b>	<b>Hardware-Detected Events</b>
100	CPU machine checks and exceptions, or generalized exception fault
101	Memory
102	Disk
103	Tape
104	Device controller
105	Adapter
106	Bus
107	Stray interrupt
108	Console event
109	Stack dump
110	Generalized machine state
113	Double error halt
115	(Un)correctable environmental
120	Reporting of correctables disabled
195	StorageWorks Command Console (SWCC)
196	I2O block storage
198	SWXCR RAID controller
199	SCSI CAM

<b>Class Code</b>	<b>Software-Detected Events</b>
201	CI port-to-port-driver
202	System communications services
203	LSM note
204	LSM warning
205	LSM continuation
206	AdvFS domain panic

<b>Class Code</b>	<b>Informational ASCII Messages</b>
250	Generic informational ASCII message

Class Code	Operational Events
300	Startup ASCII message
301	Shutdown ASCII message
302	ASCII Panic message
310	Time stamp
350	Diagnostic status ASCII message
351	Repair and maintenance ASCII message
400	Filterlog event. (Use only with filterlog)

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 <code>binlogd</code> daemon appends the binary event records.
@ <i>hostname</i>	Specifies the name of the host, preceded by an at sign (@), to which the <code>binlogd</code> daemon forwards the binary event records. If you specify <code>dumpfile</code> for an event class, you cannot forward records to a host. Operational timestamp (310) events are not forwarded automatically.

## 12.2.2 syslog Security and Remote Messages

Unless the domain host name of a remote host is entered in the local `/etc/syslog.auth` file, the local system will not log any `syslog` messages from that remote host. If you intend to make `syslogd` secure on your system, and you have configured or intend to configure other hosts to forward `syslog` messages to the system, complete the following steps:

1. Use the `su` command to become the superuser (root).

2. Create the `/etc/syslog.auth` file using a text editor. 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 the `/etc/syslog.auth` file. (Lines beginning with the # character are comments and are ignored.)
  - A host name must be a complete domain name such as `trout.fin.huk.com`.
  - If a domain host name is given, it must either appear in the local `/etc/hosts` file or the local system must resolve it through a name server (such as BIND).
  - A host name can have at most as many characters as defined by the `MAXHOSTNAMELEN` constant in the `/sys/include/sys/param.h` file, although each line in the `/etc/syslog.auth` file is limited to 512 characters.

A plus sign (+) by itself allows event forwarding from all hosts. A host name can also be preceded by a minus sign (-) to expressly prohibit that host from forwarding events. If the `/etc/syslog.auth` file is not present on the system, then forwarding from all hosts is enabled.

Specify the `-R` option when starting the daemon if you do not want the `syslogd` daemon to create an `inet` port to listen for events being sent by remote hosts. To make this the default mode of operation, edit the startup command line in the `/sbin/init.d/syslog` file. Using the `-R` option also means that the `syslogd` daemon cannot forward events to other systems.

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 MAKEDEV(8) 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.

### 12.2.4.1 The `syslogd` Daemon

You must ensure that the `init` program starts `syslogd` daemon. If the `syslogd` daemon does not start, or if you want to specify options with the command that starts the `syslogd` daemon, you must edit the `/sbin/init.d/syslog` file. When you edit the file, you must either include or modify the `syslogd` command line. You can also invoke the command manually.

The command that starts the `syslogd` daemon has the following syntax:

```
/usr/sbin/syslogd [-b rcvbufsz] [-d] [-e | -E] [-f config_file] [-m  
mark_interval] [-p path] [-r | -R] [-s]
```

The initialization of the daemon uses only the `-e` option by default. The `-e` option configures the daemon to 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 `syslogd(8)` 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 (UDP) socket, which is specified in the `/etc/services` file. For security reasons, you might want to either disable this socket using the `-R` option or specify authorized hosts in the `/etc/syslog.conf` file.



- The device special `/dev/klog` file, which logs only kernel messages.

Messages from other programs use the `openlog`, `syslog`, and `closelog` calls.

When the `syslogd` daemon is started, it creates the `/var/run/syslog.pid` file, where the `syslogd` daemon stores its process identification number. Use the process identification number to stop the `syslogd` daemon before you shut down the system.

During normal system operation, the `syslogd` daemon is called if data is put in the kernel `syslog` message buffer, located in physical memory. The `syslogd` daemon reads the `/dev/klog` file and gets a copy of the kernel `syslog` message buffer. The `syslogd` daemon starts at the beginning of the buffer and sequentially processes each message that it finds. Each message is prefixed by facility and priority codes, which are the same as those specified in the `/etc/syslog.conf` file. The `syslogd` daemon then sends the messages to the destinations specified in the file.

To stop the `syslogd` event-logging daemon, use the following command:

```
# kill `cat /var/run/syslog.pid`
```

Using the following command, you can apply changes to the `/etc/syslog.conf` configuration file without restarting the daemon:

```
# kill -HUP `cat /var/run/syslog.pid`
```

#### 12.2.4.2 The `binlogd` Daemon

You must ensure that the `init` program starts the `binlogd` daemon. If the `binlogd` daemon does not start, or if you want to specify options with the command that starts the `binlogd` daemon, 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 `binlogd` command supports the following options

```
/usr/sbin/syslogd [-d] [-f config_file] [-r | -R]
```

Refer to `binlogd(8)` for information on command options.

The `binlogd` daemon reads binary event records from the following:

- An Internet domain socket (`binlogd, 706/udp`), which is specified in the `/etc/services` file. For security reasons, you might want to either disable this socket using the `-R` option. You can also specify authorized hosts in the `/etc/binlog.conf` 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, you can apply changes to the `/etc/binlog.conf` configuration file without restarting the daemon:

```
# 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 kB (8192 bytes) and 1 MB (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_ASCIIION` 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 Managing Log Files

On a well maintained system, the size of the various log files should not become a problem as you will:

- Carefully select only those events that you want to log
- Monitor the logs for error conditions that result in many postings
- Regularly archive and back up your important event logs

The `/var/spool/cron/crontabs/root` file contains the following model entry for managing log files:

```
0 2 * * 0 /usr/sbin/logclean /var/adm/wtmp > /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 that cleans up the older logs in the `/var/adm/syslog.dated` directory:

```
5 1 * * * find /var/adm/syslog.dated -type d -mtime +5 -exec rm -rf '{}' \;
```

This entry causes all directories under the `/var/adm/syslog.dated` directory that were modified more than 5 days ago to be deleted, along with their contents, every day at 1:05. Refer to Chapter 3 and `crontab(1)` for more information.

## 12.5 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 bytes, for systems with up to 128 MB of physical memory. For systems with greater than 128 MB of physical memory, the value of `msgbuf_size` is calculated and set automatically at 64 bytes for every 1MB of memory. For example, in a system with 512 MB the value is  $512 * 64 = 32,768$ , which is equivalent to 32K bytes.

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. Click 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.



---

## 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 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.
- Section 13.4 describes how to troubleshoot common EVM problems.

### 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 ready to take action when certain unusual conditions occur, such as when a disk fills or a processor reports hardware errors. It is also important to verify that certain routine tasks run successfully each day, and to review certain system configuration values. Such conditions or task completions are described as system events.

An event is an indication that something interesting has occurred – an action has been taken, some condition has been met, or it is time to confirm that an application is still operational. A particular event 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. Examples of event channels are:

- Log files, where messages are stored in a file that is usually in ASCII text format
- Event management systems
- Programs that you run to obtain a snapshot of status information

An event management system is an active event channel and as such, it provides services for distributing, storing, and retrieving event information.

The operating system supports a number of channels through which system components can report event and status information. You must check the information available at each channel regularly to be sure that the system is operating normally. The system logger, `syslog`, and the binary error logger, `binlog`, are familiar examples of event management systems. They provide simple event distribution facilities for other components to use, and their daemons actively manage the event information they receive. By contrast, the `cron` daemon's log file, `/var/adm/cron/log`, is an example of a passive event channel. The `cron` daemon writes new event information to the end of its file, and takes no special action to notify interested entities when it does so.

Apart from `syslog` and `binlog`, there are several other log files stored in various locations on the system. To facilitate management of these log files, 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. The EVM viewing facilities include a graphical event viewer, which is integrated with the SysMan Menu and SysMan Station, and a full set of command-line utilities, which enable you to filter, sort, and format events as needed. You can configure EVM to automatically notify you (or other system entities) 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, with EVM the existing channels are much more accessible.

EVM provides the following features:

- Facilities for users and applications to post and monitor events
- Support for all 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 or detailed event data, including online explanations
- Full set of command-line utilities that you can use to post and handle events from shell scripts and from the command line
- Configurable event logger that allows full control over which events are logged and optimizes storage space used by identical events
- Configurable event forwarding that enables you to automatically notify other system entities of selected events
- Log file management that automatically archives and purges log files daily.
- Support for the application programming interface (API) library
- Centralized access to event information
- Configurable authorization for posting or accessing events

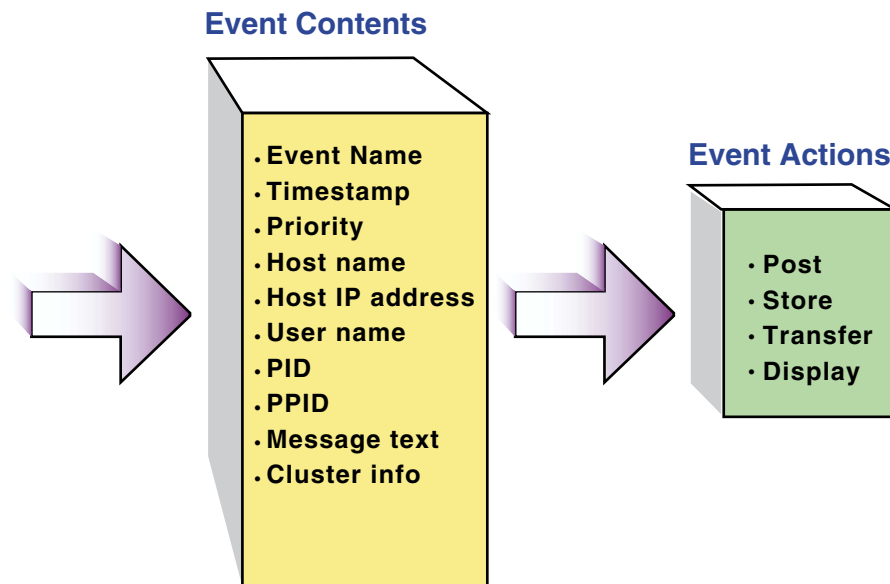
### 13.1.1 Understanding EVM Events

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. An event might also contain 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 path name 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 receiving process can then extract and process the information contained in the event.

Although the EVM logger captures posted events and stores them in a system log file, you can easily capture your own set of events and store them in your own file for later analysis. You use the `evmwatch` monitoring utility, or reconfigure the logger to capture your own events.

Figure 13–1 shows a graphical representation of an event.

**Figure 13–1: Event Model**



ZK-1549U-AI

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 possible actions performed on any event.

Because an event is a package of binary data, you cannot view it directly with a text viewer such as the `more` command. However, EVM includes command-line utilities which understand the format of the event, and which you use to perform basic operations at the command prompt or in shell scripts. You can use EVM commands to:

- Retrieve events from storage, sort them into a preferred order, and format them for display
- Watch for new events being posted
- Post new events

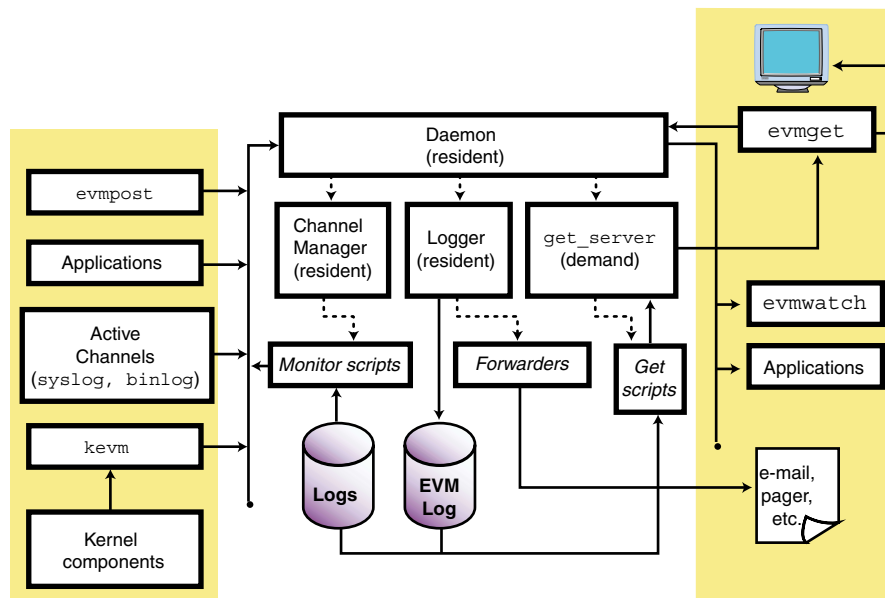
The EVM utilities are designed to be used together in pipelines. For example, you might pipe a set of events from a file into the `evmsort` utility, pipe the output into the `evmshow` formatting utility, then pipe the output of that command into the `more` command, or redirect it to a file. Section 13.3 provides examples of using EVM commands to monitor and review event activity.

Once the event file is 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 the `sort -u` and `wc -l` commands 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**



ZK-1371U-AI

In Figure 13–2, client components involved in posting events are shown at the left, EVM system components in the center, and client components involved in subscribing to and retrieving of events are at the right. Active event channels post events directly to 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 you boot the system, 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 starts 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 establishes 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, `evmllogger`, runs as a resident process. It is configured to subscribe to a selected set of events, and to store them in managed log files for later retrieval. The logger is also configured by default to:

- Write high-priority events to the system console
- Send mail to the system administrator when high-priority events occur

The resident channel manager process, `evmchmgr`, is configured to run periodic channel-monitoring scripts, which post events when they detect noteworthy activity in the channel. The channel manager also runs the daily log cleanup functions.

The get server process, `evmget_srv`, is a transient (demand) process that executes event retrieval scripts for the various event channels. The `evmd` daemon runs an instance of `evmget_srv` whenever a user runs the `evmget` command.

Entities on the left side of the model create posting connections to the daemon in order to post events. After it receives events from the posters, the daemon merges them with corresponding event templates from its template database, and distributes them to its subscribing clients.

On the right side of the model, `evmwatch` and other application programs that need to receive event information as it happens create subscribing connections to the daemon, and pass filter strings to it to specify their event subscriptions. The `evmget` command, which a user can run to retrieve historical event information from log files, creates a service connection, and passes a filter string to specify the set of events to be retrieved. The daemon then runs an instance of the get server to handle the request. The e-mail and pager actions are examples of forwarding commands, which the logger might execute 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. Refer to Section 13.3 for examples of how to use these commands to monitor and review event activity.

**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 EVM administrative commands which are normally invoked during system initialization. The individual command reference pages discuss other conditions under which the command is used.

**Table 13–2: EVM Administrative Utilities**

Command	Description
evmchmgr	The EVM daemon automatically starts the EVM channel manager. Do not start it manually. This command executes the periodic functions defined for any channel.
evmd	The EVM daemon receives events from posting clients and distributes them to subscribing clients. (A subscribing client is one that has indicated that it wants to receive the events.) The daemon is a critical system facility that starts automatically at system boot. Do not terminate it.
evmlogger	The EVM daemon automatically starts the EVM logger. The logger receives events from the daemon and writes them to each of the logs whose filter string they match. The <code>evmlogger</code> also serves as an event forwarding agent that you can configure to take an action when required.

**Table 13–2: EVM Administrative Utilities (cont.)**

Command	Description
evmreload	This command posts control events, which instruct EVM components to reload their configuration files. When you modify an EVM configuration file you must use this command to load the new configuration.
evmstart	This command starts the EVM daemon. It is intended for use by the system startup scripts, but you can also use it to restart EVM if it terminates for any reason.
evmstop	This command stops the EVM daemon, preventing entities from posting or subscribing for events. It is intended for use by the system shutdown scripts. Because EVM is required for many system functions to operate correctly, you will not use this command 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*. (See EVM(5) for a list of individual API reference pages.)

### 13.1.2.3 EVM System Files

EVM creates or uses the following system files:

- Executable files for EVM administrative commands are located in the `/usr/sbin` directory.
- Command executable files are located in the `/usr/bin` directory.
- Initialization files are located in the `/sbin/init.d` directory.
- Configuration files are located in the `/etc` directory 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 `evmdaemon.conf(4)` for a complete description of this file.
  - The `/etc/evmchannel.conf` file is the event channel configuration file, read by the channel manager, `evmchmgr`, and the `evmshow` command. This file describes all the channels through which events can be posted and retrieved. Refer to Section 13.2.1.2 and `evmchannel.conf(4)` 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 `evmlogger.conf(4)` 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 `evm.auth(4)` 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.lock`. 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 `yyymmdd` is the date of the log, and `_nn` is a sequential generation number. A new log generation starts if the log reaches its configured maximum size during the course of the day, or if the logger finds an error in the current file. The day's first log file has no generation number. A new log file is started automatically when it receives the first event after midnight, system time.  
  
This directory also contains a lock file, `evmlog.dated.lock`, 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 starts. 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 that holds 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), that is saved by the `evmd` daemon for future actions, such as stopping EVM.
- The `/var/run/evmlogger.info` file contains the logger's PID and information about the log files being managed. The `evmlog` channel retrieval and daily cleanup functions use this information.
- System-supplied definition files for templates, channels, and filters are located in the following subdirectories of the `/usr/share/evm` directory. Do not modify these files:
  - 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 and subdirectories.

### 13.1.3 Related Utilities

The following subsystems or optional components also provide event handling capabilities:

- System logger (`syslogd`)

The system logger logs text messages on behalf of the kernel and many user-level system components. In addition to storing events in its own log files, the default configuration of the `syslogd` daemon forwards selected events to 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 `syslogd(8)` for more information.

- Binary error logger (`binlogd`)

The binary error logger logs system errors and configuration information in binary format. Events are translated by the DECEvent translation facility (`dia`), or by Compaq Analyze (`ca`) depending on the system type. In addition to storing events in its own log files and distributing them to its own clients, the `binlogd` daemon forwards events to EVM for further distribution. EVM retrieves binary error log events from storage through the `binlog` event channel functions. Refer to `binlogd(8)` for more information.



- DECEvent and Compaq Analyze

DECEvent is a rules-based translation and reporting utility that provides event translation for binary error log events. EVM uses DECEvent's translation facility, `dia`, to translate binary error log events into human-readable form. Refer to `dia(8)` for more information. Compaq Analyze performs a similar role on most EV6 series processors. Refer to the Compaq Analyze documentation and `ca(8)` 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 are suitable for most installations. However, you can change the configuration for your system if, for example, an event channel is to be added or modified, the log file archive and expiration options need to be changed, 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 must be reestablished by running the `evmreload` command. See `evmreload(8)` for information on this command.

Configuration files are described in the following sections and in the corresponding reference pages.

### 13.2.1.1 EVM Daemon Configuration

The EVM daemon reads the `/etc/evmdaemon.conf` configuration file at system startup and whenever you issue a reload request by using the `evmreload` command. For a complete description of the contents and syntax of the configuration file, see the `evmdaemon.conf(4)` reference page. Example 13–1 shows some sample entries in the EVM 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.

- ❷ These commands start the `evmllogger` and the `evmchmgr` components as synchronized clients, ensuring that both clients complete their subscription requests before the daemon accepts any events from posting clients. The command-line options for these commands define the clients' log files and, in the case of the logger, an output file that is used to make operational details available to the `evmlog` event channel functions.
- ❸ These statements define the `event_get` event retrieval service, which the `evmget` command uses to retrieve events.
- ❹ 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. Activity monitoring (counting of events) is then suspended for the hold-off period of four hours (240 minutes).
- ❺ This line sets the `remote_connection` to `false` to disable connection to this system by remote EVM clients. Refer to `evmdaemon.conf(4)` 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 the `evmreload` command (8) 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, the `evmshow` command, and 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 file entries.

#### Example 13–2: Sample EVM Channel Configuration File

---

```
# Global path for channel functions
path /usr/share/evm/channels ❶

# Time-of-day at which daily cleanup function will run
cleanup_time 02:00:00 ❷

# =====
# Event channel: EVM log
# =====
channel
{ ❸
    name evmlog ❹
```

### Example 13–2: Sample EVM Channel Configuration File (cont.)

---

```
path          /usr/share/evm/channels/evmlog    5
events       *                       6
fn_get       "evmlog_get"           7
fn_details   "evmlog_details"
fn_explain   "evmlog_explain"
fn_monitor   "evmlog_mon"
fn_cleanup   "evmlog_cleanup 7 31"  8
mon_period   15:00 # Monitor every 15 minutes 9
}
```

---

- 1 This line declares the `/usr/share/evm/channels` directory as the default path for all channel functions. This path is prefixed to the names of any channel functions defined in this file that do not begin with a slash (`/`) character, unless the channel group supplies its own path value.
- 2 This line defines a daily 2:00 am cleanup for all channels.
- 3 This line specifies a configuration group that defines an event channel.
- 4 This line specifies that the name of the channel is `evmlog`.
- 5 This line overrides the default path `/usr/share/evm/channels` defined at the global level.
- 6 In this line, the asterisk (`*`) indicates that the channel provides default event handling, meaning that its functions are invoked to provide details and explanations for any events whose names do not match the `events` value of any other channel.
- 7 Any line beginning with `fn_` defines a script that runs for each function.
- 8 The argument values on this line are passed to the cleanup program to control its operation. In this example, log files older than 7 days are compressed and those older than 31 days are deleted. The meanings of the arguments are specific to individual programs, and may not be the same in all cases.
- 9 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 the `/etc/evmllogger.conf` configuration file. 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 is 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
    maxqueue  200 11

# Don't forward mail events through mail
    filter    "[prio >= 600] & ![name @SYS_VP@.syslog.mail]" 12

    suppress 13
  {
    filter    "[name *]"
    period    120 # minutes
    threshold 1 # No. of duplicates before suppression
  }
}

# This evmshow command writes a subject line as the
# first line of output, followed by a detailed display
# of the contents of the event. The resulting message is
# distributed by mail(1).
  command    "evmshow -d -t 'Subject: EVM ALERT [@priority]: @@" | \
  mail root" 14

# Secondary configuration files can be placed in the following
# directory. See the evmlogger.conf(5) reference page for
# information about secondary configuration files.
```

### Example 13–3: Sample EVM Logger Configuration File Entries (cont.)

---

```
configdir      /var/evm/adm/config/logger
```

---

- ❶ This line specifies the configuration group that defines an event log.
- ❷ This line specifies that the `evmlog` event channel uses the name to obtain configuration information about the log.
- ❸ This line specifies that the log files are stored in the `/var/evm/evmlog` directory. Each day, when the log for that day is first written, the dated suffix is replaced by the date in the format `yyyymmdd`.
- ❹ This line specifies that the `type` of events written to this log are binary EVM events, rather than formatted (ASCII text) events.
- ❺ This line specifies the maximum size of the log file in kilobytes (KB). In this case, if the size of the current log file exceeds 512 KB the logger closes it and begins a new log file, with a sequentially numbered suffix (for example, `_2`) appended to the file name.
- ❻ If this line is not commented out (by `#`) and the sample path is replaced by the path name of an existing write-enabled directory, an alternate log file is opened in this directory if the primary directory becomes write-disabled.
- ❼ 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.
- ❽ 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 `evmlogger.conf(4)` for a detailed description of event suppression.
- ❾ 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.
- ❿ In this line, `name` identifies the forwarder.
- ⓫ The `maxqueue queue_limit` keyword limits the number of events that a forwarder can queue while a previous event is being handled. If the maximum number of events is already queued when a new event arrives, the event is ignored by this forwarder. If not specified, this keyword has a default value of 100 events. If you specify a value greater than 1000 events, the logger automatically limits it to 1000 events.

- 12** This line establishes filtering for the events. As with an event log definition, the filter string specifies the set of events that are handled by this forwarder. To prevent an event loop from occurring if the mailer posts high-priority events, signifying a possible problem in the mail subsystem, mail events are explicitly excluded from this forwarder.
- 13** These lines suppress multiple forwarding of events. The suppression mechanism for a forwarder is similar to that for an event log. Here, the purpose is to prevent the command from being sent multiple times in a short period due to the same event being posted repeatedly. In the example, a particular event is forwarded once every two hours.
- 14** This line defines the command that executes when an event is handled by the forwarder. The event is piped into the command's `stdin` stream. The result of this command is shown in the comments preceding the command line.

If you make any changes to the logger configuration file you must run the `evmreload` command to make the changes known to the logger.

#### 13.2.1.4 Secondary Logger Configuration Files

Secondary logger configuration files enable you to add event logs or forwarders without modifying the primary configuration file, `/etc/evmlogger.conf`. This feature ensures that any problems with secondary files do not affect the primary configuration. It enables you to safely experiment with different logger configurations. Should the logger encounter a syntax error in a secondary configuration file, it displays an error message and rejects the file. The primary configuration file and any additional (and correct) secondary files are processed and EVM will function correctly. The secondary configuration directory feature also allows individual system components, products and applications to install or change logfiles and forwarders by installing or replacing files, rather than having to insert or maintain lines in the primary configuration file. You can uninstall entries by removing the file.

The default and recommended location of secondary configuration files is the `/var/evm/adm/config/logger` directory, or a subdirectory of that directory. You can also place the configuration file elsewhere and create a symbolic link to it from the default directory. Although supported, it is recommended that you avoid adding `configdir` lines to the primary configuration file. Your secondary configuration files must have file name suffix `.conf` and the file syntax must follow the rules stated in Section 13.2.1.3.

It is important that you give appropriate permissions to the secondary logger configuration files and directories. The logger runs with superuser privileges and can execute commands specified in any secondary configuration file. For

this reason, the logger rejects any configuration files that do not have the correct permissions and posts a warning event. See `evmllogger.conf(4)` for the correct file permissions.

In a cluster environment, the logger configuration files are shared by all the cluster members. If you require a member-specific event log or forwarder, you can specify it in a secondary configuration file. Create a context-dependent symbolic link (CDSL) in the secondary configuration directory to reference the file. See `mkcdsl(8)` for instructions on creating a CDSL.

## 13.2.2 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, event information security is maintained 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. . In a cluster, users requesting a connection from another node of the same cluster are also authenticated. 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`.

The root user can authorize individual users or groups of users to do the following:

- Post selected events



- Access (subscribe to or retrieve from storage) selected events
- Execute selected services

By default, all events are protected. Event rights are granted by supplying, for each event class, a list of users who have the specified right or who are explicitly denied rights. A plus sign (+) that is not followed by a user list implicitly grants the right to all users. A minus sign (-) that is not followed by a user list implicitly denies the right to all users. The root user has implicit posting and access rights to all events unless explicitly denied them. Example 13–4 shows sample entries in an authorization file. See `evm.auth(4)` for 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 Only the root user can post the class of events that have names beginning with `sys.unix.evm.control`. Such events are accessible

by all users. The `@SYS_VP@` entry is a macro that is replaced with `sys.unix` when the file is read.

- ❷ Only the root user can post the class of events that have names beginning with `sys.unix.evm.msg.admin`. Such events can be accessed by root or other users in the `admin` group.
- ❸ All users can post or access the class of events that have names beginning with `sys.unix.evm.msg.user`.
- ❹ 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

EVM is accessible to clients that are running on remote systems, allowing you to monitor and retrieve events from a central system. You can make a remote connection by specifying a host name or IP address by using the `-h` option with the `evmwatch`, `evmget` and `evmpost` command-line utilities. Alternatively, you can specify a remote host name in the event viewer's Get Events From... dialog box. See Section 13.3.11 for information on the event viewer.

Because EVM currently has no means of authenticating remote clients, it imposes the following restrictions to avoid compromising security:

- By default, remote access is disabled in the daemon's configuration file, `/etc/evmdaemon.conf`. To enable remote access, edit this file and set the value of the `remote_connection` keyword to `true`, then run the `evmreload` command specifying the `-d` option.  
  
Enable remote access only if your system is running in a fully secure environment.
- The EVM daemon grants remote clients only the lowest level of access or posting privileges. This means that remote clients can access or post only events that any local client can post or access.

### 13.2.3 Managing Log Files

The EVM channel manager, `evmchmgr`, provides log management capability through the channel `fn_cleanup` function. You can define this capability for any channel through the channel configuration file, `evmchannel.conf`. See Section 13.2.1.2 for additional information on this file.

By default, channel cleanup functions run when EVM starts, and then run at 2:00 am each day. You can change the time of day by editing the `cleanup_time` value in the channel configuration file. When a cleanup is

scheduled, the channel manager scans the event channel list, and executes the `fn_cleanup` command for each channel identified in the file.

The `evmlog` cleanup function, `evmlog_cleanup`, takes two arguments:

- The archive period, which has a default value of 7 days.
- The delete period, which has a default value of 31 days.

The function uses the `find` utility to locate and compress (`zip`) all logs older than the archive period, and to delete any archived files older than the delete period. You can change the period values by editing the function definition in the channel configuration file. Setting either of these values to zero disables the corresponding function. You can compress logs for archiving by using the `gzip` command. See `gzip(1)` for more information.

The default channel configuration also provides a similar cleanup function for the SysMan Station message log files, through the `misclog` event channel. You can manage the `syslog` and binary error log channels by using entries in the `crontab` file. 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, review the log entries and, if necessary, use the cleanup options in `binlogd` to initiate a cleanup. See `binlogd(8)` for more information.

The `evmget` command does not retrieve `evmlog` events that are stored in archived (zipped) logs. To retrieve events from archived logs you must first uncompress them with the `gunzip` command. See `gunzip(1)` for information on unzipping archive files.

## 13.2.4 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 the event will 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`. If you have installation-specific or third-party event templates, load them as follows:

1. Create an appropriately-named subdirectory of the local template directory, `/var/evm/adm/templates`, and copy the event templates into it.

2. Run the `evmreload` command, specifying the `-d` option to signal the EVM daemon to reload its internal template database.

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.

To be recognized by EVM, template files require specific ownership and permissions. See `evmtemplate(4)` for details. Refer to the *Programmer's Guide* for more information on installing 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 users.

Add new event templates as follows:

1. Create new template files as described in Section 13.2.4.
2. Copy the template files to the `/var/evm/adm/templates` directory or to a subdirectory.
3. Run the `evmreload` command, specifying the `-d` option, to signal the EVM daemon to reload its internal template database.

See `evmtemplate(4)` 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 that are stored in the system's event log file and to analyze events from other systems, including other operating systems such as OpenVMS and Windows NT.

Although the EVM infrastructure directly recognizes events only in its own EVM format, events are posted through other channels, such as the `binlogd` daemon. These events can be passed to EVM within a wrapper EVM event by inserting the lower-level event into the EVM event as variable data. The whole package is then 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 the `binlogd` daemon receives an event from the operating system it first stores the event in its own log file and distributes it to its own clients. It then creates an EVM event named `sys.unix.binlog`, and adds a variable called `binlog_event`, which contains 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 is 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
```

### Example 13–5: A binlogd Event Showing the DECEvent Translation (cont.)

---

```
Timestamp           : 26-Jan-2000 20:54:36
Host IP address     : 16.69.224.11
Host Name           : kopper
Format              : System shutdown msg: $message
Reference           : cat:evmexp.cat:300
Variable Items:
  subid_class = 301
  message = "System rebooted by root:"
  binlog_event = [OPAQUE VALUE: 96 bytes]
===== Translation =====
DECEvent version: V3.2

Logging OS                2. operating system
System Architecture       2. Alpha
Event sequence number     752.
Timestamp of occurrence   26-JAN-2000 20:54:36
Host name                 kopper
System type register      x0000000F AlphaStation 600 or 500
Number of CPUs (mpnum)    x00000001
CPU logging event (mperr) x00000000
Event validity            1. O/S claims event is valid
Event severity            5. Low Priority
Entry type                301. Shutdown ASCII Message Type
SWI Minor class          9. ASCII Message
SWI Minor sub class      2. Shutdown
ASCII Message             System rebooted by root:
=====
```

---

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 will 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 need to use a remote Compaq Analyze server to do translations, you must edit the local channel configuration file, as described below.
- Recent processors produce `binlogd` events with a new header format that differs from the format produced by earlier platforms. The newer format events are known as Common Event Header (CEH) events. If your system does not produce CEH events you cannot use Compaq Analyze to translate them, and you must install the DECEvent formatter utility, `/usr/sbin/dia`.

If your system will use DECEvent or will use a Compaq Analyze server running on the local system for `binlogd` event translation, you do not need to change the standard configuration. If you plan to use a Compaq Analyze server running on a remote system, you 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 attempts 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 choose 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 is broken at the backslash (`\`) to fit the page, and appears as a single line in the file).

```
fn_details      "binlog_details -decevent -ca localhost -ca gandalf \  
-ca tigger"
```

After you edit the configuration file, run the `evmreload -c` command to make the EVM channel manager aware that the file is updated.

EVM does not start the Compaq Analyze server; it must already be running on the selected system for the translation to succeed. The server usually starts automatically when the system is initialized. For more information, log on to a system that has Compaq Analyze installed and refer to `desta(8)`.

See Section 13.4 for procedures that enable you to determine 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 the root user when events with a priority of 600 (alert) or greater are posted. You should review the full event log on a daily basis by using the event viewer or command-line utilities. You can configure the logger to take other actions, such as sending a pager message according to any criteria you choose. You can monitor events at your terminal as they occur by using the `evmwatch` command.

The following sections illustrate the commands you can use to monitor and review event activity. As you become familiar with the EVM command set, you will build up a set of favorite commands, shell scripts, and filters that will help you to keep track of what is happening on your system.

### 13.3.1 Displaying Events Using `evmshow`

Because an EVM event is a binary data package, it must be converted to text form before you can display it on a terminal. The `evmshow` command reads binary EVM events from its `stdin` stream or from a named file, and outputs the same events in text form to `stdout`. For example, you might display the contents of a file containing EVM events by using the following command:

```
# cat my_events | evmshow | more
```

This command displays the events from the log file in the default manner, meaning that it takes the format data item from each event, expands it with the values of any variables it references, and displays it. References to variables are identified by a dollar sign (`$`). Therefore, if the `my_events` file contains an event with a format data item of `AdvFS: AdvFS domain $domain is full`, and the event also contains a variable named `domain` with a value of `root_domain`, the corresponding line of the output is:

```
AdvFS: AdvFS domain root_domain is full
```

This information tells you what happened, but not when it happened, or the importance of the event. You can modify the output of the `evmshow` command to include any of the data items in the event, including its timestamp and priority, by using the `-t` option to specify a show-template. A show-template is a text string that indicates which data items you want to be displayed for an event, and how you want them to be displayed.

The following example illustrates the use of a show-template to display an event with a timestamp, a priority, and the formatted event message. In the show-template, the names of the items to be displayed are each preceded by an at sign (`@`). Two at signs (`@@`) indicate that the event's format item should



be expanded and displayed. The second line shows the output for the domain full event. In the output, the event priority is surrounded by brackets, and there are two spaces before the message text, exactly as specified in the show-template:

```
# cat my_events | evmshow -t "@timestamp [@priority] @@" | more
22-Jun-2000 11:22:27 [600] AdvFS: AdvFS domain root_domain is full
```

You can set up your own show-template to display the items that are important to you, in any format you want. See EvmEvent(5) for a list of all the data items. After you determine your preferred style you can set a default show-template in the environment variable EVM\_SHOW\_TEMPLATE and use fewer keystrokes at the command line. The following Korn shell (ksh) commands are equivalent to those in the previous example:

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority] @@"
# cat my_events | evmshow | more
```

If you want more information about an event you can request a detailed display, including an explanation and a full dump of its contents, by using the evmshow command with the -d option. The following example shows a detailed display of the AdvFS domain full event:

```
# cat my_events | evmshow -d | more
===== EVM Log event =====
EVM event name: sys.unix.fs.advfs.fdmn.full

    This event is posted by the AdvFS filesystem to provide
    notification that the specified AdvFS domain is full. No more
    space is available for writing. [1]
=====

Formatted Message:
    AdvFS: AdvFS domain root_domain is full [2]

Event Data Items: [3]
    Event Name      : sys.unix.fs.advfs.fdmn.full
    Cluster Event   : True
    Priority         : 600
    PID             : 1177
    PPID            : 724
    Timestamp       : 22-Jun-2000 11:22:27
    Host IP address : 0.0.0.0
    Host Name       : x.x.compaq.com
    User Name       : root
    Format          : AdvFS: AdvFS domain $domain is full [4]
    Reference       : cat:evmexp.cat:450

Variable Items: [5]
    domain (STRING) = "root_domain"
=====
```

[1] The explanation of the event. In some cases, this data field contains a recommended action to rectify a problem.

- 2 The Formatted Message section.
- 3 The Event Data Items section, which lists all of the standard data items contained in the event. See `EvmEvent(5)` for a description of each of these items.

The items shown here are typical of many events, but sometimes some of these are missing, and occasionally you might see additional items. For example, most events are not distributed across all nodes of a cluster, and so in most cases the Cluster Event item is not displayed.

- 4 The Format data item is almost the same as the content of the Formatted Message data item, but it includes a reference to a variable called *domain*, indicated by the `$` symbol preceding it.
- 5 The Variable Items section, which contains the value of the domain variable.

Refer to Section 13.3.12.2 for information on how to select events for detailed display.

You can use the `evmshow -x` command to display the explanation alone. Alternatively, use the `-x` and `-t` options together to provide a summary of the event followed immediately by its explanation. For example:

```
#cat my_events | evmshow -x -t "@timestamp
[@priority]  @" | more
22-Jun-2000 11:22:27 [600] AdvFS: AdvFS domain root_domain is full
  This event is posted by the AdvFS filesystem to provide
  notification that the specified AdvFS domain is full.
  No more space is available for writing.
```

The examples in this section show how to display EVM events that are contained in a single log file. You can display events that are stored in the various system log files, or monitor them as they occur by using the `evmget` and `evmwatch` commands, which are introduced in Section 13.3.3 and Section 13.3.6. Most systems produce a large number of events, many of which report normal operation. Use event filters to limit the display to a set of events that you consider interesting. Section 13.3.2 introduces the EVM filtering facilities.

Regardless of where the events come from, you use the `evmshow` command to format them for display. See `evmshow(1)` for more details of the `show-template`.

## 13.3.2 Introducing Event Filters

This section introduces event filters and relates them to the `evmshow` command examples from the previous section. Filtering is used more extensively in later sections, which describe event retrieval and monitoring

techniques. The full filter syntax is defined in the `EvmFilter(5)` reference page.

An EVM event filter is a text string that tells EVM which events you want to retrieve. For example, the filter string `[priority >= 600]` selects events that have a priority of 600 or higher. A filter can be very simple, but the filter language is powerful, and with some practice you can easily build and store a filter expression that defines precisely the set of events that you want to monitor. Filters are used by several of the EVM command-line utilities, by the EVM logger, and by system daemons and client applications.

The `evmshow`, `evmget` and `evmwatch` commands support the `-f` option which you use to specify a filter string. You can select the events to be displayed from the `my_events` file, as shown in the following example:

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority] @@"
# cat my_events | evmshow -f "[priority >= 600]" | more
```

(The preceding example was introduced in Section 13.3.1.) In this example, the `-f` option specifies the filter, and selects events that have a priority of 600 or higher. The command reads all events from the file, but returns only those events that match the filter string.

If you know the names of the events you want to retrieve, you can specify them in a filter, as shown in the following example:

```
# cat my_events | evmshow -f "[name sys.unix.fs.advfs.fdmn.full]" | more
```

You can use wildcard characters in place of name components as follows:

- An asterisk (\*) character matches zero or more complete components
- A question mark (?) matches exactly one complete component

For example, use the following command to shorten the preceding example command:

```
# cat my_events | evmshow -f '[name *.advfs.fdmn.full]' | more
```

The wildcard asterisk matches the components `sys.unix.fs`. To avoid any possibility that the shell will expand the wildcard character with filenames, enclose the filter string in single quotes instead of the double quotes. This is always a wise precaution when special characters are used in shell commands.

When you filter by name, EVM assumes that there is a wildcard `.*` at the end of the name string, even if it is not included in the command. Therefore, you might receive events with more name components than you specify. The following two commands are equivalent to each other, but the final wildcard (`.*`) in the first command is unnecessary:

```
# cat my_events | evmshow -f '[name *.advfs.]'
# cat my_events | evmshow -f '[name *.advfs]'
```

You can find the names of events by specifying @name as one of the items in your show-template when you run the `evmshow` command.

Use the filter syntax to combine multiple conditions into a single filter with the AND, OR and NOT keywords, and you can use parentheses to group conditions. The following example command selects all events whose names include the component `advfs`, and that have a priority of 600 or higher:

```
# cat my_events | evmshow -f '[name *.advfs] and [priority >= 600]'
```

The following command also selects events with the name component `binlog`, regardless of their priority. Notice that in this example the keyword `priority` is abbreviated to `pri`, and `name` is abbreviated to `na`. Most filter keywords can be abbreviated as described in the `EvmFilter(5)` reference page.

```
# cat my_events | evmshow -f '((na *.advfs] and [pri >= 600]) or [na *.binlog]'
```

The examples in this section illustrate the most commonly used filter keywords. When you are familiar with applying filters to the `evmshow` command and the EVM commands described in the following sections, you can use the more advanced filter features to create and save useful filters, and to increase your ability to select the events that are most interesting. Advanced filter techniques are described in Section 13.3.12, and the full syntax is given in the `EvmFilter(5)` reference page.

### 13.3.3 Retrieving Stored Events Using `evmget`

System log files store events in many different formats and with different levels of detail making it difficult to produce an ordered view of all events by using traditional system utilities. You can use the `evmget` command to produce an ordered view by retrieving events from each of the various log files, converting them to EVM events if they are not already in that form, and returning a single stream of EVM events. Using the `evmshow` command, you can then turn the EVM event stream into a display format.

The following command pipeline uses the `evmget` command to retrieve all system events, and passes them to the `evmshow` command for display:

```
# evmget | evmshow -t "@timestamp [priority] @@" | more
```

The `evmget` command makes a service connection to the EVM daemon, which starts a new copy of the `get-server` program, `/usr/sbin/evm_getsrv`. The `get-server` program reads the channel configuration file, and runs the `get` function, usually a shell script, for each channel configured in the channel configuration file, `/etc/evmchannel.conf`. This configuration file is described in Section 13.2.1.2.

The `get` function does the following:

- Reads the channel's log file

- Converts the events into EVM format
- Feeds events back to the `evmget` command which writes them to its `stdout` stream

After all the channel `get` functions run and all the events are returned, the `get-server` daemon and the `evmget` command both terminate.

---

### Note

---

Even though events might be stored in log files as lines of text, or in a special binary format, the `evmget` command returns all events in the form of binary EVM events, which can be passed to `evmshow` for display. If you send the output of `evmget` directly to your terminal, the command displays an error message because the binary output cannot be displayed properly and could affect the settings of your terminal. If you pipe the output into another command, such as the `more` command, the `evmget` command is unable to detect the error, and random characters are displayed.

---

Like the `evmshow` command, the `evmget` command supports a filter option to allow you to limit the events it returns. For example, the following command displays only high-priority events:

```
# evmget -f '[pri >= 600]' | evmshow | more
```

It is more efficient to specify a filter with the `evmget` command than with the `evmshow` command. This is because the `evmget` command passes its filter string to the event channel's `get` function, which only returns events that match the filter. Fewer events are passed back through the `get-server` daemon to the `evmget` command, and the commands operate faster because they transfer and process fewer events.

If you want to save retrieved events for later analysis, or to copy them to another system, you can redirect the output of the `evmget` command into a file. For example:

```
# evmget -f '[pri >= 600]' > my_events
```

Saving the binary output of the `evmget` command provides greater flexibility than saving the text output of the `evmshow` command. At a later time you can sort and filter the binary file and pass it to the `evmshow` command to view it in any format you like.

When you experiment with the `evmget` command you will notice that the events appear in batches, probably with all of the binary error logger events appearing first. Within each batch, the events are likely to be ordered chronologically. This is because the `binlog` event channel is specified first in the default channel configuration file, so its `get` function runs first. Each

get function feeds its events back to the `evmget` command in turn, and the `evmget` command outputs them in the order in which it receives them. Because you will usually want to see events in some order (often, but not always, chronological order) you will need to pipe the events through the `evmsort` command, which is described in Section 13.3.4. Section 13.3.5 introduces using the `evmget` command with the `-A` option, which makes it possible to retrieve, sort, and display events without building a pipeline.

Depending on the size and type of your system and the number of events being logged, event retrieval might take a noticeably long time. This is because each retrieval operation requires every channel's `get` function to read through its log files, convert its events to EVM events, and then apply the filter string (if any) to determine whether the event is passed back to the `evmget` command. The larger the log files, the longer this process takes. Careful management of log files will help to speed up the process. If you know that you want to display events that belong to a particular event channel, you can shorten the process by using the `evmget -C` command to display only the specified channel. For example:

```
# evmget -f '[pri >= 600]' -C binlog | evmshow | more
```

In this example, the `get` function runs only on the `binlog` channel, so the command completes its task quickly. A filter string is specified to return events that have a priority greater than 600. You can determine what channels are configured by using the `evminfo -lc` command, or by examining the channel configuration file. See `evminfo(1)` for details.

### 13.3.4 Sorting Events Using `evmsort`

The `evmsort` command takes a stream of EVM events as input, sorts them into the requested order, and writes them to its `stdout` stream. The command is most useful in sorting the output from the `evmget` command, but it can be used to sort EVM events from any source. Full details of the command are given in the `evmsort(1)` reference page.

Section 13.3.3 explained that the events retrieved by the `evmget` command are output in batches, corresponding to the event channel configuration. You can use the `evmsort` command to sort the events into a preferred order, before passing them to the `evmshow` command for display. The following example shows a typical command sequence:

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority] @@"
# evmget -f '[pri >= 600]' | evmsort | evmshow | more
```

By default, the `evmsort` command sorts events into chronological order, so the previous command is suitable for most cases. You can use the `-s` option to declare a sort specification if you want the events sorted differently. A sort specification is a text string that defines one or more sort keys, which are the

data items on which you want to sort the events. The specification is a list of data item names, separated by colons (:). For example:

```
priority:timestamp
```

The preceding specification sorts events by timestamp within priority, so the first group of events that are returned are those with the lowest priority, sorted in their order of occurrence. You might use this specification as follows:

```
# evmget -f '[pri >= 600]' | evmsort -s "priority:timestamp" | evmshow | more
```

The default sort order is ascending, but you can change it to descending for an individual item specifier by appending a minus sign (-). You can explicitly request ascending order by specifying a plus sign (+). For example, the following command displays the highest priority events first (descending order), but within each priority range the events are sorted oldest first (ascending order):

```
# evmget -f '[pri >= 600]' | evmsort -s "priority-:timestamp+" | evmshow | more
```

For consistency with the show-template syntax, the evmsort command allows you to precede each item specifier with an at (@) character, as described in Section 13.3.1. There is no requirement to do this, and it does not affect the operation.

When you establish your sorting preferences, you can create a new default sort sequence by setting the environment variable EVM\_SORT\_SPEC. The following Korn shell (ksh) commands are equivalent to the previous example:

```
# export EVM_SORT_SPEC="priority-:timestamp+"
# evmget -f '[pri >= 600]' | evmsort | evmshow | more
```

You can override the value of the EVM\_SORT\_SPEC variable at any time by supplying a different sort specification with the -s option.

### 13.3.5 Using the -A Option to Simplify the Command String

The EVM commands are designed to be building blocks, with each command doing one specific operation. This gives you great flexibility in developing shell scripts to manipulate event information. When you enter commands from the command line you might prefer to simplify the command.

The most common command sequence for event retrieval is the evmget command, piped into the evmsort command, piped into the evmshow command. You can then pipe the text output into the more command to page the display. Consider the following example:

```
# evmget -f '[pri >= 600]' | evmsort -s "priority-:timestamp+"
| evmshow | more
```

You can simplify the preceding command by using the `evmget -A` command option, which automatically pipes the command output to other EVM commands. For example, you can use the `-A` option to simplify the previous command example as follows:

```
# evmget -A -f '[pri >= 600]' -s "priority-:timestamp+" | more
```

When the `evmget -A` command starts, it automatically runs the `evmsort -A` command, and pipes its output into that command. When the `evmsort` command starts, the `-A` option causes it to start the `evmshow` command, piping events into it for display. You can supply a sort specification with the `-s` option and a show-template with the `-t` option. These options are passed along to the `evmsort` command and `evmget` commands respectively.

The `evmwatch` command supports the `-A` described in Section 13.3.6.

### 13.3.6 Monitoring Events Using `evmwatch`

You can use the `evmwatch` command to monitor event activity through a terminal window. This command is an EVM subscribing client. It makes a connection to the EVM daemon, sends it a subscription request, and waits to receive events. As events arrive, the `evmwatch` command writes them to the standard out stream (`stdout`) as binary EVM events.

Because the output of the `evmwatch` command is a stream of binary events, you cannot display them directly on your terminal. You must use the `evmshow` command to format the events. The following example monitors all events, and displays them on your terminal as they occur:

```
evmwatch | evmshow -t "@timestamp [@priority] @@"
```

Depending on your system type, and the level of event activity, this command might run for a while before any events are displayed. The command continues to run until you terminate it to regain control of your terminal. In most cases you can do this by pressing `Ctrl/c`.

When a system is operating correctly, many of the events posted are low-priority informational events. You might want to filter these events out, particularly if your system has a high level of event activity. You can do this by supplying a filter to the `evmwatch` command:

```
# evmwatch -f "[priority >= 400]" | evmshow -t  
"@timestamp [@priority] @@"
```

This example watches for events with a priority of error or higher. You can change the filter string to exclude any set of events that occur regularly and are not interesting. Alternatively, you might need to watch for a particular set of events.



Unlike the examples in the earlier sections, the preceding examples do not show the output of the `evmshow` command being piped into the `more` command for paging. This is because the `evmwatch` command is a realtime monitor. The `evmwatch` command outputs events as they occur, rather than displaying them from a file. After it displays the first screen of data, a paging command might wait for operator input before reading more data from its input pipe which, over time, could lead to congestion in the pipeline. Because the EVM daemon cannot wait for its client (the `evmwatch` command) to clear its backlog, this results in the `evmwatch` command missing events. Instead of piping events to a paging command, you should display the output from the `evmwatch` command directly on a terminal window and use the scrollbar to review the event list.

Avoid piping the output of the `evmwatch` command into the `evmsort` command because the `evmsort` command cannot sort events until it reads to the end of its input. As a monitoring program, the `evmwatch` command usually waits for input until it is explicitly killed. As a result, if you pipe the output of the `evmwatch` command directly into the `evmsort` command, you will see no output from the `evmsort` command.

Section 13.3.5 introduced the use of the `-A` option, which simplifies the command string by running the `evmsort` command and the `evmshow` command automatically. The `evmwatch` command also supports the `-A` option and automatically runs the `evmshow` command when you use it. You can specify a show-template as an option to the `evmwatch` command as follows:

```
# evmwatch -A -f "[priority >= 400]" -t "@timestamp  
[@priority]  @@"
```

As with the `evmget` command, you can capture a set of interesting events in a file, so that you can review them later. It is generally more useful to store events in binary form than in text form, so you should send the output of the `evmwatch` command directly to a file, rather than piping it into the `evmshow` command first, as follows:

```
# evmwatch -f "[priority >= 400]" > my_events
```

The `evmwatch` command supports additional options that are useful for monitoring events from within a shell script. See `evmwatch(1)` for more information.

### 13.3.7 Posting Quick Message Events Using `evmpost`

Although most events are likely to be posted by system and application software, there might be times when you want to post an event from the command line or from a shell script. For example, you might want to post a message event in the system log to note that a task is complete, or that you

noticed something interesting. Making an entry in the system log makes it easy to establish when other events occurred relative to your entry.

You can post an event by using the `evmpost` command. The simplest form of this command is the quick message form, which you can specify by using the `-a` (administrator) or `-u` (user) option. To post a message, you supply the message on the command line as a quoted string:

```
# evmpost -a "Fire drill started - evacuating computer room"
```

Administrative quick messages are posted with the name `sys.unix.evm.msg.admin`, so you can search for them with a name filter:

```
# evmget -f '[name *.msg.admin]' | \
evmshow -t 'timestamp [@priority] @@'
27-Jun-2000 15:40:49 [200] EVM admin msg: Fire drill
started - evacuating computer room
```

By default, the message is posted as a notice event, with a priority of 200. You can change the priority with the `-p` option. For example, setting the priority to 400 categorizes the message as an error event:

```
# evmpost -p 400 -a \
"Users reporting possible network problems"
```

By default, only the root user or members of the `adm` group can post events with the `-a` option, although you can make it available to other privileged users by editing the authorization file, `/etc/evm.auth`, as described in Section 13.2.2.2. Any user can specify the `-u` option to post messages in the same way. If necessary you can restrict this privilege to trusted users by editing the authorization file.

### 13.3.8 Listing Registered Events

You register events by adding template file entries as described in Section 13.2.4, and running the `evmreload` command with the `-d` option to make them known to the EVM daemon, or restarting the system.

You can use the `evmwatch -i` command to retrieve a list of registered events. Pipe the output from the `evmwatch -i` command to the `evmshow` command to display the event templates in any desired format. For example:

```
# evmwatch -i | evmshow -t "@name [@priority] @format" -x
```

Templates are returned as binary EVM events which you can either redirect into a file or pipe to the `evmshow` command for display. In the preceding example, the `show-template` (`-t` option) displays the name of the event, the priority, and the message format. The `-x` option causes each summary line to be followed by an explanation of the event.

Because you are displaying templates (not real system events) you specify a command sequence that requests only the event's message format, not an

expanded message. In the output, the summary lines display the messages with names of variables rather than their values. For example you might see the following summary line and explanatory text:

```
sys.unix.fs.advfs.fdmn.bal.error [400] AdvFS: Balance error on AdvFS domain $domain
This event is posted by the balance(8) command to indicate that an
error has occurred while balancing the domain.

Action: Please refer to the balance(8) reference page for further
information.
```

In this example, the `$domain` variable is replaced by the domain name when you use the `evmget` command to retrieve a posted instance of the event.

If you do not want to see all registered events, you can use a filter to limit the output of the `evmwatch` command to the events in which you are interested:

```
# evmwatch -i -f '[name *.evm]' | evmshow -t "@name
[@priority] @format" -x
```

### 13.3.9 Posting Events from a Shell Script

Use the `evmpost` command to post a newly registered event, by passing event information to the command in source (text) format. A full description of the event syntax is provided in the `evmpost(1)` reference page. Source-level posting is most useful in a shell script that performs a routine operation, where the event might indicate success or failure of the operation. This section describes a procedure to create and post a new event that informs you when a backup is finished. The basic steps are:

1. Create a template file and verify its syntax.
2. Install the template file and make it known to the EVM daemon.
3. Update the authorization file to allow the events to be posted.
4. Write shell script commands to post the event.

Event design guidelines are given in the *Programmer's Guide*. You should be familiar with the concepts described in that book before you begin designing a new event. In this example, the backup script posts one of two events, `local.admin.backup.ok` with a priority of 200 (notice) and `local.admin.backup.failed`, with a priority of 400 (error). The failure event includes a variable item named `result_code`, to hold the exit code returned by the backup program. The variable is an 8-bit unsigned integer, and in the template it has a dummy value of zero. This dummy value is replaced with an actual value when the event is posted. The template file syntax is described in the `evmtemplate(4)` reference page.

The following procedure describes how to create and post a new event:

1. Use a text editor, such as `vi`, to create the following text file:

```

# This file contains EVM event templates for local
# backup notification events.
event {
    name local.admin.backup.ok
    format "BACKUP: Backup completed OK"
    priority 200
}

event {
    name local.admin.backup.failed
    format "BACKUP: Backup failed - code $result_code"
    var {name result_code type UINT8 value 0}
    priority 400
}

```

2. Save the file in the `/var/evm/adm/templates/local` directory with the name `backup.evt`. (Create the `/local` directory if it does not exist.)

You can install new template files in any directory under `/var/evm/adm/templates`, but name subdirectories and template files according to the names of your events for ease of identification. Keeping a small number of closely-related event templates in a single template file simplifies maintenance.

3. Verify the template syntax. The syntax of a template file is identical to the syntax used to post an event, so you can use the `evmpost -r` command to check the syntax. The `-r` option instructs the `evmpost` command not to post the event, but to validate the syntax, convert the input into binary EVM events, and then write the EVM events to its standard output (`stdout`) stream. Use the `evmpost -M` command option to prevent the merging of template items into the event, or to add any environmental items such as a timestamp or host name.

As with any stream of binary EVM events, you can then use the `evmshow` command to verify the output of the `evmpost` command. To do this, enter the following command:

```
# cat /var/evm/adm/templates/local/backup.evt \
| evmpost -r -M | evmshow -t "@priority @@"
```

If you created the file correctly, the following output is displayed:

```
200 BACKUP: Backup completed OK
400 BACKUP: Backup failed - code 0
```

4. Check that the file is owned by `root` or `bin`, and its permissions are set to `0400`, `0600`, `0440` or `0640`. Correct the permissions by using the `chown` command and the `chmod` command if necessary.
5. Run the following command to instruct the EVM daemon to reload its configuration:

```
# evmreload -d
```

If the command displays an error message, correct the problem and reenter the command. The most likely problem is that the ownership or permissions of the file are incorrect.

6. Verify template registration by using the `evmwatch -i` command option, which retrieves templates from the EVM daemon's database. Because the `evmwatch` command outputs the templates in the form of binary EVM events, you can use the `evmshow` command to display them. You need to show only the names of the events to be sure that they are registered correctly, as shown in the following example:

```
# evmwatch -i -f "[name local.admin.backup]" \  
| evmshow -t "@name"  
local.admin.backup.ok  
local.admin.backup.failed
```

7. Update the authorization file, `/etc/evm.auth`, to allow the events to be posted. Add the following lines to ensure that only the root user can post the events, but any user can see the events:

```
# Local backup events:  
event_rights {  
    class      local.admin.backup  
    post       root  
    access     +  
}
```

Only the first three components of the name are specified. These components are common to the two new events, and when either of the events is posted its name will match this entry,

8. Run the `evmreload -d` command option, so that the daemon recognizes the new authorizations.
9. Verify that the events were logged correctly by using the following commands:

```
# echo 'event {name local.admin.backup.ok}' | evmpost  
# echo 'event {name local.admin.backup.failed}' | evmpost  
# evmget -f '[name local.admin.backup]' \  
| evmshow -t '@timestamp [@priority] @@'  
  
28-Jun-2000 15:21:39 [200] BACKUP: Backup completed OK  
28-Jun-2000 15:21:40 [400] BACKUP: Backup failed - code 0
```

In the preceding example, the `evmpost` command reads the source input from its standard input (`stdin`) stream, converts it to an EVM event, and posts it. The output from the final command shows the posted events. It includes the priorities specified in the template file because the EVM daemon merges the template information into each event as it is posted. Notice that the value of the code in the second event is zero, because that is the dummy value supplied in the template

file, and that value was not overridden in the posted event. In the backup script the value is set to something other than zero.

10. Add the posting commands to your backup script, as shown in the following example:

```
#!/bin/sh
# This shell script runs the backup operation
# and posts an EVM event to indicate success
#or failure.

do_backups # Performs the backup operation
if [ $? -eq 0 ]
then
  echo 'event {name local.admin.backup.ok}' | evmpost
else
  RES=$?
  evmpost << END
  event {
    name local.admin.backup.failed
    var { name result_code type UINT8 value $RES }
  }
END
fi
```

In the preceding example, the input to the `evmpost` command for the success event is simple, so it is supplied on the same line by using the `echo` command. For the failure event, the value of the `result_code` variable must also be supplied. To supply this value, the shell's `<<` syntax provides a more structured multi-line form of input. Both forms of input supply source code input to the `evmget` command through its standard input (`stdin`) stream.

See `evmpost(1)` for more information about posting events from the command line, or from within a shell script.

### 13.3.10 Understanding the EVM Mark Event

When you review or monitor event activity you will observe the following event that occurs every 15 minutes:

```
26-Jun-2000 08:57:45 [200] EVM: Mark event
```

The `evmlog` event channel posts this event to ensure that there is periodic event activity. If your system has a problem and you need to determine when it was last operational, you can look for mark commands in the system log by using the following command:

```
# evmget -f "[name *.evm.mark]" | evmshow -t "@timestamp @last_timestamp @@"
26-Jun-2000 00:57:35 26-Jun-2000 04:42:40 [16 times] EVM: Mark event
26-Jun-2000 04:57:41 - EVM: Mark event
```

```
26-Jun-2000 05:12:41 - EVM: Mark event
26-Jun-2000 05:27:41 - EVM: Mark event
26-Jun-2000 05:42:41 26-Jun-2000 09:12:45 [15 times] EVM: Mark event
```

If the default logger configuration file is in use, you will usually see three individual mark events, followed by a single event preceded by [n times], where n is a number up to 16. This is the result of the logger's suppression facility, which minimizes wasted space by combining multiple events over a period of up to four hours. The normal timestamp value shows the first occurrence of a combined event, and the `last_timestamp` data item shows the time of the last occurrence. The example includes the `last_timestamp` data item in the `show-template` which displays the last mark event, posted at 09:12:45. This mark event tells you that the system was operational at that time.

To disable mark event posting, edit the channel configuration file to make either of the following changes:

- Comment out the `evmlog` channel's `fn_monitor` entry to disable it completely
- Change the `mon_period` value for the channel to change the frequency with which the event is posted

Refer to Section 13.2.1.2 and `evmchannel(4)` for details of the channel configuration file. Refer to Section 13.2.1.3 and `evmlogger.conf(4)` for more information about event suppression.

### 13.3.11 Viewing Events Using the SysMan Event Viewer

The SysMan graphical event viewer provides a simple and convenient interface to the system event logs. Because the event viewer is an integral part of the SysMan system management suite you can use it in a variety of graphical domains, including an X Windows display or a character cell terminal, as a PC application, or from a Web browser. You can also launch the viewer from the SysMan Station. Refer to Chapter 1 for information about using SysMan.

To launch the event viewer from the command line, enter the `sysman` command, then open the Monitoring and Tuning menu branch. Select the View Events option to start the event viewer. To launch the event viewer directly from CDE, open the tool drawer on the CDE front panel and select in turn System\_Admin, DailyAdmin, and Event Viewer.

When you run the event viewer for the first time a warning message might indicate that events are filtered to show only high priority events. If your system is operating normally it is likely that no events are displayed in the event summary window. To choose the events you want to see, select the Filter... button at the bottom of the window, and change the filter criteria in

the Filter window. If you want to see all stored events, make sure that all of the check boxes at the left side of the window are in the unchecked state, and select the OK button. If your system produces a high level of event activity you can reduce the number of events shown, and the time taken to display them, by checking the Priority box and adjusting the priority range. Setting the range to 400-700 displays all events with a priority of `error` and higher. Setting the low end of the range to 300 includes warning events in the display.

You can check any of the buttons at the left of the Filter window to include additional criteria in the display filter. Each time you make a change you must select the Apply button to apply the change to the event list, or select OK to apply the change and return to the main viewer window.

The Filter dialog window offers an intuitive and convenient way for you to build an event filter string without having to type it. If you are familiar with the filter syntax and you want to make better use of its power, you can enter a filter string through the Advanced Filter dialog box, which you access by selecting the Options... button at the bottom of the main event window. You can also save a filter string and reuse it later. For more information about the filter syntax refer to the `EvmFilter(5)` reference page.

One of the most important features of the viewer is the ease with which you can display a detailed view of any event. Simply select the event in the summary window and select the Details... button to see all the information available, including explanation text and, in the case of a `binlog` event, the translation from `DECEvent` or `Compaq Analyze`. From the Event Details window you can browse through the event list without returning to the main window.

You can change the viewer display, including the source of events, by selecting the Customize... and Options... buttons. To change the order in which events are displayed, select the Sort... button. Select the Help... button from any window for detailed information about the viewer and its facilities.

---

#### Note

---

The event viewer does not monitor event activity in real time. To display an updated view of the event list, select the Refresh button from the main window.

---

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.



## 13.3.12 Advanced Selection and Filtering Techniques

Section 13.3.2 introduced event filters, and provided examples of filtering events by name and priority. This section describes some additional filtering techniques that you can use to further improve event selection, so that you receive only the events in which you are interested.

- Section 13.3.12.1 describes how to filter events according to their time of posting.
- Section 13.3.12.2 describes how to filter by using the `event-id` identifier.
- Section 13.3.12.3 describes how to filter by using reserved component names.
- Section 13.3.12.4 describes how to use filter files.

### 13.3.12.1 Filtering By Time

You can filter for events according to the time at which they were posted by using the `timestamp`, `before`, `since`, and `age` keywords. You might find that the `age` keyword is the easiest of these keywords to use, and the most useful for everyday operation.

When you use the `timestamp` keyword you must supply a string that defines a time range in the following way:

```
year:month-of-year:day-of-month:day-of-week:hours:minutes:seconds
```

You can use an asterisk (\*) as a wildcard character for any of the components, so to select events that occurred on July 6, 2000 you might use the following commands:

```
# export EVM_SHOW_TEMPLATE="@timestamp [@priority] @@"  
# evmget -A -f '[timestamp 2000:7:6:*:*:*]' | more
```

The asterisks (\*) in the final four components indicate that you are interested in all events that occurred on that day, no matter what time they occurred. You can also specify one or more ranges in any position, as shown in the following command:

```
# evmget -A -f '[timestamp 2000:*:*:1-3,5:*:*]' | more
```

The fourth component specifies the day of the week. Searching for events with posting times in the range 1-3 or 5 yields all events that were posted on a Monday, Tuesday, Wednesday or Friday in the year 2000.

The `before` and `since` keywords use similar specifier strings, but you cannot use wildcard characters and there is no day of the week indicator. For example, the following command finds events that were posted after 3:00p.m. on July 6, 2000:

```
# evmget -A -f '[since 2000:7:6:15:0:0]' | more
```

The `age` keyword provides a more convenient and intuitive way to select events according to their timestamps. As a system administrator you might be most interested in recent events that indicate a system problem. You can combine the event filter's `priority` and `age` keywords to find such events. For example, the following command sequence shows all events with a priority of error (400) or higher, that occurred either yesterday or today (the age of the event is less than 2 days):

```
# evmget -A -f '[pri >= 400] and [age < 2d]' | more
```

In the preceding example, `2d` specifies events that are less than 2 days old. You can specify an age in seconds (s), minutes (m), hours (h), days (d), or weeks (w). See `EvmFilter(5)` for information about how each specifier is used in calculating an event's age.

You can use a more complex filter to return events that occurred within a more specific period. The following example finds error events that occurred more than 3 days ago, but less than 6 days:

```
# evmget -A -f '[pri >= 400] and ([age < 6d] and [age > 3d])' | more
```

See `EvmFilter(5)` for detailed information on selecting events according to their timestamps, and the full filter syntax.

### 13.3.12.2 Using the Event-Id to Select Events for Detailed Display

Using the `evmshow -d` command option to display events can result in a large amount of output and you might want to limit the number of displayed events. Events that are posted through EVM contain a sequential identifier known as the `event-id`. You can use the `event-id` to select a specific event or a range of events for detailed display.

The `event-id` is not guaranteed to be unique within any particular set of events because the daemon's counter is set to zero each time it is restarted. To ensure that an event is unique, you must also use the timestamp when selecting events as shown in the following example:

```
# evmget -A -f '[age < 1d]' -t "@timestamp @event_id @@" | more
15-Apr-1999 14:19:06 0  EVM daemon: Configuration completed
15-Apr-1999 14:19:06 1  EVM daemon: Initialization completed
15-Apr-1999 14:19:06 2  EVM logger: Logger started
15-Apr-1999 14:19:06 3  EVM: Mark event - initial
15-Apr-1999 14:19:06 5  EVM logger: Started eventlog /var/evm/evmlog/evmlog.19990415
[1]                               [2]
.
.
.
```

- [1] The age filter keyword selects all events that have occurred today, as indicated by the timestamp in the first column of data.

- 2 The `@event_id` specifier in the show template instructs the `evmshow` command to display the `event-id` for each retrieved event, which is shown in the second column of data.

When the `event-ids` are displayed, you can select the interesting events. For example, use the following command to display details of the initial mark event, which has an `event-id` of 3 in the preceding example output:

```
# evmget -f '[age < 1d] and [event_id = 3]' | evmshow -d | more
```

You can select a range of events by using a more complex filter as shown in the following example:

```
# evmget -f '[age < 1d] and [event_id >= 1] and [event_id <= 3]' \
| evmshow -d | more
```

Choose the time range carefully to select the right set of events. If you recently rebooted your system, specify a filter of `[age < 2h]` to select events occurring within the preceding 2 hours.

The most convenient way to select events for detailed display is to use the event viewer described in Section 13.3.11.

### 13.3.12.3 Searching for Reserved Component Names

Some event names include reserved component names as name extensions. These components begin with an underscore character (`_`), and are usually followed by a component that identifies the item for which the event is being posted. For example, the names of many hardware-related events include the component `_hwid`, followed by the numeric hardware identifier of the item. You can search for all such events by using the following command:

```
# evmget -A -f '[name *._hwid]' | more
```

If you know the hardware identifier of a specific device, you can narrow the search for events related to that device by using a command similar to the following:

```
# evmget -A -f '[name *._hwid.4]' | more
```

### 13.3.12.4 Using Filter Files

You can save a useful filter in a file and recall it by using EVM's indirect filter facility. Filter files have names with the suffix `.evf`, and can contain any number of named filters. For example, the following filter file entry selects all binlog events that refer to SCSI devices:

```
filter {
    name "scsi"
    value "[name @SYS_VP@.binlog.hw.scsi]"
    title "Binlog SCSI events"
}
```

In this example, the @SYS\_VP@ is a standard EVM macro that is replaced by `sys.unix` when the filter is used.

To use indirect filtering, specify the at sign (@), followed by the name of the file containing the filter instead of a filter string, as shown in the following example:

```
# evmget -A -f @binlog
```

You do not need to include the `.evf` suffix when you specify a filter file name in such commands.

The previous example uses the first filter in the file, but you can choose a different filter by specifying its name as follows:

```
# evmget -A -f @binlog:scsi
```

You can include as many filters as you like in a single file, or you can keep each filter in its own file. The preceding example specifies the `binlog` filter, which is included in EVM. Other filters are provided in the `/usr/share/evm/filters` directory. Use these files as examples for establishing your own filter library.

The `evmshow -F` command option provides an easy way for you to see the contents of a stored filter. The `-F` option causes the `evmshow` command to display the filter string and then exit without reading any events. In the following example, the `evmshow` command displays the contents of the filter named `scsi`, stored in the `binlog.evf` file:

```
# evmshow -f @binlog:scsi -F  
( [name sys.unix.binlog.hw.scsi] )
```

See `evmfilterfile(4)` for complete information about the syntax of filter files, and where to locate your files.

---

**Note**

---

Do not edit the filter files provided in the `/usr/share/evm/filters` directory. Your changes might be overwritten without warning by a future installation update.

---

### 13.3.13 Logging and Forwarding Events

The response to an event is any action determined by your site-specific needs and conditions. This response can range from activating alarms or paging responsible personnel, to making a log entry or ignoring an expected occurrence of a regular activity.

You can configure the event processing sequence to perform a series of dependent tasks, by using an event output by one task as the trigger to

activate the next process. EVM provides an interface to the response activity through the logging facility. The available options are event storage and event forwarding.

The EVM logger, `evmllogger`, started automatically by the EVM daemon, is responsible for the following:

- Displaying selected events on the system console or other device  
If a terminal device is indicated as the `logfile` in the configuration file, all events meeting the filter specifications of an `eventlog` statement are formatted for display on the terminal. (See Section 13.2.1.3 for a discussion of the configuration file.)
- Storing selected events in one or more log files
- Forwarding selected events to interested parties in some other form

The logger is an ordinary EVM client that is controlled through a configuration file. The default is the `/etc/evmllogger.conf` file, described in Section 13.2.1.3. See `evmllogger.conf(4)` for additional information on this file and `evmllogger(8)` for additional information on the command.

### 13.3.13.1 Logging Events

All events meeting the specifications of an `eventlog` statement in the configuration file are written to the event log. 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 `evmllogger.conf(4)` for the explanation of this criterion.

### 13.3.13.2 Using Forwarding to Handle Events Automatically

If you want to automate the handling of selected events, you can configure the 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 `evmllogger.conf(4)` for more information.

All events meeting the filter specifications of a `forward` statement in the configuration file are written to the standard input (`stdin`) of the command

specified in the statement. The command is the name of a shell script, a single UNIX command, a series of UNIX commands (pipeline), or any other executable statement. The following operations are typically specified as a forwarding action:

- Specifying the `mail` command or `mailx` command, or another command-line mail processor, to send a mail message to a responsible person or paging service
- Invoking additional software that causes emergency shutdown procedures to commence
- Invoking a dependent process that is waiting for the event to occur

When configuring the logger to forward an event, note the following:

- The event selected for forwarding is piped into the configured forwarding command. If your commands need to deal with text information, the `evmshow` command must be the first command in the pipeline so that the event is converted to text form.
- The logger executes the forwarding command asynchronously, meaning that it starts the command and then continues with its normal operation without waiting for the command to finish. The following behaviors are normal:
  - If multiple forwarders are specified in the logger's configuration file, and the same event is to be handled by more than one forwarder, the logger starts each forwarding command without waiting for the others to finish, so the commands may execute simultaneously.
  - If the logger receives another event to be processed by a forwarding command, and the command is still processing the previous event, the logger queues the new event. When the command finishes, the logger restarts it, passing it the new event. By default, the logger will queue up to 100 events for each forwarding command. You can increase this limit by specifying a `MAXQUEUE` keyword in the forwarder's configuration.

See `evmlogger.conf(4)` for more information.

- Event text 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 must exclude mail events.

Use the logger's secondary configuration file facility for adding forwarders or other configuration items as described in Section 13.2.1.4 .

## 13.4 Troubleshooting EVM

If you suspect that EVM is not operating correctly, the first step is to check the message files in the `/var/evm/adm/logfiles` directory. 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 to take in trying to resolve such problems:

Kernel events are not being posted.

Check the EVM daemon log file for errors by using the following command:

```
# more /var/evm/adm/logfiles/evmdaemon.log
```

Check for the presence of the kernel interface pseudodevice by using the following command:

```
# ls -l /dev/kevm
```

If this pseudodevice is not present, create it by using the following command:

```
# dsfmgr -vF
```

A subscribing application fails to receive expected events.

Verify that the poster is authorized to post these events by checking the authorization file by using the following command:

```
# more /etc/evm.auth
```

Verify that the event is registered by using the following command:

```
# evmwatch -i -f '[name event_name]' | \  
  evmshow -t "@name"
```

If the events are still not shown, run `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, by using the following command:

```
# more /etc/evm.auth
```

Verify that the expected events are actually being posted by using the following command:

```
# evmwatch | evmshow -t "@name @"
```

Run the program that posts the event, and check that the preceding `evmwatch` command displays them correctly.

A posting program is unable to post events.

Verify that the EVM daemon is running by using the following command:

```
# ps -aef | grep evmd
```

Verify that the poster is authorized to post these events by checking the authorization file by using the following command:

```
# more /etc/evm.auth
```

Verify that the event is registered by using the following command:

```
# evmwatch -i -f '[name event_name]' | \  
  evmshow -t "@name"
```

If the events are still not shown, run the `evmreload` command and 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.

By default, EVM only retrieves `binlog` events that were posted within the last 8 days. If you want to see older `binlog` events, edit the channel configuration file, `/etc/evmchannel.conf`. In the `binlog` channel group, the default `fn_get` line includes the option `-r 8d`, meaning that events for only the past 8 days are retrieved. You can either remove this option completely to see all `binlog` events or change the 8 to some other value.

Check that the `binlogd` and `syslogd` daemons are running by 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 &#amp; 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 SysMan Station daemon log files (`/var/adm/sysman/sysman_station/logs`).



Use the `ls -L` command when listing file sizes to ensure that you see the file itself and not a symbolic link or a context-dependent symbolic link (CDSL).

See `binlogd(8)` for details of binary log size management, but note that 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 the `cron` utility to perform a regular archiving task. You can reduce the sizes of the `evmlog` files by changing configuration values in the `/etc/evmlogger.conf` file and the `/etc/evmchannel.conf` file.

Expected events are not being logged.

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:

1. Set `remote_connection` to `True` in the daemon configuration file.
2. Run the following command:

```
# evmreload -d
```

Be sure to consider the security implications of enabling a remote connection.

Invalid filter message from remote connections.

This might happen when an attempt to connect to a remote system to retrieve or monitor events results in an invalid filter message, even though the same filter works correctly when used on the local system.

The filter syntax changes with new releases of the operating system, and newer keywords or abbreviations might not be recognized by older versions. Log in to the remote system and review `EvmFilter(5)` to determine whether the syntax used in your filter is supported by that version of the operating system.

Binlog events are not being translated.

Use the following procedures to troubleshoot the absence of a translation utility:

1. Run the following command:

```
# usr/sbin/dia
```

If DECEvent is installed, this command displays the translated contents of the current binary error log file, `/var/adm/binary.errlog`.

2. If the `dia` command is not found, use the following command to test the status of the DECEvent software subset (the distribution kit):

```
# setld -i | grep OSFDIA
```

This command returns the string `OSFDIABASE***` when the DECEvent Base Kit (Translation/Analysis) is installed. If it is not installed, mount the installation media and use the `setld` command to install the subset. 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 the `desta` daemon is not running, the Compaq Analyze utility might still be installed but not running or properly configured. To verify that Compaq Analyze is installed, look for the binaries by using the following command:

```
# ls /usr/opt/compaq/svctools/bin/desta*
```

3. If you do not find the binaries, install Compaq Analyze from the distribution media by using the `setld` command. Contact your sales and Support organization or your local vendor for information on Compaq Analyze.

---

## 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) or to memory. 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 the error core dumps produced by an application, after which the system usually keeps running. After a crash dump, the system is shut down to the console prompt (>>>) and 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 virtual memory swap space provided on disk. Administering the swap space is described in Chapter 3. System event management is described in Chapter 12, which describes the `binlogd` and `syslogd` event management channels.

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. You might 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 or from memory 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 is also known as a continuable dump and is useful for estimating crash dump size during dump configuration planning.
  - `sysconfig(8)` and `sysconfigdb(8)` – Maintains the kernel subsystem configuration and is used to set kernel crash dump attributes that control crash behavior. You can also use the graphical interface `/usr/bin/X11/dxkerneltuner` to modify kernel attributes. See `dxkerneltuner(8)` 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 command 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 the `/var/adm/crash` directory, although you can redirect crash dumps to any file system that you designate and also to a remote host. In common with many other system directories, the `/var/adm/crash` directory is a context-dependent symbolic link (CDSL) which facilitates joining systems into clusters. The CDSL for this directory is `/var/cluster/members/member0/adm/crash`. Within this directory, the following files are created or used:

- `/var/adm/crash/bounds` – A text file specifying the incremental number of the next dump (the `n` in `vmzcore.n`)
- `/var/adm/crash/minfree` – A file that specifies the minimum number of kilobytes to be left after crash dump files are written
- `/var/adm/crash/vmzcore.n` – The crash dump file, named `vmcore.n` if the file is 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`, `/etc/binlog.conf`, and `/etc/evmdaemon.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 `savecore` command moves crash dump information from the swap partitions or memory into a file and copies the kernel that was running at the time of the crash into another file. You can analyze these files to help you determine the cause of a crash. The `savecore` command also logs the crash in system log files.

You can invoke the `savecore` command from the command line. For information about the command syntax, see `savecore(8)`.

### 14.2.1 Setting Dump Kernel Attributes in the Generic Subsystem

You can control the way that a crash dump is taken by setting kernel attributes defined in the `generic` subsystem, as follows:

- `dump_savecnt` – Limits the number of successful crash dumps that are generated for a single crash and reboot sequence or disables dumping. See Section 14.2.2.
- `dump_to_memory` – Specifies whether primary system core dumps are written to memory or to disk. See Section 14.2.2.

- `dump_sp_threshold` – Controls the partitions to which the crash dump is written. The default value causes the primary swap partition to be used exclusively for crash dumps that are small enough to fit the partition. See Section 14.2.4.
- `dump_user_pte_pages` – Specifies whether or not you want to include user page tables in partial crash dumps. This attribute is off by default. See Section 14.3.2.
- `expected_dump_compression` – Specifies the level of compression that you typically expect the system to achieve. The setting is 500 by default, but can be an integer from 0 to 1000. See Section 14.3.4.
- `partial_dump` – Specifies whether a partial crash dump or a full crash dump is preserved. This attribute is on by default. See Section 14.3.3.
- `compressed_dump` – Specifies whether a dump is compressed to save space. This attribute is on by default. Even if set to off, the value of other dump attributes might cause it to be automatically set to on. See Section 14.3.5 and also Section 14.3.6.
- `dump_kernel_text` – Enables or disables the inclusion of kernel text pages in the dump creating a larger dump file. This attribute only applies when partial dumps are enabled. See Section 14.3.3.
- `live_dump_dir_name` – Specifies the full path to the directory where of continuable dumps are written. See Section 14.4.1.
- `live_dump_zero_suppress` – Enables or disables zero compression of continuable dumps that. Dump files take longer to create but occupy less space. See Section 14.4.1.
- If available, dumping to exempt memory is controlled by the following attributes:
  - `dump_exmem_addr` – Identifies the starting address (virtual or physical) for a region of exempt memory used for writing primary dumps.
  - `dump_exmem_size` – Specifies the size (in bytes) of the exempt memory region to which dumps are written.
  - `dump_exmem_include` – Specifies whether or not exempt memory pages are included in the dump.

See Section 14.3.6 for a description of this feature.

The following command displays typical dump attribute settings:

```
# sysconfig -q generic | grep dump
compressed_dump = 1
dump_exmem_addr = 0
dump_exmem_size = 0
dump_exmem_include = 0
```

```
dump_kernel_text = 0
dump_savecnt = 1
dump_sp_threshold = 4096
dump_to_memory = 0
dump_user_pte_pages = 0
expected_dump_compression = 500
live_dump_zero_suppress = 1
live_dump_dir_name = /var/adm/crash
partial_dump = 1
```

See `sys_attrs_generic(5)` for a description of the dump attributes and settings. See `sysconfig(8)` and `sysconfigdb(8)` for information on setting attribute values.

## 14.2.2 Crash Dump File Creation

When the `savecore` command begins running during the reboot process, it determines whether a crash dump occurred and whether the file system contains enough space to save it. (The system saves no crash dump if you shut it down and reboot it; that is, the system saves a crash dump only when it crashes.)

The value of the `dump_savecnt` attribute controls the number of dumps. Possible values are:

- 0 (zero) – Never generate a crash dump.
- 1 – Generate a primary crash dump (the default).
- 2 – Generate a secondary crash dump.

The value of the `dump_to_memory` attribute controls the location of dumps and interacts with the value of the `dump_savecnt` attribute as follows:

- -1 – Writing dumps to memory is disabled. This value also disables writing a secondary dump when the value of the `dump_savecnt` attribute is 2.
- 0 (zero) – Dumps are written to disk except in the event of disk failure, in which case they are written to memory. This is the default behavior.
- 1 – Dumps are written only to memory when sufficient memory is available. A special case is if secondary dumps are enabled (`dump_savecnt=2`). See `sys_attrs_generic(5)` for more information.

Under certain circumstances, dumps in memory might be overwritten. To prevent an overwrite from happening, you can also write dumps to a protected region of memory called exempt memory. See Section 14.3.6 for more information.

If a crash dump exists and the file system contains enough space to save the crash dump files, the `savecore` command moves the crash dump and a copy of the kernel into files in the default crash directory, `/var/adm/crash`. (You can modify the location of the crash directory.)

You can also choose to:

- Write all crash files to a remote host using a network connection as described in Section 14.3.7.
- Write continuable dump files to an alternate directory as described in Section 14.4.1.

The `savecore` command stores the kernel image in the `vmunix.n` file, and by default it stores the (compressed) contents of physical memory in the `vmzcore.n` file.

The `n` variable specifies the number of the crash, which is recorded in the `bounds` file in the crash directory. After the first crash, the `savecore` command creates the `bounds` file and stores the number 1 in it. The command increments that value for each succeeding crash.

The `savecore` command runs early in the reboot process so that little or no system swapping occurs before the command runs. This practice helps ensure that crash dumps are not corrupted by swapping.

### 14.2.3 Crash Dump Logging

After the `savecore` command writes the crash dump files, it performs the following steps to log the crash in system log files:

1. Writes a reboot message to the `/var/adm/syslog/auth.log` file.

If the system crashed 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. See `syslogd(8)` for more information about how system logging is performed.

3. Attempts to save the binary event buffer from the crash dump.

The binary event buffer contains messages that can help you identify the problem that caused the crash, particularly if the crash resulted from a hardware error.

The `savecore` command saves the binary event buffer in the `/usr/adm/crash/binlogdumpfile` file by default. You can change the location to which `savecore` writes the binary event buffer by modifying the `dumpfile` entry in the `/etc/binlog.conf` file. If you remove the `dumpfile` entry from the `/etc/binlog.conf` file, `savecore` does not save the binary event buffer.

Later in the reboot process, the `binlogd` daemon starts up, reads the contents of the `/usr/adm/crash/binlogdumpfile` file, and moves those contents into the `/usr/adm/binary.errlog` file, as specified in the `/etc/binlog.conf` file. The `binlogd` daemon then deletes the `binlogdumpfile` file. See `binlogd(8)` for more information about how binary error logging is performed.

4. The system might crash before all kernel events are handled and posted. In such cases, the `savecore` program recovers such events and stores them for later processing. This recovery happens only if any such events are available and if the `savecore` program is able to successfully extract and save the events. By default, the events are stored in the `/var/adm/crash/evm.buf` file. See `savecore(8)` and `EVM(5)` for more information.

## 14.2.4 Swap Space

When the system creates a crash dump to disk, it writes the dump to the swap partitions. The system uses the swap partitions because the information stored in those partitions has meaning only for a running system. After the system crashes, the information is useless and can be 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` file entry shows three swap partitions available:

```
vm:
  swapdevice=/dev/disk/dsk0b, /dev/disk/dsk3h, /dev/disk/dsk13g
  vm-swap-eager=1
```

The following list describes how the system determines where to write the crash dump:

- If the crash dump fits in the primary swap partition it 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 any of the available swap partitions, the system writes the crash dump spanning the secondary and tertiary swap partitions until those partitions are full. If it requires more space, it then writes the remaining crash dump information starting from the end of the primary swap partition (possibly filling the primary swap partition also).
- If the aggregate size of all the swap partitions is too small to contain the crash dump, the system creates no crash dump.

Each crash dump contains a header, which the system always writes to the end of the primary swap partition. The header contains information about the size of the dump and where the dump is stored. This information allows `savecore` to find and save the dump at system reboot time.

In most cases, compressed dumps will fit on the primary swap partition. The next section describes `dump_sp_threshold`, which is relevant in understanding how a crash dump is created. The use of the remaining kernel attributes controls the content of the dump. These attributes are described in Section 14.3.

### Controlling the Use of Swap Partitions

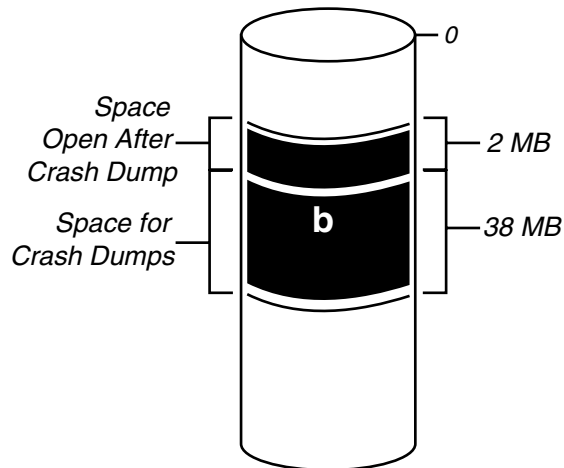
You can configure the system so that it fills the secondary swap partitions with dump information before writing any information (except the dump header) to the primary swap partition. The attribute that you use to configure where crash dumps are written first is the `dump_sp_threshold` attribute.

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. (40 MB is not typical of a swap partition size on most systems, the example uses small numbers for the sake of simplicity.)

**Figure 14–1: Default `dump_sp_threshold` Attribute Setting**

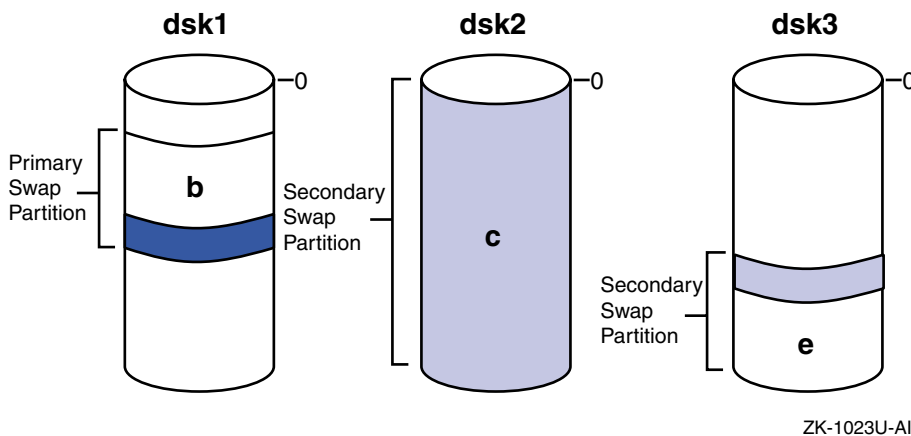


ZK-1024U-AI

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. 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 `dumpsys` command, which produces a run time or continuable dump. See Section 14.4.1 for information on using the `dumpsys` command. You might need to temporarily create file system space to hold the experimental dumps. You can produce both full and partial dumps using this method. Crash dumps are compressed by default unless you specify the `dumpsys -u` command option. You use the `expand_dump` command to produce a noncompressed dump from the compressed output of the `dumpsys` command.

Because the crash dumps written to swap are about the same size as their resulting saved crash dump files, you can easily determine how large a crash dump was by examining the size of the resulting crash dump file. For example, to determine the size of the first crash dump file created by your system, enter the following command:

```
# ls -s /var/adm/crash/vmzcore.0
20480 vmzcore.0
```

This command displays the number of 512-byte blocks occupied by the crash dump file. In this case, the file occupies 20,480 blocks, so you know that a crash dump written to the swap partitions also occupies about 20,480 blocks.

In some cases, a system contains so much active memory that it cannot store a crash dump on a single disk. For example, suppose your system contains 2 GB of memory but only has several 4 GB disks, most of which are dedicated to storing data. Crash dumps for this system might be too large to fit on a single swap partition on a single device. To cause crash dumps to spread across multiple disks, create a second (and perhaps tertiary) swap partitions on several disks. The system automatically writes dumps that are too large to fit in the specified portion of the primary swap partition to other available swap partitions.

## 14.2.5 Planning Crash Dump Space

Because crash dumps are written to the swap partitions on your system, you allow space for crash dumps by adjusting the size of your swap partitions, thereby creating temporary or permanent swap space. See `swapon(8)` for information about modifying the size of swap partitions.

---

### Note

---

Be sure to list all permanent swap partitions in the `/etc/sysconfigtab` file. The `savecore` command, which copies the crash dump from swap partitions to a file, uses the information in the `/etc/sysconfigtab` file to find the swap partitions. If you omit a swap partition from the `/etc/sysconfigtab` file, the `savecore` command might be unable to find the omitted partition.

---

Space requirements can vary from system to system. For example, during the installation procedure uses the following algorithm to calculate required space in the `/var` file system:

```
3 * memsize / 24MB + 3 * 15MB
```

Where `memsize` is the amount of physical memory in megabytes and `15MB` is the approximate size of a custom kernel. This algorithm allows for the preservation of three dumps. The sections that follow give you guidelines for estimating the amount of space required for partial and full crash dumps on your system. In addition, setting the `dump_sp_threshold` attribute is described.

## 14.2.6 Planning and Allocating File System Space for Crash Dump Files

Using the information on typical crash dump sizes for your system, you can plan and allocate the file system space that you need for the `/var/adm/crash` directory.

For example, suppose you save partial crash dumps. Your system has 96 MB of memory and you have reserved 85 MB of disk space for crash dumps and swapping. In this case, you should reserve 20 MB of space in the file system for storing crash dump files. You need to reserve considerably more space if you want to save files from more than one crash dump. If you want to save files from multiple crash dumps, consider archiving older crash dump files. See Section 14.5 for information about archiving crash dump files.

By default, the `savecore` command writes crash dump files to the `/var/adm/crash` directory. To reserve space for crash dump files in the default directory, you must mount the `/var/adm/crash` directory on a file system that has a sufficient amount of disk space. (For information about mounting file systems, see Chapter 6 and `mount(8)`.) If you expect your crash dump files to be large, you 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. When in single-user mode, you can make space available in the crash directory or change the crash directory. One possibility in this situation is to issue the `savecore` command at the single-user mode prompt. On the command line, specify the name of a directory that contains a sufficient amount of file space to save the crash dump files. For example, the following `savecore` command writes crash dump files to the `/usr/adm/crash2` directory:

```
# savecore /usr/adm/crash2
```

After the `savecore` command has saved the crash dump files, you can bring 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, the `savecore` command copies the crash dump to files in the default `/var/adm/crash` directory.

You can control the default location of the crash directory 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 are not helpful in finding the cause of the crash.

A partial crash dump contains the following:

- The crash dump header
- A copy of part of physical memory

The system writes the part of physical memory believed to contain significant information at the time of the system crash, basically kernel node code and data. By default, the system omits user page table entries.

A full crash dump contains the following:

- The crash dump header
- A copy of the entire contents of physical memory at the time of the crash

You can modify how crash dumps are taken:

- By adjusting the crash dump threshold
- By overriding the default so that the system writes user page table entries to partial crash dumps
- By selecting partial or full crash dumps
- By revising the expected dump compression
- By selecting compressed or noncompressed crash dumps

These options are explained in the following sections.

### 14.3.1 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. The value you assign to this attribute indicates the amount of space that you normally want available for system swapping after a system crash, as described in Section 14.2.

To adjust the `dump_sp_threshold` attribute, issue the `sysconfig` command. For example, suppose your primary swap partition is 40 MB. To raise the value so that the system writes crash dumps to secondary partitions, issue the following command:

```
# sysconfig -r generic dump_sp_threshold=81920
```

In the preceding example, the `dump_sp_threshold` attribute, which is in the `generic` subsystem, is set to 81,920 512-byte blocks (40 MB). In this example, the system attempts to leave the entire primary swap partition open for system swapping. The system automatically writes the crash dump to secondary swap partitions and the crash dump header to the end of the primary swap partition.

The `sysconfig` command changes the value of system attributes for the currently running kernel. To store the new value of the `dump_sp_threshold` attribute in the `sysconfigtab` database, modify that database by using the `sysconfigdb` command. For information about the `sysconfigtab` database and the `sysconfigdb` command, see `sysconfigdb(8)`.

---

**Note**

---

After the `savecore` program has copied the crash dump to a file, all swap devices are immediately available for mounting and swapping. The sharing of swap space only occurs for a short time during boot, and usually on systems with a small amount of physical memory.

---

### 14.3.2 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, your technical support person might instruct you to include user page tables for crash dump analysis.

To include user page tables in partial crash dumps, set the value of the `dump_user_pte_pages` attribute to 1. The `dump_user_pte_pages` attribute is in the `generic` subsystem. The following example shows the command you issue to set this attribute:

```
# sysconfig -r generic dump_user_pte_pages = 1
```

The `sysconfig` command changes the value of system attributes for the currently running kernel. To store the new value of the `dump_user_pte_pages` attribute in the `sysconfigtab` database, modify that database by using the `sysconfigdb` command or use the `dxkerneltuner` GUI.



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
```

You can also use the `dxkerneltuner` GUI or the `sysconfigdb` command to modify kernel entries and preserve the modifications across reboots. To return to partial crash dumps, reset the `partial_dump` variable to 1.

When partial dumps are enabled, you can also enable the `dump_kernel_text` attribute to include kernel text pages.

### 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 the `dumpsys` command, as described in Section 14.4.1. Using the `ls -s` command, record the size of this dump as value `a`.
2. Use the `expand_dump` command to produce a noncompressed version of the dump. Using the `ls -s` command, record the size of this dump as value `b`.
3. Divide `a` by `b` to produce the approximate compression ratio.
4. Repeat the previous steps several times and choose the largest value of the compression ratio. Multiply the compression ratio by 1000 to produce an expected dump value.
5. Add 10 percent of the expected dump value to create a value for the `expected_dump_compression` variable.

Set the kernel's `expected_dump_compression` variable to the required value using the `sysconfig` command as follows:

```
# sysconfig -r generic expected_dump_compression=750
expected_dump_compression: reconfigured
# sysconfig -q generic partial_dump
generic:
expected_dump_compression=750
```

You can also use the `dxkerneltuner` GUI or the `sysconfigdb` command to modify kernel entries and preserve 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 realtime operations. On reboot after a crash, the `savecore` command runs automatically and detects that the dump is compressed, using information in the crash dump header in the swap partition. It then copies the crash dump file from the swap partition to the `/var/adm/crash` directory. The compressed crash dump files are identified by the letter `z` in the file name, to distinguish them from noncompressed crash dump files. For example: `vmzcore.1`.

You can use this type of compressed crash dump file with some debugging tools such as `dbx`, which is not true of the type of compression produced by tools such as `compress` or `gzip`. If you need to use a tool that does not support compressed crash dump files, you can convert it to a conventional noncompressed format with the `expand_dump` utility. The following example shows how you use the `expand_dump` utility:

```
# expand_dump vmzcore.2 vmcore.2
```

You might want to disable compressed dumps if you always use tools or scripts that do not work with the compressed format, and it is not convenient to use the `expand_dump` command. To disable compressed dumps, use the following `sysconfig` command:

```
# sysconfig -r generic compressed_dump=0
```

The preceding command temporarily changes the mode of dumping to noncompressed and the mode reverts to compressed dumps on the next reboot. To make the change persistent, use the `sysconfigdb` command to update the value of the `compressed_dump` attribute in the `/etc/sysconfigtab` file or use the `dxkerneltuner` GUI to modify the value in the `generic` subsystem.

---

**Note**

---

Memory dumps must be compressed. If the `compressed_dump` system attribute is not set, the system automatically enables compression before attempting to write a memory dump.

---

See `savecore(8)`, `expand_dump(8)`, and `sysconfig(8)` for more information on crash dump compression and how to produce a noncompressed crash dump file.

### 14.3.6 Dumping to Exempt Memory

Exempt memory is a region of physical memory that is set aside for a specific purpose. You can create an exempt region of memory by specifying it in the `/etc/sysconfigtab` file. This will cause the exempt region to be created when the system boots. For example:

```
cma_dd:
  CMA_Option = Size-0x3000000, Alignment - 0, /
  Addrlimit - 0x4000000, Type - 0x96, Flag-0
```

The preceding `/etc/sysconfigtab` file entry reserves a region of exempt memory that is 48MB in size. Its `Type` is specified as `M_EXEMPT` by the value `0x96`, the value of `Addrlimit` sets the starting position of the exempt region, which at `0x4000000` is 64MB into physical memory. Each time the machine boots, it will attempt to reserve this same area of physical memory, making it unavailable for any other use.

Another way of creating exempt regions of memory is by using the `contig_malloc()` function call with the type `M_EXEMPT` in a pseudodevice driver. See the `malloc.h` file for information on the `M_EXEMPT` type. See `config_malloc(9r)` for information on using the function call.

You can use the `vmstat` command with the `-M` option to examine exempt memory regions.

To dump to exempt memory, the `dump_to_memory` attribute must be enabled as described in Section 14.2.2. You also configure the following attributes as required:

- `dump_exmem_size` – Specifies the size (in bytes) of the exempt memory region to which dumps are written. By default, the value is 0 (zero), which disables writing a dump to an exempt memory region.
- `dump_exmem_addr` – Identifies the starting address (virtual or physical) for a region of exempt memory used for writing primary dumps.

- `dump_exmem_include` – Specifies whether or not exempt memory pages are included in the dump. By default, the value is 0 (zero) and exempt memory pages are excluded.

The setting of the `dump_exmem_addr` attribute has no effect unless you also configure the `dump_exmem_size` attribute. Ensure that you keep a record of any run-time settings for the attributes so that you will be able to find the crash dump after recovery from a system failure.

The following example shows how you reconfigure these attributes:

```
# sysconfig -q generic dump_to_memory
generic:
dump_to_memory = 0
# sysconfig -r generic dump_to_memory=1
dump_to_memory: reconfigured
# sysconfig -q generic dump_to_memory
generic:
dump_to_memory = 1
```

Memory dumps are always compressed by default. The `compressed_dump` system attribute is automatically enabled if it is not set to on. The `savecore` command uses the `vmzcore` character special device file to recover the compressed dumps. See `savecore(8)` and `vmzcore(7)` for more information.

### 14.3.7 Dumping to a Remote Host

Use the `savecore` command with the `-r` option to write crash dump files from a client host to a remote host using an ftp connection. You can specify either of the following definitions for a remote destination:

- The name of the remote host and a valid account and password
- The path to a configuration file containing the ftp connection and login information

For example, the following command specifies a login to the remote host in verbose mode, which enables you to debug the ftp connection.

```
# savecore -r -v soserv:jeffdump:Cr$hDeBuG
```

When it connects to the target host, the `savecore` utility directs the remote `ftpd` server daemon to create a directory named after the client host name. The crash dump files (`bounds`, `msgbuf.savecore`, `evm.buf`, `vmunix.N`, and `vmcore.N` or `vmzcore.N`) are written to the directory. You must ensure that you have adequate space for the crash dump on the remote device.

See `savecore(8)` and `ftpd(8)` for more information and for restrictions when using this feature.

## 14.4 Generating a Crash Dump Manually

The following sections describe 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 `crash` console command 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 the `dumpsys` command takes a snapshot, memory might change as its content is copied. Analysis of the resulting dump can often demonstrate incomplete linked lists and partially zeroed pages, which are not problems, but reflect the transitory state of memory. For this reason, some system problems cannot be detected by using the `dumpsys` command and you might 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 the `/var/adm/crash` directory.

The `/var/adm/crash/minfree` text file specifies the minimum number of kilobytes that must be left on the file system after the `dumpsys` command copies the dump. By default, this file does not exist, indicating that no minimum is set. To specify a minimum, create the file and store the number of kilobytes you want reserved in it. You can override the setting in the `minfree` file by using the `-i` option. The `-s` option displays the approximate number of disk blocks that full and partial dumps 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 from a system with 512 KB of physical memory. The examples show a noncompressed crash dump. Dumps are usually compressed by default:

```
# dumpsys -s
Approximate full dump size = 1048544 disk blocks,
```

```
if compressed, expect about 524272 disk blocks.
Approximate partial dump size = 94592 disk blocks,
if compressed, expect about 47296 disk blocks.

# dumphsys -i /userfiles
Saving 536797184 bytes of image in /userfiles/vmzcore.0
# ls /userfiles
bounds vmzcore.0 vmunix.0
```

Two attributes in the generic kernel subsystem enable you to control continuable dumps:

- `live_dump_dir_name` – Specifies a path to the directory where the continuable dump files are written. The default value is the `/var/adm/crash` directory.
- `live_dump_zero_suppress` – Enables or disables zero compression of continuable dumps. Using this option produces files that take longer to create but occupy less space.

See `dumphsys(8)` and `sys_attrs_generic(5)` for more information. 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. Type `Ctrl/p` at the console prompt.
2. At the console prompt, enter the crash command.

If your system hangs and you force a crash dump, the panic string recorded in the crash dump is the following:

```
hardware restart
```

This panic string is always the one recorded when system operation is interrupted by pressing the Halt button or by typing `Ctrl/p`.

## 14.5 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 are 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 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 `vmcore.n` file is compressed using a standard UNIX compression utility, such as `gzip`. Standard compression applied to a `vmzcore.n` file makes the resulting file about 40 percent smaller than the equivalent `vmzcore.n` file.

If you need to apply the maximum compression possible to a `vmzcore.n` file, follow these steps:

1. Uncompress the `vmzcore.n` file by using the `expand_dump` command (see `expand_dump(8)`). The following example creates an uncompressed file named `vmcore.3` from the `vmzcore.3` file:

```
# expand_dump vmzcore.3
```

2. Compress the resulting `vmcore.n` file using a standard UNIX utility. The following example uses the `gzip` command to create a compressed file named `vmcore.3.gz`:

```
# gzip vmcore.3
```

You can uncompress a `vmzcore.n` file only with the `expand_dump` command. (Do not use `gunzip`, `uncompress`, or any other utility). After you uncompress a `vmzcore.n` file into a `vmcore.n` file by using the `expand_dump` command, you cannot compress it back into a `vmzcore.n` file.

## 14.5.2 Uncompressing a Partial Crash Dump File

This section applies only if you are uncompressing a partial crash dump file that was previously compressed from a `vmcore.n` file.

If you compress a `vmcore.n` dump file from a partial crash dump, you must use care when you uncompress it. Using the `gunzip` or `uncompress` command with no options results in a `vmcore.n` file that requires space equal to the size of memory. In other words, the uncompressed file requires the same amount of disk space as a `vmcore.n` file from a full crash dump.

This situation occurs because the original `vmcore.n` file contains UNIX File System (UFS) file holes. (UFS files can contain regions, called holes, which have no associated data blocks.) When a process, such as the `gunzip` or `uncompress` command, reads from a hole in a file, the file system returns zero-valued data. Thus, memory omitted from the partial dump is added back into the uncompressed `vmcore.n` file as disk blocks containing all zeros.

To ensure that the uncompressed core file remains at its partial dump size, you must pipe the output from the `gunzip` or `uncompress` command with the `-c` option to the `dd` command with the `conv=sparse` option. For example, to uncompress a file named `vmcore.0.Z`, issue the following command:

```
# uncompress -c vmcore.0.Z | dd of=vmcore.0 conv=sparse
```

```
262144+0 records in
```

```
262144+0 records out
```



# A

---

## Administration Utilities

This appendix identifies and defines the administrative utilities and commands.

### A.1 X11 Graphical User Interfaces (CDE Application Manager)

The X11-based graphical utilities (GUIs) are available under the CDE Application Manager or from the command line. In some cases, the GUIs have analogous SysMan utilities or are superseded by a SysMan Menu task. Invoke the CDE applications as described in Chapter 1.

Not all administrative tasks are available as SysMan Menu options. You will need to use a combination of GUIs, SysMan Menu tasks, and commands. With each release, more SysMan Menu options are added and older administrative methods become obsolete. To help you understand your administrative options, Table A-1 to Table A-6 list the utilities. Each table provides the following information:

- The first column identifies the task, which can be a subsystem that you want to configure or an administrative application that you want to run on a system component, such as a disk or a file system. There are three application formats that might appear in this column:
  - X11-compliant Graphical User Interfaces (GUIs), such as the Kernel Tuner (`dxkerneltuner`).
  - SysMan Menu utilities that you can run in different user environments.
  - Command-line scripts. You must install the `OSFRETIREDDXXX` subsets to access some of these scripts, as described in the *Installation Guide*.
- The second column lists the SysMan Menu task. Text in brackets, such as `[dns]`, are the command options that you can use with the `sysman` command to invoke a utility directly from the command prompt. For example:

```
# sysman dns
# sysman dns_client
```

The first command example invokes the submenu of all DNS tasks. The second command invokes the specific utility that you can use to configure the local system as a DNS client.

- The third column lists the commands (command-line options) that perform the equivalent task.

Many SysMan Menu tasks do not have an analogous CDE GUI. The command line provides most functions, but it has limitations. For example, you cannot select and apply changes to multiple user accounts by using the `usermod` command, but you can do it by using the Account Manager GUI (`dxaccounts`).

The following tables are organized by CDE application. The Application Manager provides the following folders containing system administration utilities:

- Configuration – Table A–1 lists the configuration utilities, which are used for initial system configuration and regular system maintenance.
- Daily Admin – Table A–2 lists the daily administration utilities, which are used for routine system administration tasks.
- MonitoringTuning – Table A–3 lists the utilities that are used for monitoring system operation and performance tuning.
- Software\_Management – Table A–4 lists the utilities that are used for 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 the utilities that provide system statistics.

**Table A–1: System Administration Configuration Applications**

Subsystem to Configure	SysMan Menu Option	Command-Line Interface
ATM ( <code>atmsetup(8)</code> )	Set up Asynchronous Transfer Mode (ATM) [ <code>atm</code> ]	<code>atmconfig(8)</code>
CDE Setup ( <code>dtsetup(8)</code> )	none	none
DHCP Server ( <code>xjoin(8)</code> , <code>DHCP(7)</code> )	Network Setup Wizard [ <code>net_wizard</code> ]	none
DNS (BIND) ( <code>bindconfig(8)</code> )	Domain Name Service (DNS (BIND)) [ <code>dns</code> , <code>dns_client</code> , <code>dns_server</code> , <code>dns_deconfigure</code> ]	Retired in Version 5.1
DOP ( <code>dop(8)</code> )	Configure Division of Privileges (DOP) [ <code>dopconfig</code> ]	<code>dop(8)</code>

**Table A-1: System Administration Configuration Applications (cont.)**

<b>Subsystem to Configure</b>	<b>SysMan Menu Option</b>	<b>Command-Line Interface</b>
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 (NFS) [nfs]	Retired in Version 5.1
NIS (nissetup(8))	Configure Network Information Service (NIS) [nis]	none
NTP	Network Time Protocol (NTP) [ntp, ntp_config, ntp_status, ntp_start, ntpstop]	Retired in Version 5.1
Network	Network Setup Wizard [net_wizard]	none
PPP (pppd(8))	Point-to-Point Protocol (PPP) [ppp]	none
Print (printconfig(8))	Configure Line Printers [lprsetup]	lprsetup(8)
SLIP	Serial Line Networking [serial_line]	startslip(8)
Audit Configuration	Audit Configuration [auditconfig]	none
Security	Security Configuration [seconfig]	none
latsetup	Configure Local Area Transport (LAT) [lat]	latsetup(8), lat_man- ual_setup(8)

**Table A-2: System Administration Daily Admin Applications**

<b>CDE Administrative Task</b>	<b>SysMan Menu Option</b>	<b>Command-Line Interface</b>
Account Manager (dxaccounts(8)). See also features offered under Advanced Server for UNIX (ASU)	Accounts [accounts, users, groups, nis_users, 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

**Table A-2: System Administration Daily Admin Applications (cont.)**

<b>CDE Administrative Task</b>	<b>SysMan Menu Option</b>	<b>Command-Line Interface</b>
Display Window (dxdw(8))	none	iostat(1), netstat(1), vmstat(1), who(1)
Event Viewer	View Events [event_viewer]	evmget(1), evmshow(1), evmpost(1), an other associated commands, See EVM(5)
File Sharing (dxfileshare(8))	Share Local Directory (etc/exports) [export]	mount(8), automount(8), exports(4)
Get/Set ACL (dxsetacl(8))	none	none
Host Manager (dxhosts(8))	none	none
License Manager (dxlicenses(8))	Register License Data [lmfsetup]	lmf(8), lmfsetup(8)
Mail User Admin (mailusradm(8))	none	none
Power Management (dxpather(8))	none	sysconfig(8)
SysManShutdown	Shut Down the System [shutdown]	shutdown(8)
System Information (dxsysinfo(8))	none	du(1), df(1), swapon(8)

**Table A-3: System Administration Monitoring and Tuning Applications**

<b>CDE Administrative Task</b>	<b>SysMan Menu Option</b>	<b>Command-Line Interface</b>
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)
Insight Manager	Set Up Insight Manager [imconfig]	none

**Table A-3: System Administration Monitoring and Tuning Applications (cont.)**

<b>CDE Administrative Task</b>	<b>SysMan Menu Option</b>	<b>Command-Line Interface</b>
Kernel Tuner (dxkerneltuner(8))	none	sysconfig(8), sysconfigdb(8)
Process Tuner (dxproctuner(8))	none	nice(1), renice(8), ps(1), kill(1)

**Table A-4: System Administration Software Management Applications**

<b>CDE Administrative Task</b>	<b>SysMan Menu Option</b>	<b>Command-Line Interface</b>
Software Management	Installation, [install, setldload, setldlist, setlidd].	setld(8)
Update Installation Cleanup	Cleanup After an OS Update (updadmin), [updadmin]	updadmin(8)

**Table A-5: System Administration Storage Management Applications**

<b>CDE Administrative Task</b>	<b>SysMan Menu Option</b>	<b>Command-Line Interface</b>
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(8X))	Logical Storage Manager (LSM) Utilities [lsm, volsetup].	volsetup(8)
Prestoserve I/O Accelerator (dxpresto(8X))	Configure the Prestoserve software [presto]	prestosetup(8))

**Table A–6: System Administration Tools**

<b>CDE Administrative Task</b>	<b>SysMan Menu Option</b>	<b>Command-Line Interface</b>
I/O Statistics (dxdw(8))	View Input/Output (I/O) statistics [iostat]	iostat(1)
Network Statistics (dxdw(8))	none	netstat(1)
System Messages (dxdw(8))	See the Event Viewer	syslogd(8)
Virtual Memory Statistics (dxdw(8))	View Virtual Memory (VM) Statistics [vmstat]	vmstat(1)

## A.2 SysMan Menu Tasks and Associated Utilities

The tables in Section A.1 identify the GUIs and command-line utilities that perform functions similar to the various SysMan Menu tasks. The following SysMan Menu utilities are available. You can use the accelerator keyword, such as [accounts], with the `sysman` command to launch a utility from the command line:

- **Accounts [accounts]** – Enables you to maintain user accounts and manage system resources. Refer to Chapter 7 for information on administering user accounts. The Accounts tasks provided are as follows:
  - **Manage Local Users [users]** – Administer the `/etc/passwd` file, which records user accounts data
  - **Manage Local Groups [groups]** – Administer the `/etc/group` file, which records user resource access data
  - **Manage NIS Users [nis\_users]** – Administer NIS user accounts
  - **Manage NIS Groups [nis\_groups]** – Administer NIS user groups
- **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 and all devices attached to the buses. See `hwmgr(8)`.
  - **View Cluster [hw\_cluhierarchy]** – Displays the hierarchy of all members of a cluster.
  - **View Device Information [hw\_devices]** – Displays a list of all devices (such as disks) attached to the system. See `hwmgr(8)`.

- View central processing unit (CPU) information [hw\_cpus] – Displays the type of processors on the system and their status, such as time on line.
- Manage CPUs [hw\_manage\_cpus] – Manage CPUs on multiprocessor systems.
- Online Addition/Replacement (OLAR) policy information [hw\_olar\_policy\_info] – Administer the policy information for removal and addition of components. See `olar_config(4)`.
- Mail [mail] – Enables you to configure e-mail and manage mail accounts. The mail tasks provided are as follows:
  - Configure Mail [mailsetup] – Enables you to configure the electronic mail services on the system.
  - Manage Users' Mail Accounts [mailusradm] – Enables you to set up 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 `class_admin(8)` and Chapter 3 for more information. This task provides the following utilities:
    - Configure Class Scheduler [class\_setup] – Enables you to create scheduling databases that govern the use of system resources (such as CPU time) by processes. You can 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 `vmstat(1)` for more information.
  - View Input/Output (I/O) Statistics [iostat] – Enables you to monitor I/O (input/output) statistics. See `iostat(1)` for more information.

- View Uptime Statistics [uptime] – Enables you to monitor how long the system has been up, and determine the average workload has been since the last boot. See `uptime(1)` 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: Services* 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:
    - Configure asynchronous transfer mode (ATM) [atm] – Set up ATM services and configure ATM adapters. See the *Network Administration: Connections* guide for more information.
    - 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. Static routes are the most common form of communication with local and remote networks. See the *Network Administration: Services* guide for more information.
    - Set Up Routing Services (`gated`, `routed`, IP router) [routing] – Enables you to configure the network to use a particular method of routing. Your options are Gateway Routing Daemon (`gated`), Routing Daemon (`routed`), or an Internet Protocol (IP) Router. See the *Network Administration: Services* 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.
  - Additional Network Services [networkadditional] – A set of utilities you can use to configure other networking features. This task provides the following utilities:
    - Domain Name Service (DNS (BIND)) [dns] – Configure domain name services on the local system. The following utilities are provided:
      - Configure System as DNS server [dns\_server]
      - Configure System as DNS client [dns\_client]
      - Deconfigure DNS on This System [dns\_deconfigure]
    - Serial Line Networking [serial\_line] – Enables you to configure the following serial-line networking options:
      - Point-to-Point Protocol (PPP) [ppp] – Enables you to configure PPP, including the following tasks:
        - Create Option Files [ppp\_options]
        - Modify pap-secrets File [pap]
        - Modify chap-secrets File [chap]
 See the *Network Administration: Connections* guide for more information.
    - Configure System for UNIX-to-UNIX Copy (uucp) connections [uucp] – Enables you to configure UUCP over a modem, TCP/IP, or hardwired connection. See the *Network Administration: Services* 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: Services* guide for more information. The following utilities are available:
    - Configure System as an NTP client [ntp\_config]
    - View Status of NTP Daemon [ntp\_status]
    - [Re]start NTP Daemon [ntp\_start]
    - Stop NTP Daemon [ntp\_stop]
  - Network File System (NFS) [nfs] — Enables you to configure Network File System, and share file systems between hosts. See the *Network Administration: Services* 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]
- Start The NFS Daemons [nfs\_start]
- Stop NFS Daemons [nfs\_stop]
- Configure Network Information Service NIS [nis] – Enables you to allow users to use the resources of networked systems, such as logging in to different hosts. User names and passwords are distributed between hosts. See the *Network Administration: Services* guide for more information.
- Configure Local Area Transport (LAT) [lat] – Enables you to set up LAT. See the *Network Administration: Connections* 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`.
- Start or Restart Network Services [inet\_start] – Enables you to start or restart any stopped networking daemons such as `gated` or `rwhod`.
- Stop Network Services [inet\_stop] – Enables you to stop all network services.
- 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 `dop(8)` for more information.
  - Manage DOP Actions [dopactions] – Enables you to create, modify, or delete DOP actions and their associated privileges.
  - Security Configuration [secconfig] – Enables you to configure base or enhanced security.
  - Audit Configuration [auditconfig] – Enables you to set up and start security auditing.

- Software [software] – Enables you to manage operating system and layered software installations and updates. This task provides the following utilities:
  - Installation [install] – Enables installation of the operating system and components. This task provides the following utilities:
    - Install Software [setldload] – Enables you to add software to the system from a RIS server or from the distribution media (CD-ROM).
    - List Installed Software [setldlist] – Enables you to list the software that is currently installed on the system.
    - Remove Installed Software [setldd] – Enables you to permanently remove software from the system.
    - Clean Up After an OS Update [updadmin] – Enables you to remove unnecessary files from the system to save space or archive files to tape after running an installation update (updateinstall).
  - Register License Data [lmfsetup] – Enables you to register software product authorization keys (PAKs).
- Storage [storage] – Enables you to administer file systems and data storage. The following utilities are provided:
  - File Systems Management Utilities [filesystems] – Enables basic administration of disk storage. See Chapter 6 for more information. The following utilities are provided:
    - General File System Utilities [generalfs] – Provides utilities that you can use with either UFS or AdvFS. This task provides the following utilities:
      - Dismount a File System [dismount]
      - Display Currently Mounted File Systems [df]
      - Mount File Systems [mount]
      - Share Local Directory (/etc/exports) [export]
      - Mount Network Directory (/etc/fstab) [net\_mount]
    - Advanced File System Utilities [advfs] – Enables you to perform basic administration tasks on AdvFS domains. See `advfs(4)` and *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) Administration [lsm] – Enables you to perform basic administration of Logical Storage Manager (LSM) volumes. See the *Logical Storage Manager* guide for more information. The following utilities are provided:
  - Initialize the Logical Storage Manager (LSM) [volsetup]
  - Initialize the Logical Storage Manager (LSM) Administrator [lsmmgr]
- Create a Bootable Tape [boot\_tape] – Enables you to create a standalone kernel on a bootable tape, which can assist in disaster recovery. See the `btcreate(8)`, `btextract(8)` and Chapter 9 for more information.
- Identify SAN Appliances Wizard [idsanappl] – Enables you to identify storage area network (SAN) management appliances and add them to the SysMan Station.
- Support and Services [support] – Enables you to run preconfigured `sys_check` system census tasks as part of troubleshooting and error recovery, or in case you need to escalate a problem to your technical support organization. See Chapter 3 and `sys_check(8)` for information. The following utilities are available:
  - Create Escalation Report [escalation] – Enables you to prepare a system census report for delivery to your technical support organization.
  - Create Configuration Report [config\_report] – Enables you to prepare a system census report for baseline, troubleshooting, or tuning purposes.
- General Tasks [general\_tasks] – Provides you with a set of miscellaneous administrative utilities as follows:

- Shut Down the System [shutdown] – Enables you to perform managed shutdowns of the system. See `shutdown(8)` and Chapter 2 for more information.
- Quick Setup [quicksetup] – Runs the basic system setup wizard, which guides you through typical basic system configuration. See Chapter 1 for a description of the features.
- Configure Prestoserve Software [presto] – Enables you to configure Prestoserve. Refer to the *Guide to Prestoserve* for more information.
- Configure X Display Manager [xsetup] – Enables you to select CDE or XDM as the default windowing environment.
- Cloning Setup Information [cloneinfo] – Displays information on using the `sysman -clone` command to clone your system's configuration and apply it to other systems. 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.)**

<b>Category</b>	<b>Mnemonic</b>	<b>Description</b>
Terminals	pty	Network pseudoterminals Refer to /dev
Modems		See the QuickSpec for a list of supported devices and refer to the owner's 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 *QuickSpec* of a given release of the operating system. Only a small number of cards are 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 are using.
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 is installed and a card is 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.

See `pcmcia(7)` for specific information on configuring cards and on any current restrictions in a given release. See `modem(7)` and the *Network Administration: Connections* 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 by using the following syntax:

```
device:mode:tabletWidth:tabletHeight:numbtms:corePointer:mouseScale:\
resolution:Xincrement:Yincrement
```

(See `calcomp(7)` 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 restart completes, the tablet is configured into the Xserver and ready to use.

See `calcomp(7)` 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. Only the instance of an operating system that is running on a partition can access that partition's hardware resources.

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

Only the AlphaServer GS140 6–525 is supported. See 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 value of 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). See 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 might include a second IOP module. The maximum number of IOP modules for the entire system (the sum of all partitions) is three.

- XMI hardware might 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 might 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, see the *Release Notes* for the operating system release that you are installing, and update the firmware revision if required. See 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 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 might have several logical partitions, each running a separate instance of the operating system.

##### **primary partition**

Partition number zero. The partition with the active console terminal if 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 if partitions are disabled.

##### **secondary console**

The console terminal connected to a secondary partition. This is active only if 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 causes a complete initialization of the system. All secondary partitions are immediately terminated. The primary partition displays the normal power on self-test messages and enters console mode.

### console prompt

The prompt displayed on the console terminal of a partition to indicate the console firmware is ready to accept commands, as follows:

```
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

Holding down the control key and typing the letter `p` causes the primary processor for partition 0 to halt and enter console mode (`P00>>>` prompt). This is possible only on the primary console. 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##>>>`) causes 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 causes 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 causes the processor to resume program execution. For example:

```
P##>>>continue 4
```

If you halt a single processor you can omit the processor number (N).

```
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 is momentarily moved to the RESET position). If partitions are enabled, the console requests 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 might be skipped in certain cases.

If you have read this section previously, and require only 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 preceded 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`, automatically boots 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 you cannot interrupt the automatic boot, allow the operating system boot completely, then shut it down (do not type `ctrl/p` to halt the automatic boot). See the *Installation Guide* guide for information on factory installed software before you attempt to set up logical partitions.

The factory installed software disk might 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 is displayed 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
SR0M 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 provides up to nine slots, each of which is labeled 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 includes 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. See 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 displays 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, see 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.

Set the processor mask variable (`lp_cpu_mask`) is set shifting the number 3 by two times the slot number of the CPU module. Possible CPU masks for each slot are as follows:

```
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
```

Calculate the value of the `lp_cpu_mask` variable 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, you assign the `lp_cpu_mask0` variable a value of 00f.

Set the IO port mask variable (`lp_io_mask`) by left shifting the number 1 by the slot number of the IOP module. Potential IOP masks for each slot are as follows:

```
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 consists of two IOP modules, create the value of the `lp_io_mask` variable 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 the `lp_io_mask1` variable is 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) are assigned to each partition (using the module types and slot numbers you recorded previously). Then, you can create the console EVs.

A summary of console EVs and values follows:

Console EV	Value
<code>lp_count</code>	Number of partitions
<code>lp_cpu_maskN</code>	CPU assignment mask for partition <i>N</i>
<code>lp_io_maskN</code>	IOP module assignment mask for partition <i>N</i>
<code>lp_mem_mode</code>	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 attempts 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.

The value of the `lp_count` EV is zero (it changes later).

The following command displays the console EVs if you have created them. No output appears if the console EVs 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.

There is a 10 second delay after you issue each command and that the console echoes 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 might 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, ignore the next section and proceed to 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**

---

You must set console EVs with `lp_` prepended to the EV name by using only the console of the primary partition (partition 0). You must not change the value of these variables 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 the `boot_reset` console EV is set ON, then a system-wide reset happens after you execute the `boot` command (`P00>>>boot`). This reset immediately terminates the operation of all partitions.

Execute the following command to disable the `boot_reset` console EV:

```
P00>>>set boot_reset off
```

#### D.4.3.8 Setting Memory Interleave Mode

Set the value of the `interleave` console EV to none. When setting the interleave mode to none, the console might echo a series of Y characters to the console display screen (there might be several lines of Y characters). This is not a problem and you can ignore it.

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 Setting the Operating System Type to UNIX

Set the value of the `os_type` console EV to UNIX as follows:

```
P00>>>set os_type UNIX
```

#### D.4.3.10 Setting 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 value of 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) is displayed 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 are 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 is displayed on the secondary console screens, followed by a console prompt such as P04>>>. There is 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 displays 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

After the 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 D.4.3.1.

---

**Note**

---

Depending on how you assigned IOP modules, the name of the factory installed software (FIS) disk might 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 might not always be practical to configure a CD-ROM drive on all partitions and a RIS server might 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 might 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 prompts 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 might leave the console boot drivers in an inconsistent state. In this case, the console displays 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 halts and enters console mode after you shut down the operating system. You must 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 work only if the Power OFF/ENABLE switch is in the ENABLE position.

#### Primary Partition

Pressing Ctrl/p on the primary console terminal forces 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 do 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 prevents you from 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. You can change these console EVs only 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, 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 you configure and enable logical partitions, the operating system displays informational messages for each partition. These messages appear on the console terminal during the early stages 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). You cannot perform the following hardware management and maintenance tasks 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 tell the Customer Services representative that the system is partitioned (particularly if the service operation uses 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. You can run the `sizer -P` command on any instance of the operating system to 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
```

You might 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 might be required. Use the procedure in Section D.4.9 to determine if you need to rebuild the kernel for any partition.

If you do not require a kernel rebuild, 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

If you change your system's hardware configuration you might need to rebuild the kernel. The following table defines the hardware configuration changes that require a rebuilt kernel:

Change	Requirements
Processors -- adding, removing, or reassigning CPU modules.	Changing the <code>lp_cpu_mask#</code> EV for any partition does not require a kernel rebuild. You must assign to the same partition both processors on a dual CPU module.
IO Processors -- adding, removing, or reassigning IOP modules.	Rebuild the kernel if you added or removed an IOP module. You need only rebuild the kernel for the changed partition). Moving a IOP module across partitions requires a kernel rebuild on both partitions. The <code>lp_io_mask#</code> EV assigns IOP modules.  Adding or removing IO busses and IO controllers requires 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 change this address and require a kernel rebuild. A partition's memory starting address changes if the memory size for any lower numbered partition increases or decreases.  For example, if you replaced a 2GB memory module in partition zero with a 4GB module, the memory starting address of partition one increases by 2GB. In this example you must rebuild the kernel.  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 <code>P00&gt;&gt;&gt;lpinit</code> 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 you initialized partitions as described in Section D.4.4 and the partition requiring a kernel rebuild is halted 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 crashes with a panic message, such as the following:

```
panic (cpu 0): tlasr: \  
MACHINE CHECK Non-recoverable hardware error
```

The system then writes out a crash dump, and reboots or halts (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 forces a system reset after writing the crash dump. After the reset is completed, if `auto_action` is set to `BOOT`, the console firmware automatically reinitializes all partitions. Boot the operating system in each partition, using the following commands:

```
P00>>>boot  
P##>>>boot
```

Otherwise, the system halts and enters 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 unconditionally halts 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 console displays the following prompt:

```
Do you really want to reset ALL partitions? (Y/<N>)  
Type Y to perform the reset.
```
4. After the reset is complete, and if `auto_action` is set to `BOOT`, the console firmware automatically reinitializes 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. Repeat the recovery procedure after the reset completes. 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) the partition's console terminal displays an error message. After displaying the error message, the primary processor for the current partition halts and returns 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. This partition has no allocated processors.

```
LP_ERROR: no IOP for partition (check lp_io_mask)
```

The value of `lp_io_mask#` (`#` represents the current partition number) is set incorrectly. This partition has no allocated IO Port modules.



```
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 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. The partition's console displays one of the following messages if an anomaly is detected:

```
Do you really want to reset ALL partitions? (Y/<N>)
```

This message displays after a system reset is 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 terminates all partitions and the system is reset. 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 initializing logical partitions (by running the `lpinit` command automatically). An auto-starting event occurs after a system reset (or power on). The console firmware boots the operating system in all partitions if the `auto_action` console EV is set to BOOT and you perform, the reset by using 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. Change the setting to none.

Insufficient memory modules to partition system.

Each partition requires a dedicated memory module. Reduce the number of partitions or install a memory module for each partition.

This message indicates that the `lp_count` console EV might not be 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 caused the console boot drivers to enter an 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 might 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 (#) might 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 might be set to zero or to incorrect CPU numbers. You should correct the setting and retry the `lpinit` command.

---

# Index

## A

---

**ac command**, 10–17

**account**, 7–1, 7–9

adding, 7–1, 7–15

adding automatically, 7–15

adding manually, 7–28

adding with dxaccounts, 7–24

adduser command, 7–8

administering, 7–15

administration methods, 7–6

ASU, 7–9

checking group file, 7–6

checking passwd file, 7–6

chfn, chsl, 7–6

command-line utilities, 7–8

copying with dxaccounts, 7–26

default UNIX characteristics, 7–3

defaults, 7–1, 7–13, 7–15

deleting, 7–1, 7–15

deleting with dxaccounts, 7–25

directory, 7–12

documentation resources, 7–4

dxaccounts utility, 7–6

enhanced security, 7–3

finding with dxaccounts, 7–25

gecos, 7–12

GID, 7–11, 7–14

group, 7–8, 7–14

group file, 7–9, 7–14

group membership, 7–1, 7–15

group name, 7–14

group password, 7–14

lock is running, 7–4

logs, 7–10

modifying, 7–1, 7–15

modifying data, 7–6

modifying with dxaccounts, 7–24

NIS, 7–3, 7–7, 7–10

passwd command, 7–6

passwd file, 7–9, 7–11

password, 7–1, 7–10, 7–15

PC, 7–1, 7–15

quotas, 7–6

security, 7–10

selecting with dxaccounts, 7–25

shell, 7–13

skeleton files, 7–10

SysMan, 7–6

SysMan Menu, 7–7

system files, 7–9

UID, 7–12, 7–15

user name, 7–11

useradd command, 7–8

userdel command, 7–8

usermod command, 7–8

using dxaccounts, 7–23

utilities and commands, 7–5

Windows 2000, 7–41

### **account administration**

SysMan, A–6

### **Account Manager**, 7–6

### **accounting**, 10–2

automatic, 10–10, 10–12

charge units, 10–33

charging fees, 10–33

commands

ac, 10–17

acctcms, 10–27

acctcom, 10–24

- acctcon1, 10-18
- acctdisk, 10-33
- acctdusg, 10-32
- acctmerg, 10-39
- accton, 10-23
- acctprc1, 10-28
- acctprc2, 10-29
- acctwtmp, 10-17
- diskusg, 10-31
- fwtmp, 10-16
- last, 10-20
- lastcomm, 10-30
- list of, 10-3
- sa, 10-26
- wtmpfix, 10-15
- connect session, 10-13
- daily records, 10-8
- daily reports, 10-8
- disk samples, 10-39
- disk usage, 10-30
- error messages, 10-38
- files
  - administrative, 10-5
  - daily, 10-6
  - daily summary, 10-8
  - database, 10-5
  - extraneous, 10-5
  - monthly, 10-9
  - printer summary, 10-34
  - printer use, 10-34
- monitoring system usage, 10-2
- monthly reports, 10-9
- nonprime time, 10-2
- prime time, 10-2
- printer, 10-10
- process, 10-21
- records
  - daily, 10-6
- reports
  - daily, 10-6
- samples, 10-39
- service charges, 10-33
- setting up, 8-26, 10-9
- adm file, 10-11
- holidays file, 10-11
- rc.config file, 10-10
- root file, 10-11
- shell scripts
  - ckpacct, 10-24
  - dodisk, 10-31
  - lastlogin, 10-20
  - list of, 10-3
  - prctmp, 10-19
  - prdaily, 10-40
  - prtacct, 10-40
  - runacct, 10-35
  - shutacct, 10-13
  - startup, 10-12
  - turnacct, 10-23
  - turnacct off, 10-24
  - turnacct on, 10-23
  - turnacct switch, 10-24
- starting, 10-12
- stopping, 10-13
- submitting commands to cron, 10-12
- turning off, 10-19
- turning on, 10-19
- using the crontab command, 10-12
- utmp file structure, 10-15

**Accounts option**

- SysMan Menu, 7-7
- acctcms command**, 10-27
- acctcom command**, 10-24
- acctcon1 command**, 10-18
- acctdisk command**, 10-33
- acctdusg command**, 10-32
- acctmerg command**, 10-39
  - correcting tacct file errors, 10-39
  - syntax, 10-39
- accton command**, 10-23
- acctprc1 command**, 10-28
- acctprc2 command**, 10-29
- acctwtmp command**, 10-17
- adapter**, 5-2
- adding an option card**, 5-2

- address space**, 3–47
- adduser utility**, 7–15, 7–28
- administering**
  - account, 7–1, 7–15, A–6
  - AdvFS, A–11
  - applications, A–11
  - ATM, A–8
  - BIND/DNS daemons, A–9
  - CPU resources, 3–24, A–7
  - device special file, 5–1
  - group file, A–6
  - groups, 7–1, 7–28
  - hardware, 1–23, 5–1
  - host equivalence, A–8
  - hosts, 1–23
  - IP routing, A–8
  - lsm, A–11
  - monitoring, A–7
  - network, 1–23
  - network daemons, A–10
  - network gateways, A–9
  - network interface, A–8
  - networking, A–8
  - NFS daemons, A–9
  - NIS, A–10
  - NTP daemon, A–9
  - operating system options, A–11
  - passwd file, A–6
  - passwords, 7–1, 7–15
  - PC account, 7–15
  - PPP and SLIP, A–9
  - printers, A–10
  - privileges, A–10
  - routing, A–8
  - security settings, A–10
  - software PAKs, A–11
  - storage, 1–23, A–11
  - system events, 13–26
  - tuning, A–7
  - ufs, A–11
  - users, 7–1, 7–15
  - using a PC, 1–15
  - Windows NT user account, 7–1
- administering hardware**, 5–1
- administration utilities**, A–1
- administrative methods**, 1–8
- AdvFS**, 6–2
  - dismounting domains, 6–30
  - displaying mounted domains, 6–31
  - domain name, 6–30
  - file system
    - creating with diskconfig, 5–61
  - graphical interface, 6–28
  - replacing failed disk, 5–33
  - SysMan, A–11
- AdvFS utilities**, 6–29
- AlphaServer 1000A**
  - monitoring the environment, 11–14
- AlphaServer 4000**
  - monitoring the environment, 11–14
- Application Manager**, 1–1, 1–9, 1–11
  - class scheduler, 3–26
  - monitoring tools, 11–7
  - shut down, 2–2
  - Storage\_Management folder, 6–27
  - SysMan, 1–15
- application performance**, 11–10
- archiver**, 9–31
  - cpio, pax, and tar commands, 9–31
  - graphical interface, 9–5
- archiving services**, 9–1
- ASU**, 7–9
  - account, 7–9
  - windows utilities, 7–3
- at command**, 3–4
- ATM**
  - SysMan, A–8
- attribute**
  - configurable kernel subsystem, 4–2
  - device, 5–23
  - kernel subsystem, 4–1
  - modifying kernel subsystem, 4–10

**auth.log file**, 14–6  
**authorization file**  
EVM, 13–19  
**autofs**  
NFS mounted file system, 6–31  
**automount**, 6–37  
displaying NFS mounted file  
system, 6–31  
**autosysconfig command**, 4–9

## B

---

### **backup**

AdvFS, 9–2  
applications, 9–2  
archiver, 9–31  
avoiding backup data corruption,  
9–10, 9–11  
cloning, 9–2  
configuration cloning, 9–2  
cpio, pax, and tar commands, 9–30  
dxarchiver utility, 9–31  
full, 9–11  
incremental, 9–13  
log files, 3–15  
LSM, 9–2  
LSM mirrors, 9–2  
media changer, 9–7  
overview, 9–2  
procedure, 9–1  
remote, 9–13  
scheduling, 9–4  
tools and utilities, 9–30  
UFS quota files, 7–11  
UFS sparse files, 7–11  
using scripts, 9–14  
**baud rate**, 8–34  
**baud settings**, 1–32  
**bcheckrc script**, 3–3, 3–7  
**binary configuration file**  
event logging, 12–9  
**binary record**  
accounting, 10–29  
**BIND/DNS**

configuring, 1–14  
**binlog**, 12–4, 13–2  
and EVM, 13–2  
**binlog.conf file**, 12–9, 14–7  
log files, 12–18  
**binlogd daemon**, 12–3, 12–15,  
14–7  
and EVM, 13–10  
disable UDP socket, 12–15  
stopping, 12–16  
**binlogdumpfile file**, 14–7  
**boot**, 2–6, 5–44  
alternate kernel, 2–15  
bootable tape, A–12  
console, 2–10  
flags -i, 2–16  
genvmunix, 2–7  
interactive, 2–16  
overriding set commands, 2–15  
shutdown command, A–13  
standalone, 2–10  
troubleshooting, 2–21  
**boot block**, 6–8  
**boot command**  
options, 2–12  
**boot device**, 2–14  
**boot disk**  
duplicating, 9–28  
**boot drive**  
alternate, 9–27  
**boot log**, 12–18  
**boot preparation**  
after a system crash, 2–9  
from a halted system, 2–8  
powered-down systems, 2–8  
to single-user mode, 2–9  
**boot time errors**, 5–44  
**boot\_osflags variable**  
console, 2–10  
**bootable tape**, 9–5, 9–34  
graphical interface, 6–28  
locking file, 9–34  
SysMan, A–12  
tape requirements, 9–34

- bootdef\_dev**, 2–14
- bounds file**, 14–6
- BSD pty**, 5–54
  - removing, 5–55
- btcreate command**, 9–5, 9–34, A–12
- btextract command**, 9–34
- bttape**, 9–34
- bttape utility**, 9–5
- bttape.pid locking file**, 9–34
- bus**, 5–2

## C

---

### **CalComp DrawingBoard**

- configuring, D–2
- cam\_data.c file**
  - converting, 5–50
- category, device**, 5–22
- CDE**, A–1
  - administration graphical interfaces, 1–11
  - Application Manager, 1–11
  - configuration, A–2
  - configuration utilities, 1–13
  - Daily Administration, A–2
  - front panel
    - SysMan Station, 1–11
    - tool drawer, 1–11
  - Monitoring/Tuning, A–2
- CDFS**, 6–2
- CDSL**, 1–10, 5–39, 6–1, 12–5
  - administering, 6–15
  - and clusters, 6–11
  - as symbolic links, 6–14
  - defined, 6–12
  - device special files, 5–7
  - fixing, 6–16
  - in file system hierarchy, 6–11
  - structure, 6–14
  - verifying, 6–15

- cdslinchk command**, 6–15

### **CDSLs**

- and shared files, 6–13

### **century**

- setting, 2–21

- cfgmgr daemon**, 4–6

- cfgmgr.auth file**, 4–14

- chfn command**, 7–6

- chmod command**, 6–9

- chsl command**, 7–6

### **CI**, C–1

- configuration, C–1

- ckpacct shell script**, 10–24

- class scheduler**, 3–24, 3–25, A–7

- adding class members, 3–32

- CDE, 3–26

- changing priority, 3–32

- class\_admin, 3–25, 3–27

- class\_scheduling, 3–25

- configuring, 3–27

- creating classes, 3–29

- daemon, 3–31

- deleting class members, 3–32

- destroying a class, 3–32

- disabling, 3–31

- enabling, 3–31

- GID, 3–25, 3–30

- graphical user interface, 3–34

- identifier types, 3–30

- invoking, 3–26

- loading databases, 3–32

- managing classes, 3–29

- nice command, 3–26

- PGID, 3–25

- PID, 3–25

- planning, 3–27

- process identifiers, 3–30

- runclass command, 3–25, 3–33

- SESS, 3–25

- SysMan, A–7

- SysMan Menu, 3–26

- UID, 3-25, 3-30
- class\_admin**, 3-24
  - administering, 3-29
  - using, 3-27
- class\_admin command**, A-7
- cloning**, 9-2
- cloning a disk**, 9-20
- cloning information**, 1-8
- cluster**, 5-26
  - CDSL, 6-11
  - member, 6-14
  - shared file, 6-13
  - viewing, 5-13
- cluster root disk**
  - failed or crashed, 5-34
- cmx exerciser**, 11-24
- collect**, 11-8
- COM1\_BAUD**, 1-33
- COM1\_FLOW**, 1-33
- COM1\_MODEM**, 1-33
- comm ports**, 1-31, 1-34
- communications system**, 11-24
- Compaq Analyze**, 13-3, 13-11
  - and binlogd, 13-10
  - error reporting, 12-1
- Compaq Insight Manager**
  - auto-discovery, 1-29
  - master device, 1-29
  - slave devices, 1-29
  - WBEM, 1-28
  - XE, 1-28
  - XE agent, 1-29
- component**, 5-17
- component subsystem**, 5-4
- compressed crash dump**, 14-16
- Computer Interconnect bus**, C-1
- configuration**
  - device names, 5-37
  - documentation, 4-2
  - kernel
    - dynamic, 4-6
    - static, 4-18
  - subsystem, 4-1
- configuration file**
  - adding devices to, 4-20
  - entries, 4-32
  - event logging, 12-5
  - extensions to, 4-30
  - keywords, 4-32
  - NAME.list file, 4-30
  - param.c file, 4-31
- configuration report**, 11-7
- configure system**, 1-14
  - accounts, A-6
  - AdvFS, A-11
  - ATM, A-8
  - BIND/DNS, 1-14, A-9
  - bootable tape, A-12, A-13
  - class scheduler, A-7
  - custom, 1-15
  - DNS, A-9
  - dop, A-10
  - gated, A-8
  - group file, A-6
  - host equivalence, A-8
  - host file, A-8
  - IP router, A-8
  - LAT, A-10
  - lsm, A-11
  - mail, 1-15
  - network, 1-14
  - network gateways, A-9
  - network interface, A-8
  - network interface daemon, A-10
  - network routing, 1-14
  - networking, A-8
  - NFS, 1-15, A-9
  - NIS, 1-15, A-10
  - NTP, 1-14, A-9
  - passwd file, A-6
  - PPP, A-9
  - printers, 1-15, A-10
  - routed, A-8
  - rwhod, A-8
  - security, A-10
  - SLIP, A-9
  - software, A-11



- static routes, A-8
- storage, A-11
- SysMan, A-6
- SysMan Menu, 1-15
- ufs, A-11
- configuring**
  - kernel, 4-1
  - kernel subsystem, 4-1
  - printers, 8-2
  - system
    - using SysMan, 1-12
  - system utilities for, A-2
  - using CDE utilities, 1-13
- connect session**
  - date change, 10-19
  - line usage records, 10-19
  - overall record, 10-19
- connection types, 8-32**
- console**
  - auto\_action reboot, 2-3
  - boot, 2-10
  - boot command, 2-10
  - boot device, 2-14
  - boot environment variables, 2-12
  - boot\_osflags, 2-10
  - boot\_osflags options, 2-13
  - environment variables, 2-12
  - messages, 1-35
  - overriding variables, 2-15
  - port, 1-31
  - setting up, 1-32
  - setting variables, 2-5
  - show devices, 2-13
  - standalone boot, 2-11
- console environment, 1-33**
- console environment variables**
  - defined, 2-12
- console port, 1-34**
- console prompt, 2-27**
- console variables, 2-3**
- consvar command**
  - getting console variables, 2-3
  - setting console variables, 2-5
- context-dependent symbolic link, 6-11**
- continuable dump, 14-19**
- controller, 5-2**
- copying a disk, 9-20**
- cpio command, 9-5, 9-30**
- CPU, 5-2**
  - administering resources, 3-24
  - viewing information about, 5-15
- crash command, 14-20**
  - using at the console prompt, 14-20
- crash directory**
  - changing the location of, 14-12
  - default location, 14-5
- crash dump, 12-17**
  - allocating space for, 14-12
  - changing default location of, 14-12
  - compressed, 14-16
  - compressing and uncompressing, 14-21
  - continuable, 14-19
  - disk space, 14-11
  - dumpsys, 14-19
  - estimating the size of dumps, 14-10
  - file, 14-1
  - forcing on a hung system, 14-20
  - header, 14-8
  - how created, 14-3, 14-7
  - including user page tables in partial dumps, 14-14
  - saving in files, 14-13
  - selecting partial or full, 14-15
  - version number assignment, 14-6
  - vmcore.n, 14-5
  - vmunix.n, 14-5
- crash recovery, 2-9, 12-17**
- crashed device, 5-33**
- creating a group, 7-28**
- creating an account, 7-1**

**cron**  
 cleaning log files, 3–15  
 log file management, 3–15  
 submitting commands to daemon,  
 3–14  
 sys\_check, 11–12  
**cron daemon**, 12–18  
 setting up automatic accounting,  
 10–11  
**crontab command**, 3–14  
**crontab file**, 12–18  
**crontabs directory**  
 modifying files in, 3–14  
**csh shell**, 7–6  
**custom setup**, 1–7

## D

---

**daemon**  
 binlogd, 12–3  
 class scheduler, 3–31  
 evmd, 13–5, 13–7, 13–47  
 gated, A–8  
 lpd, 8–34  
 network services, A–10  
 NTP, A–9  
 routed, A–8  
 rwhod, A–8  
 syslogd, 12–2, 12–15  
**Daily Administration tools folder**,  
 A–2  
**data**  
 recovering, 9–1  
**data block**, 6–9  
**data limit**, 3–47  
**date command**, 2–21  
**DCD**, 1–33  
**dd command**, 5–67  
 cloning on a data disk, 5–67  
**DDR**, 5–49  
 compiling changes to databases,  
 5–51  
 conforming to standards, 5–50

converting cam\_data.c file, 5–52  
 database, 5–51  
 ddr.dbase file, 5–51  
 help option, 5–51  
 SCSI-2 Standard, 5–50  
 synchronizing databases, 5–51  
**ddr\_config command**, 5–49  
 help option, 5–51  
 TagQueueDepth parameter, 5–51  
**DECEvent**, 13–3, 13–11  
 error reporting, 12–1  
**deferred mode swapping**, 3–45  
**/dev directory**, 5–37  
**device**, 5–4, 5–26  
 adding to the kernel, 4–20  
 administering specific, D–1  
 category, 5–45  
 class, 5–45  
 configuring, 5–37  
 database, 5–49  
 deleting, 5–29  
 shared, 5–8  
 testing, 5–9  
 utilities  
   finding, 5–57  
   related, 5–9  
 viewing, 5–13  
 WWID, 5–8  
**device definition keyword**  
 configuring into the kernel, 4–20  
**device mnemonics**, B–1  
 configuration file syntax, B–1  
 in configuration data base, B–1  
**device name**, 5–30, 5–37  
 converting, 5–42  
 disk, 5–42  
 tape, 5–42  
**device pathname**  
 explanation of, 8–31  
 representation in printcap, 8–31  
**device special file**, 5–39, 5–44,  
 6–10  
 converting, 5–42

- creating manually, 5–49
- databases, fixing, 5–46
- deleting, 5–48
- device categories, 5–45
- device classes, 5–45
- displaying name, 6–31
- dn\_setup, 5–44
- dsfmgr, 5–1
- errors, 5–44
- file system, 6–30
- finding, 5–14
- managing, 5–43
- printer, 8–31
- reassigning, 5–49
- representation in printcap, 8–31
- rz (disk) devices, 5–40
- tz (tape) devices, 5–40
- device special file errors**, 5–44
- device utilities**, 5–1
- devices**
  - adding or removing, 5–44
- devswmgr**, 5–60
- df command**
  - checking free disk space, 5–69, 6–46
- dia command**
  - binary event-logging reports, 12–1
- directory**, 6–10
  - backing up, 9–1
  - hierarchy, 6–3
  - link, 6–9
  - recovering, 9–1
  - type, 6–3
- disaster**
  - recovery, 9–1
- disaster planning**
  - bootable tape, A–12
- disaster recovery**, 9–22
- disk**
  - adding automatically, 5–4
  - adding manually, 5–55
  - adding static, 5–55
  - alias name, 5–61
  - attributes, 5–23
  - backup, 9–1
  - basename, 5–62
  - category, 5–22
  - checking usage, 5–70
  - cloning, 5–67
  - copying, 5–67
  - deleting, 5–29
  - device name, 5–37
  - device special file, 5–49
  - device special files, 5–41
  - duplicating and copying, 9–20
  - examples of management, 5–21
  - failed or crashed, 5–33
  - finding, 5–21
  - finding location, 5–14
  - identifier, 5–24
  - label, 5–63, 6–4, 6–5, 6–18
  - monitoring, 6–46
    - df command, 5–69
    - space, 5–69
  - name, 5–30, 5–35
    - deleting or removing, 5–36
  - partition table, 6–19t
  - partitioning, 5–61
  - persistence, 5–35
    - deleting or removing, 5–36
  - power management, 3–37
  - recovering, 9–1
  - rz (SCSI), 5–3
  - shared, 5–31
  - swapping, 5–33
  - viewing, 5–26
    - properties, 5–16
    - worldwide identifier, 5–31
- disk configuration utility**, 5–39
- disk drive**
  - testing with diskx, 11–22
- disk partition**, 6–5
  - changing parameters, 5–66

- changing size, 5–65
- defined, 5–63, 6–4
- file system, 6–3
- overlapping partitions, 5–66
- sizes, 5–63, 6–4
- writing the default label, 5–65
- disk quota**, 6–41
  - creating UFS, 6–1
  - recovering from over-quota condition, 6–42
- disk space**
  - checking blocks used, 5–71, 6–48
  - checking free space, 5–69, 6–46, 6–47
  - checking usage, 6–47
  - reallocating, 5–63
- disk usage**, 5–68
- diskconfig**, 5–39, 5–61, 6–6
  - creating file system, 6–32
- disklabel**, 5–38, 5–61, 6–6
- disklabel command**
  - changing disk partition size, 5–65
  - labeling a disk, 9–15
  - using in recovery, 9–26
  - using the -e option, 5–66
  - writing a default partition table, 9–15
  - writing the default label, 5–65
  - zeroing label, 5–68
- disktab**, 6–4
- diskusg command**, 10–31
- diskx exerciser**, 11–22
- dismounting a file system**, 6–30
- dn\_setup utility**, 5–44
  - install tape with, 9–26
- DNS/BIND**
  - SysMan, A–9
- doconfig program**, 4–4, 4–20, 4–24, 4–26
- dodisk shell script**, 10–31
- domain name server**, 1–14
- DOP**
  - SysMan, A–10
  - sysman -cli, 1–18
- driver, loading**, 5–49
- dsfmgr**, 5–43
  - listing device special files, 6–33
- dsfmgr command**, 5–1, 5–37
  - using in recovery, 9–27
- DTR**, 1–33
- du command**
  - reporting blocks used, 5–70, 6–47
- dump**, 14–1
  - to exempt memory, 14–17
  - to remote host, 14–18
- dump command**, 9–5
- dump\_sp\_threshold system attribute**, 14–8
- dump\_user\_pte\_page system attribute**, 14–14
- dumpfs command**
  - checking free disk space, 5–69, 6–47
- dumpsys**
  - estimating the size of dumps, 14–10
- dumpsys command**, 14–19
- duplicate UID**, 7–26
- duplicating a root disk**, 9–20
- dxaccounts utility**
  - defaults, 7–13
  - system setup, 7–6
- dxarchiver**, 9–31
- dxfileshare**
  - mounting (importing) shared file system, 6–37
  - sharing file system, 6–35
- dxkerneltuner**
  - monitoring, 11–7
- dxpower**, 3–37
- dxproctuner**, 3–26
- dxshutdown**, A–13
- dxshutdown utility**, 2–24
- dxsysinfo**
  - monitoring swap space, 3–47
- dynamic configuration**, 4–6
- dynamic device recognition**, 5–49
- dynamic subsystem**

- configuring into the kernel, 4–7
- determining the state of, 4–7
- list of, 4–5
- unloading, 4–9

## E

---

### **e-mail**

- configuring, 1–15

### **ECU**

- running, 1–32

### **ed editor, 2–11**

### **edquota editor**

- activating, 6–43
- setting grace period, 6–43

### **energy conservation, 3–37**

### **energy star, 3–37**

### **enhanced security**

- user account changes, 7–3

### **envconfig utility, 11–14**

- checking thermal levels, 11–17
- displaying option values, 11–18
- setting threshold levels, 11–18
- stopping and starting envmond daemon, 11–17
- turning envmond daemon on or off, 11–17

### **environment configuration**

- utility, 1–32**

### **environmental monitoring**

- checking thermal levels, 11–17
- components of, 11–14
- configuring the envmond daemon, 11–14
- displaying option values, 11–18
- model of, 11–14
- setting threshold levels, 11–18
- shutdown, 2–23
- stopping or starting, 11–17
- turning on or off, 11–17
- using the configuration utility, 11–14

- using the envmond daemon, 11–14
- using the get\_info function, 11–15
- using the kernel module component, 11–14
- using the Server System MIB daemon, 11–14

### **envmond daemon, 11–14**

- broadcasting a message, 11–17
- checking thermal levels, 11–17
- customizing, 11–17
- displaying option values, 11–18
- enabling during system boot, 11–17
- initiating system shutdown, 11–17
- querying system thresholds, 11–17
- reading rc.config file, 11–17
- setting threshold levels, 11–18
- stopping and starting, 11–17
- turning on or off, 11–17

### **error, 13–1**

- cautionary system shutdown, 2–22
- event, 13–1
- found by sys\_check, 11–11
- logging, 12–1

### **error log**

- explanation of file, 8–35
- printer, 8–35
- representation in printcap, 8–36

### **escalation report, 11–7**

### **/etc/disktab file, 5–61**

### **/etc/evmlogger.conf, 13–47**

### **/etc/exports**

- sharing file system, 6–35

### **/etc/fstab, 5–37**

### **/etc/group file, 7–9, 7–14, A–6**

### **/etc/hosts, A–8**

### **/etc/passwd, 7–11**

### **/etc/passwd file, 7–10, A–6**

### **/etc/rc.config file, 3–3**

### **/etc/securettys file, 5–55**

- /etc/sysconfigtab file, 1–33, 4–14**
  - multiple versions, 4–14

- event**
  - client, 13–3, 13–22
  - defined, 13–1
  - log
    - log file protections, 12–2
    - syslogd daemon, 12–2
    - system event-logging facility, 12–2
  - model of, 13–3
  - poster, 13–3
  - reporting, 12–1
  - suppression, 13–47
- event logging**
  - binary configuration file, 12–9
  - binary event-logging facility, 12–3
  - binlog.conf file, 12–9
  - binlog\_data.c file, 12–16
  - binlogd daemon, 12–3
  - configuration file, 12–5
  - configuring binary event logger, 12–16
  - crash recovery, 12–17
  - creating daily files, 12–8
  - creating special files, 12–13
  - default configuration, 12–4
  - maintaining files, 12–18
  - setting up, 12–4
  - starting, 12–14
  - syslog.conf file, 12–6
  - syslogd daemon
    - stopping, 12–15
  - using the dump file, 12–17
- event management**, 13–1, A–7
- Event Manager**, 13–1
- event-logging daemon**
  - command syntax, 12–14
- EVM**, 13–1
  - administering, 13–11
  - API, 13–3, 13–8
  - archived (zipped) logs, 13–21
  - authorization file, 13–19
  - channel manager, 13–6, 13–20
  - command line, 13–7
  - components, 13–5
    - configuring, 13–11
    - configuring event logs, 13–47
    - daemon, 13–3
    - displaying events, 13–47
    - event forwarding, 13–3, 13–6
    - event logger, 13–3
    - event logging, 13–47
    - event suppression, 13–47
    - event template, 13–21
    - event viewer, 13–3
    - evmchmgr
      - configuration, 13–13
    - evmchmgr command, 13–20
    - evmd configuration, 13–12
    - evmget, 13–6, 13–21
    - evmlog, 13–21
    - evmlog\_cleanup, 13–21
    - evmlogger, 13–6
    - evmlogger configuration, 13–14
    - evmreload, 13–12
    - evmshow, 13–4
    - evmsort, 13–4
    - evmtemplate file, 13–22
    - evmviewer, 13–41
    - evmwatch, 13–3, 13–6
    - features, 13–2
    - forwarding events, 13–47
    - get server, 13–6
    - hardware management, 5–4
    - installing clients, 13–22
    - log file cleanup, 13–21
    - log file management, 13–3, 13–20, 13–21
    - logger, 13–3, 13–6
    - message log file management, 13–21
    - misclog, 13–21
    - posting client, 13–5
    - processing events automatically, 13–47
    - remote access, 13–20
    - responding to events, 13–46
    - reviewing logged events, 13–41

- security, 13–18
- shut down, 2–4
- startup, 13–5
- subscribing client, 13–5
- SysMan Station, 13–21
- system files, 13–8
- troubleshooting, 13–49
- user authentication, 13–18
- using in administration, 13–26
- utilities, 13–4
- evm daemon**, 13–47
- EVM viewer**
  - SysMan, A–7
- evm.auth**, 13–19
- evm.buf**, 14–7
- evmchmgr command**, 13–7, 13–20
- evmd daemon**, 13–5, 13–7
- evmget command**, 13–6, 13–7
- evmget\_srv process**, 13–6
- evmlog\_cleanup function**, 13–21
- evmlogger command**, 13–7, 13–47
- evmpost command**, 13–7
- evmreload command**, 13–8, 13–12
- evmsort command**, 13–4, 13–7
- evmsstart command**, 13–8
- evmstop command**, 13–8
- evmtemplate file**, 13–22
- evmviewer utility**, 13–41
- evmwatch command**, 13–3, 13–6, 13–7
- exempt memory**, 14–17
- exercisers**, 11–19
- expected\_dump\_compression variable**, 14–15
- exporting file system**, 6–35
- extendfs command**, 6–24
- extending UFS file systems**, 6–24

## F

---

- failed device**, 5–33
- fan failure**, 11–14
- fastboot command**, 2–30
- fasthalt command**, 2–30
- fault**, 13–1
- fdi controller**, 5–2
- FFM**, 6–2
- file**, 6–3
  - backing up, 9–1
  - evmlogger.conf, 13–47
  - protection (mode), 6–9
  - recovering, 9–1
- file system**, 6–1
  - administering with SysMan, 6–27
  - AdvFS, 6–2
  - AdvFS domains, displaying, 6–31
  - automount, 6–37
  - boot block, 6–8
  - bootable tape, 6–28
  - CDFS, 6–2
  - checking, 6–45
  - corrupted, 6–45
  - corruption shutdown, 2–23
  - creating quotas, 6–1
  - creating UFS using SysMan, 6–40
  - creating with diskconfig, 5–61
  - creating with newfs, 6–17
  - current mount points, displaying, 6–31
  - data block, 6–9
  - device special file name, 6–30
  - directory, 6–10
  - directory hierarchy, 6–4
  - disabling access, 6–23
  - disk partitions, 6–3
  - dismounting, 6–30
  - dismounting currently mounted, 6–32
  - displaying mounted using SysMan, 6–30

- displaying NFS mounted file system, 6-31
- displaying setup, 5-64
- dsfmgr, 6-33
- enabling access, 6-20
- exercising with fsx, 11-20
- exporting (sharing), 6-35
- FFM, 6-2
- file system full message, 6-24
- file types, 6-3, 6-10
- file-on-file mounting, 6-2
- file-on-file, displaying, 6-31
- free space, 5-68
- fstab, 6-31
- fstab file, 6-20
- increasing capacity of, 6-24
- inode, 6-8
- inode blocks, 6-8
- limiting usage, 6-41
- link count, 6-9
- LSM, 6-2, 6-28
- managing directories, 6-10
- managing files, 6-10
- MFS, 6-2
- monitoring, 6-46
- monitoring and tuning, 6-1
- monitoring free space, 5-69
- monitoring usage, 5-70
- mount NFS using SysMan, 6-37
- mounting, 6-20, 6-23
- mounting excluded file system, 6-34
- mounting single, using SysMan, 6-33
- mounting specific type, using SysMan, 6-33
- mounting standalone, 2-11
- mounting using dxfilesare, 6-37
- mounting using SysMan, 6-33
- pipes, 6-3
- protection (mode), 6-9
- quotas for groups, 6-41
- quotas for user accounts, 6-41
- refreshing currently mounted listing, 6-32
- repairing interactively, 6-46
- sharing using dxfilesare, 6-35
- socket, 6-3
- structure, 6-3
- superblock, 6-8
- supported block size, 6-20
- symbolic link, 6-3
- SysMan utilities, 6-29
- troubleshooting, 6-1, 6-49
- tuning, 6-49
- UFS, 6-1
- unmounting, 6-23
- unsharing, 6-36
- using diskconfig to create, 6-32
- using hwmgr to check disk, 6-32
- verifying disk partition, 6-32
- verifying mount point, 6-32

**file system quota, 6-41**

- activating, 6-43
- activating edquota editor, 6-43
- disk quotas, 6-41
- for groups, 6-41
- for user accounts, 6-41
- setting automatic, 6-44
- setting grace period, 6-43
- turning off, 6-43
- verifying, 6-44

**file system utilities**

- CDE Storage\_Management, 6-27

**file systems**

- SysMan, A-11

**file types, 6-10**

- device, 6-10
- domain socket, 6-10
- named pipes, 6-10
- symbolic link files, 6-10

**file-on-file file system**

- displaying mounted, 6-31

**finger command, 7-6**

**firmware**

- setting variables, 2-5



**firmware sources**, 9–21

**floppy**

interface, 5–2

**fsck command**, 6–46

checking file system, 6–45

correcting file system, 6–45

overlapping partitions, 6–46

shut down, 2–5

syntax, 6–46

**fsck program**, 6–20

**fstab file**, 6–20, 6–23

displaying current mounted file  
system, 6–31

editing, 6–21

**fsx exerciser**, 11–20

**full crash dump**, 14–12

**fwtmp command**, 10–16

correcting wtmp file, 10–16

## G

---

**gated daemon**

SysMan, A–8

**gateway**

network gateways, A–9

**gecos data**, 7–12

**generic kernel**, 4–4

**genvmunix**, 2–5, 2–7, 2–15

**get server**, 13–6

**get\_info function**, 11–15

**getty command**, 2–18, 2–20, 3–7

**getty setting**, 1–33

**gettydefs file**, 3–4

**GID**, 7–10

in group file, 7–14

in passwd file, 7–11

limits, 7–15

maximum number, 7–10

**Graphical User Interfaces**, 1–9

**group**, 7–1, 7–9, 7–14, A–6

adding, 7–28

administering, 7–28

checking file, 7–6

copying with dxaccounts, 7–26

defaults, 7–28

deleting, 7–28

group membership, 7–28

groupadd command, 7–8

groupdel command, 7–8

groupmod command, 7–8

modifying, 7–28

password, 7–28

PC, 7–28

user name, 7–14

**group file**, 7–9

adding a group to the, 7–14

line length limits, 7–15

LINE\_MAX, 7–15

/usr/include/limits.h, 7–15

**group identifier**, 7–10

**group membership**, 7–1, 7–15,  
7–28

**group\_id**, 7–14

**groupadd command**, 7–8

**groupdel command**, 7–8

**groupmod command**, 7–8

**groupname**, 7–14

**grpck**, 7–6

**GUI**, A–1

## H

---

**halt command**, 2–27, 2–29

**halting the system**, 2–23, 2–27,  
2–30

**hardware**

adding to the kernel, 4–20

administering specific devices, D–1

**hardware administration**

SysMan, A–6

**hardware component**, 5–4

**hardware management**, 5–1

- adapter, 5-2
- adding an option card, 5-2
- adding components automatically, 5-4
- adding devices manually, 5-55
- adding hardware, 5-1
- attribute, 5-23
- bus slot, 5-3
- buses, 5-2
- category, 5-22
- CDE Application Manager, 5-10
- cluster, 5-26
- component subsystem, 5-17
- configuring, 5-19
- controller, 5-2
- copying disk, 5-67
- CPU, 5-2
- deleting devices, 5-29
- device name, 5-35
  - deleting, 5-36
  - removing, 5-36
- device overview, 5-3
- device persistence, 5-35
- device special file, 5-1
- device utilities, 5-1, 5-56
- devices
  - failed, 5-33
  - failed cluster root, 5-34
  - identifier, 5-31
  - naming, 5-30
  - shared, 5-31
  - viewing, 5-26
- devswmgr, 5-60
- disk device special files, 5-41
- disk space, 5-69
- disk usage, 5-70
- diskconfig, 5-61
- documentation, 5-6
- edit scsi, 5-31
- environment, 5-19
- finding hardware, 5-21
- hardware identifier (HWID), 5-24
- HSZ and HSG devices, 5-3, 5-33
- HWID, 5-24
- hwmgr command, 5-1, 5-16
- hwmgr command options, 5-18
- launching tasks from SysMan Station, 5-16
- locating hardware, 5-21
- MAKEDEV, 5-55
- manual methods, 5-49
- model, 5-17
- monitoring disk space, 5-69
- monitoring disk usage, 5-70
- name subsystem, 5-17, 5-35
- naming devices, 5-30
- overview, 5-2
- partitioning disk, 5-61
- persistence
  - deleting, 5-36
- pseudodevices, 5-1
- RAID devices, 5-33
- related utilities, 5-9
- rz (SCSI) devices, 5-3
- SCSI device, 5-3
- SCSI subsystem, 5-17
- scu, 5-59
- sharing devices, 5-31
- subsystems, 5-17
- sysconfig, 5-9
- SysMan, 5-10
- SysMan Menu tasks, 5-10
- SysMan Station, 5-10, 5-15
- system device hierarchy, 5-4
- system files, 5-7
- system topology, 5-4
- tape device special files, 5-42
- testing device, 5-9
- testing SCSI CAM devices, 5-9
- transactions, 5-28
- usage examples, 5-21
- using SysMan, 5-1
- viewing
  - cluster, 5-13, 5-26
  - CPU data, 5-15
  - device attributes, 5-23

- device categories, 5–22
- device properties, 5–16
- devices, 5–13, 5–26
- hierarchy, 5–21
- system hierarchy, 5–11
- worldwide identifier (WWID), 5–31
- hardware-specific parameters**, 11–16
- Hierarchical Storage Controllers**, C–1
  - ( *See also* HSC )
- hierarchy**, 5–21
- host equivalence**
  - SysMan, A–8
- host file**
  - SysMan, A–8
- HSC**
  - configuration, C–1
  - controller failures, C–2
  - host sharing, C–3
  - restrictions, C–3
- HSG**, 5–3
  - failed or crashed disk, 5–33
  - hardware management, 5–33
- HSZ**, 2–14, 5–3, 5–40
  - failed or crashed disk, 5–33
  - hardware management, 5–33
- hung system**
  - forcing a crash dump on, 14–20
- HWID**, 5–24
  - uniqueness, 5–8
- hwmgr**
  - checking disk availability, 6–32
  - viewing devices, 6–40
- hwmgr command**, A–6
  - using, 5–18

---

**I**

- i18n**, 3–17
- ikdebug debugger**, 1–34
- immediate mode swapping**, 3–45

- importing file system**, 6–37
- init command**, 2–7, 2–17, 2–23
  - changing run level, 2–18
  - multiuser run level, 2–19
  - reexamining the inittab file, 2–20
- init.d directory**, 3–3, 3–9
- init.d directory structure**, 3–9
- initialization**
  - script, 2–7
  - tasks, 2–6
- inittab file**, 2–7, 2–18, 3–2, 3–4, 3–8
  - activating terminal lines, 2–20
  - boot entry, 2–18
  - bootwait entry, 2–18
  - changing run level, 2–18
  - initdefault entry, 2–18
  - rc scripts, 2–19
- inittab**, 2–4
- inode**, 6–8
  - blocks, 6–8
  - free, 6–8
  - number, 6–9
- Insight Manager**, 11–7
  - hardware management, 5–5
  - port, 1–29
- installing applications**, A–11
  - setld, A–11
- interactive boot**, 2–16
- international**
  - printing, 8–41
- internationalization**, 3–16, 3–21
  - character sets, 3–17
  - code\_page, 3–17
  - configuration, 3–17
  - i18n\_intro, 3–17
  - i18n\_printing, 3–17
  - iconv, 3–17
  - setting locale, 3–17
- internet protocol**, A–8
  - configuring using SysMan, A–8

**iostat**, A-7  
SysMan, A-8  
**iostat command**, 3-26, 11-5

## K

---

### kernel

attributes  
environmental monitoring,  
11-15  
boot alternate, 2-7, 2-14  
configuration file entries, 4-32  
configuration manager  
support of the kernel module,  
11-14  
configuring, 4-1  
using options, 4-24  
continuable dump, 14-19  
debugging, 14-19  
debugging remote, 1-34  
drivers, 5-49  
dynamic configuration, 4-6  
dynamic reconfiguration, 5-49  
module, 11-14  
loading and unloading, 11-15  
supported parameters, 11-14  
postinstallation configuration, 4-4  
static configuration, 4-18  
static reconfiguration, 5-49  
subsystem  
setting configuration variables,  
3-3  
using `ddr_config`, 5-49  
`vmunix.n` file, 14-3  
**kernel set management**, 5-17  
**killall command**, 2-23  
**kmknod command**, 5-49  
**ksh shell**, 7-6  
**KZPBA-CB buses**, 2-14

## L

---

label

zeroing, 5-68  
**large log files**  
cron, 3-15  
**last command**, 10-20  
**lastcomm command**, 10-30  
**lastlogin shell script**, 10-20  
**LAT**  
latsetup, A-10  
SysMan, A-10  
**LBN**, 6-4  
disk label, 5-63  
partitioning disks, 5-63  
**lf print parameter**  
error log, 8-35  
**license**, 1-14  
**limit**, 3-47  
**line printer daemon**, 8-34  
**lineuse file**, 10-18  
**link**, 6-9  
CDSL, 6-14  
**lmfsetup command**, A-11  
using SysMan, A-11  
**local-language**  
printing, 8-41  
**locale**, 3-16, 3-17  
**lock**  
account administration, 7-4  
**log facility**  
displaying events, 13-47  
event forwarding, 13-47  
event suppression, 13-47  
EVM, 13-46  
storing events, 13-47  
**log files**  
account, 7-10  
backup by cron, 3-15  
cleaning by cron, 3-15  
maintaining, 12-18  
management in EVM, 13-3  
`/var/adm/messages`, 11-7  
**logged events, reviewing**, 13-41  
**logger utility**, 13-6  
**logging**, 14-6  
**logical block number**, 5-63, 6-4

**logical partition**  
configuring, D-2

**Logical Storage Manager**, 6-28

**login**  
directory, 7-12  
disabled, 2-26  
shell, 7-13

**lpc command**  
arguments, 8-24

**lpd daemon**, 8-34  
error log, 8-35  
filter representation in printcap,  
8-27, 8-36

**lpd filter**  
explanation of, 8-36

**lprsetup utility**, 8-1, 8-18  
choosing options in, 8-19  
example, 8-20  
main menu, 8-19  
modifying printer configuration,  
8-19  
running, 8-12, 8-19

**lptest command**, 8-42

**lsm**  
SysMan, A-11

**LSM**, 6-28  
creating a UFS file system, 6-40  
replacing failed disk, 5-33  
utilities, 6-28

## M

---

**M\_EXEMPT memory type**, 14-17

**magnetic tape drive**  
adding static, 5-55  
testing with tapex, 11-23

**mail**  
configuring, 1-15, A-7

**mailsetup**, A-7

**MAKEDEV command**, 5-49

**max-per-proc-address-space**, 3-47

**max-per-proc-data-size**, 3-47

**max-per-proc-stack-size**, 3-47

**media changer**, 9-7  
WWID, 5-31

**member**, 6-14

**memory**  
dumping to, 14-17  
exercising with memx, 11-20  
testing shared memory, 11-21

**memory file system**, 6-2

**memx exerciser**, 11-20  
swap space restrictions of, 11-21

**message**  
catalogs, NLS, 3-16  
receiving from system, 11-17

**MFS**, 6-17

**mirroring**, 9-2

**misclog**, 13-21

**mkdir command**  
creating a mount point, 6-32

**mkfdm command**, 5-61

**mknod command**, 5-49

**modem**  
connecting, 1-32  
connections  
closing, 1-35  
troubleshooting, 1-35  
setting up, 1-32, 1-34  
settings, 1-33  
timer settings, 1-33

**monitoring**, 11-1  
Application Manager tools, 11-7  
collect utility, 11-8  
commands and utilities, 11-3  
configuration report, 11-7  
devices, using SysMan Station,  
5-15  
dxsysinfo, 11-8  
escalation report, 11-7  
EVM, A-7  
file system, 6-1  
insight manager, 11-7  
iostat, A-8

- iostat command, 11–5, A–7
- iostat interface, 11–7
- MPH, 11–13
- netstat command, 11–5
- netstat interface, 11–7
- SysMan Station, 1–21
- System Information interface, 11–8
- uptime command, 11–5
- using sys\_check, 11–11
- /var/adm/messages log, 11–7
- vmstat command, 11–4
- vmstat interface, 11–8
- who command, 11–5
- monitoring and tuning**
  - SysMan, A–7
- monitoring the system**, 11–1
- Monitoring/Tuning tools folder**, A–2
- mount**
  - changing status, 6–20
  - command, 6–20
  - command, using, 6–23
  - displaying mounted file system, 6–30
  - file system, 6–32
  - overlapping partitions, 6–23
  - SysMan graphical interface, 6–33
- mount command**, 6–20
  - standalone boot, 2–11
- mount point**, 6–30
  - checking, 6–32
  - displaying current used, 6–31
  - temporary, 6–31
- MPH utility**, 11–13
- msgbuf.savecore file**, 14–6
- multibus failover**, 2–14
- multiuser boot**, 2–6

## N

---

- name server**, 1–14
- name subsystem**, 5–17
- National Language Support**, 3–16

- netman command**, A–8
- netstat command**, 11–5
- network**
  - administration using SysMan, A–8
  - configuring, 1–14
  - configuring gateways using SysMan, A–9
  - interface daemon, A–10
  - routing, configuring, 1–14
  - services, configuring, A–9
  - using SysMan to configure, A–8
- network device**, 9–23
- Network File System**, 1–15
- Network Information Service**, 1–15
- Network Time Protocol**, 1–14
- newfs**, 5–61, 6–32
  - command, using, 6–16
  - creating a file system, 6–17
  - options used in SysMan, 6–40
- NFS**, 1–15
  - autofs utility, 6–31
  - automount, 6–37
  - configuring, 1–15
  - displaying mounted file system, 6–31
  - mount using SysMan, 6–37
  - mounting (importing) file system, 6–37
  - SysMan, A–9
  - SysMan utilities, 6–37
- nice command**, 3–26
- NIS**, A–6
  - configuring, 1–15
  - SysMan, A–10
- NLS**
  - character tables, 3–16
  - environment tables, 3–16
  - libraries, 3–16
  - local directories, 3–16
  - locale, 3–16
  - locale categories, 3–19
  - LOCPATH variable, 3–20

message catalogs, 3–16, 3–20  
NLSPATH variable, 3–20  
setlocale, 3–16  
setting locale, 3–18

## **NTP**

configuring, 1–14  
SysMan, A–9

## **O**

---

### **online help**

SysMan Menu, 1–5  
SysMan Station, 1–5  
web/java mode, 1–5

### **osf\_boot command**, 2–16

### **out of storage space**, 6–24

### **over-commitment mode swapping**, 3–45

### **overlapping disk partitions**

checking for, 5–68

## **P**

---

### **pac command**, 10–34

### **pagination**, 8–40

### **paging**

allocating disk space for, 3–41  
description, 3–42

### **PAK**

registering, 1–14, A–11

### **panic string**

for hardware restart, 14–20  
for hung system, 14–20

### **param.c file**, 4–31

### **partial crash dump**, 14–11

### **partial\_dump variable**, 14–15

### **partition**, 5–61, 5–63, 6–4, 6–5, 6–19t

### **Passed (startup message)**, 5–44

### **passwd**, 7–6, A–6

checking file, 7–6  
dxaccounts, 7–6

file, 7–9, 7–11  
hashed database, 7–13  
performance, 7–13  
usermod, 7–6

### **passwd command**, 7–6

### **passwd file**

NIS distributed, 7–13

### **password**, 7–1, 7–14

administering, 7–15  
group, 7–28  
setting, A–6

### **pax**, 9–30

### **pax command**, 9–5

### **PC**

administering from, 1–15  
group, 7–28  
user account, 7–15

### **PC account**, 7–1

ASU, 7–9

### **PCF**

control file, 8–36  
general purpose filter, 8–36  
PostScript printing, 8–36

### **pcfof**, 8–41

### **pcfof print filter**, 8–36

### **PCL**, 8–36

### **PCMCIA card**

configuring, D–1

### **per-proc-address-space**, 3–47

### **per-proc-data-size**, 3–47

### **per-proc-stack-size attribute**, 3–47

### **per-process memory limits**, 3–46

### **performance**

application, 11–10  
degradation shutdown, 2–23

### **performance monitoring**, 11–10

### **performance monitors**

real-time performance monitor,  
11–14

### **peripheral**, 5–23

### **persistence**, 5–35

- pipe**, 6–3
- PostScript printing**, 8–36
- power off**, 2–27
- PPP**
  - SysMan, A–9
- prctmp shell script**, 10–19
- prdaily shell script**, 10–40
- primary swap partition**, 14–7
- print filter**, 8–41
  - general purpose, 8–36
- print services**, 8–1
- printcap file**, 8–24, 8–29
  - printer characteristics database, 8–28
  - understanding entries in, 8–20
- printconfig**
  - choosing options in, 8–12
  - modifying printer configuration, 8–12
  - using, 8–7
- printer**
  - accounting, 8–26, 8–27, 10–34
    - pac command, 10–34
  - adding, 8–22
  - adding comments to the /etc/printcap file, 8–21
  - ASU, 8–18
  - baud rate, 8–34
  - connecting, 8–32
  - control utility, 8–24
  - controlling jobs and queues, 8–24
  - data files, 8–33
  - device special file name, 8–31
  - error log file, 8–35
  - /etc/printcap file, 8–29
  - flag bits, 8–37
  - installing local, 8–11
  - installing remote, 8–13
  - internationalization settings, 3–17
  - jobs, 8–24
  - line printer daemon, 8–34
  - local, 8–2
  - lock file, 8–32
  - lpc command, 8–24
  - lprsetup utility, 8–18
  - maintaining, 8–22
  - name, 8–29
  - network, 8–2
  - pac command, 10–34
  - pagination, 8–40
  - personal computer, 8–18
  - printconfig, 8–7
  - reference data, 8–27
  - reference names, 8–28
  - related documentation, 8–4
  - remote, 8–2
    - client, 8–40
    - printcap symbols, 8–28
    - server, 8–40
  - removing printers, 8–23
  - reporting usage, 10–34
  - setting up manually, 8–23
  - spooling directory, 8–32
  - spooling queue, 8–24
  - starting lpd daemon, 8–24
  - status, 8–24
  - status file, 8–32
  - synonyms, 8–30
  - system files, 8–5
  - TCP/IP printing, 8–14, 8–15
  - testing, 8–42
  - troubleshooting, 8–42
  - type, 8–30
  - using lprsetup utility, 8–19
  - using printconfig, 8–12
  - utilities, 8–6
- printer control file**, 8–36
- printers**
  - configuring, 1–15
  - SysMan, A–10
- privileges**, A–10
- problem solving**
  - sys\_check, 11–11
- proc subsystem**, 3–47
- process**
  - initializing, 2–6



- virtual memory, 3–46
- process tuner**, 3–26
- Product Authorization Key**, 1–14
- protection**
  - files, 6–9
- prtacct shell script**, 10–40
- pseudodevices**, 5–1
- pseudoterminal**, 5–49, 5–53
  - adding, 5–53
  - BSD STREAMS-based, 5–53
  - clist-based, 5–53
  - creating device special file, 5–54
- psrinfo command**, 5–15
- pty driver**, 5–37, 5–49
  - adding, 5–53
  - creating, 5–54
  - securettys, 5–55
- pwck command**, 7–6

## Q

---

- QIC tape**, 5–40
- quick setup**, 1–6
- quot command**
  - checking blocks used, 5–71, 6–48
- quota**, 6–44
  - UFS, 6–41
  - user and group, 6–41
- quota command**, 7–6
  - verifying block usage, 6–44
- quota limits**
  - reaching, 6–42
- quotacheck command**
  - verifying block usage, 6–44
  - verifying file system quotas, 6–44
- quotaoff command**
  - turning file system quotas off, 6–43
- quotaon command**, 7–6
  - activating file system quotas, 6–43

## R

---

- radisk utility**, 5–59n
- RAID**, 5–3
  - failed or crashed disk, 5–33
  - hardware management, 5–33
- rc directory structure**, 3–9
- rc.config**, 6–13
- rc.config file**, 2–11
  - use by the envmond daemon, 11–17
- rc.config.common**, 6–13
- rc.config.common file**, 2–11
- rc0**, 2–5
- rc0 script**, 3–3
- rc0.d directory**, 3–10
- rc1**, 2–5
- rc2 script**, 3–3
- rc2.d directory**, 3–11
- rc3**, 2–5
- rc3 script**, 3–3
- rc3.d directory**, 3–12
- rcmgr command**, 2–11, 3–3
- rcn.d directory**, 3–3
- rdump command**, 9–5
- reboot**, 2–27, 14–3
  - automatic, 2–3
  - performing abrupt reboot, 2–30
- reboot command**, 2–29
- record**
  - binary accounting, 10–29
  - daily accounting, 10–29
  - overall connect session, 10–19
- recovering a root disk**, 9–20
- recovery**
  - bootable tape, A–12
- remote connection**, 1–32
- remote event monitoring**, 13–20
- remote host**
  - denying shared files system access, 6–36
  - dumping to, 14–18

- enabling shared file system access, 6–36
- remote system administration**, 1–31, 1–34
- remote systems**, 1–31, 1–34
- removing**
  - account, 7–1
  - group, 7–28
  - user, 7–15
- restore**
  - command
    - retrieving files, 9–16
    - retrieving files interactively, 9–17
    - retrieving remote files, 9–19
  - procedures, 9–1
  - retrieving a file system, 9–15
  - retrieving data, 9–14
- restore command**, 9–5
  - using in recovery, 9–28
- restoring**
  - /usr file system, 9–29
  - /var file system, 9–29
- RIS**, 9–22
- root disk**
  - duplicating, 9–28
  - recovering, 9–20
- root file system**
  - mounting read-write from
    - single-user mode, 2–7, 6–46
  - verifying, 2–16
- root login**
  - enabling on terminal, 3–9
- routed daemon**
  - SysMan, A–8
- routing**
  - gated, A–8
  - routed, A–8
  - static routes, A–8
- rrestore command**, 9–5
- RTS**, 1–33
- run command script**, 2–7, 2–17
  - rc0, 3–10
  - rc2, 3–11

- rc3, 3–12
- run level**, 2–17
  - changing, 2–18
  - console, 3–7
  - defaults, 3–5
  - identifying, 2–17
  - initdefault, 3–7
  - initializing, 3–7
  - multiuser, 2–17
  - process, 3–8
  - single-user, 2–17
  - using init command, 2–18
  - wait, 3–7
- runacct shell script**, 10–35
- runclass command**, 3–24, 3–33
- runsyscheck**, 11–12
- rwhod daemon**
  - SysMan, A–8
- rz**, 5–40

## S

---

- sa command**, 10–26
- sanity\_check**, 5–44
- SAS**
  - bootable tape, 9–34
  - bttape, 9–34
- savecore**
  - changing dump files location, 14–12
  - command, 14–3
  - crash dump file creation, 14–5
  - creation of bounds file, 14–6
  - default location of dump files, 14–5
  - logging performed for crash dumps, 14–6
  - setting variables, 14–13
- SAVECORE\_DIR variable**
  - setting, 14–12
- SAVECORE\_FLAGS variable**
  - setting, 14–13
- /sbin/kopt command**, 4–24
- SCSI**, 5–3, 5–17
  - device name, 5–37

- device recognition, 5–49
- dynamic device recognition, 5–50
- standards supported, 5–50
- WWID, 5–8
- scsi disk**, 5–40
- scsi tape**, 5–40
- scu**, 5–9
- scu command**, 5–59
- secondary swap partition**, 14–7
- sector**
  - defined, 5–63, 6–4
- securetty**
  - securing terminal line, 3–9
- security**
  - event management, 13–18
  - SysMan, A–10
- Server System MIB**, 11–16
  - variables, 11–16
- setld**
  - using SysMan, A–11
- setting locale**, 3–17
- setup**, 1–1, 1–14
  - custom, 1–15
- sh shell**, 7–6, 7–13
- shared device**, 5–8
- shared directories**
  - unsharing, 6–36
- shared disk**, 5–31
- shared memory**
  - testing with shm<sub>x</sub>, 11–21
- sharing file system**, 6–35
  - modifying a share, 6–36
  - mounting (importing), 6–37
  - using dxfileshare, 6–35, 6–37
- shell**
  - ksh, 3–47
  - limit, 3–47
  - ulimit, 3–47
- shm<sub>x</sub> exerciser**, 11–21
  - shm<sub>x</sub>b subprocess, 11–21
  - using with mem<sub>x</sub>, 11–21
- show devices**, 2–13
- shut down**, 2–1
  - for backups, 9–10
- shutacct command**
  - syntax, 10–13
- shutdown and startup**
  - remote system, 1–35
- shutdown command**, 2–2, 2–23, 2–27, 9–10
  - changing to single-user mode, 2–19
  - console, 2–27
  - console prompt, 2–27
  - emergency, 2–29n
  - messages, 2–26
  - power off, 2–27
  - shutdown and reboot, 2–28
  - SysMan, A–13
  - using halt flag, 2–27, 2–29
  - using reboot flag, 2–28
  - warning users, 2–25
- single-user boot**, 2–6
- single-user mode**
  - accounting, 10–19
- sizer program**, 4–4
- SLIP**
  - SysMan, A–9
- slot**, 5–3
- SMP**, 2–20
  - adding cpus, 2–20
  - cpu-enable-mask attribute, 2–15
  - enable cpu, 2–15
  - rebooting failed processor, 2–20
  - unattended reboots, 2–20
- snmp\_request command**, 11–17
- socket**, 6–3
- software license**, 1–14
- software management**
  - SysMan, A–11
- sparse files**, 7–11
- spooling**
  - directory in printcap, 8–32
  - directory, 8–32
  - handling, 8–34

- queue, 8–35
- SRM**, 2–10
- SRM console**, 9–21
- stack limit**, 3–47
- standalone boot**, 2–10
  - mounting file systems, 2–11
- standalone system**, 9–34
- starting the system**, 2–30
- startup**, 2–6
- startup shell script**
  - syntax, 10–12
- static configuration**, 4–18
- static routes**
  - SysMan, A–8
- storage**, 6–29
  - SysMan, A–11
- Storage Management tools folder**, A–2
- Storage\_Management folder**
  - file system utilities, 6–27
- stty, setting**, 1–35
- subsystem**
  - configurable attributes, 4–12
  - configuring, 4–1
  - determining attribute values, 4–11
  - determining the type of, 4–8
  - dynamic subsystem, 4–8
  - kernel, 3–3
  - listing values of attributes, 4–15
- superblock**, 6–8
- SVR4 pty name space**, 5–52
- swap**
  - device, 3–46
  - monitoring with dxsysinfo, 3–47
  - partitions in sysconfigtab, 6–22
  - virtual memory, 3–46
- swap space**
  - adding, 3–43, 3–46
  - allocating, 3–43
  - allocating disk space for, 3–41
  - deferred mode, 3–45
  - deferred mode allocation, 3–45
  - description, 3–42
  - establishing size, 3–41
  - estimating, 3–43
  - estimating requirements, 3–45
  - immediate mode allocation, 3–45
  - immediate mode swapping, 3–45
  - shortage, 3–46
  - use of for storing crash dump, 14–8
- symbol names**
  - changing values in lprsetup, 8–20
- symbolic link**, 6–3
  - CDSL, 6–14
- Symmetric Multiprocessing**, 2–15
- sync command**, 2–23
- sys\_check**, 5–5, 11–11
- sys\_check command**
  - configuration report, 11–7
  - escalation report, 11–7
- sysconfig command**, 3–37, 4–6
  - using for remote subsystem management, 4–13
  - using to adjust the
    - dump\_sp\_threshold attribute, 14–14
  - using to set the
    - dump\_user\_pte\_pages attribute, 14–14
- sysconfigdb command**, 4–15
  - adding attributes with, 4–16
  - deleting subsystem entries with, 4–18
  - listing attribute values with, 4–15
  - merging attribute definitions with, 4–16
  - removing attribute definitions with, 4–17
  - updating attribute definitions with, 4–17
- sysconfigtab**, 5–53
  - setting exempt memory, 14–17
  - swap, 6–22
  - swap space, 3–46
- sysconfigtab command**, 1–33
- syslog**, 12–4, 13–2
  - and EVM, 13–2

- default syslog.conf file, 12–5
- event logging, 12–5
- log files, 12–18
- log setup in syslog.conf, 12–6
- maintaining syslogd files, 12–18
- remote messages, 12–12
- security, 12–12
- starting the daemon, 12–14
- stopping the daemon, 12–15
- syslog.auth file, 12–12
- syslog.conf, 12–18, 14–6
- syslogd daemon, 12–14, 14–7
- syslogd daemon**, 12–2
  - and EVM, 13–10
  - console messages, 1–35
- SysMan**, 1–1, 1–12, 1–15
  - account, 7–7
  - account administration, 7–1, A–6
  - administering file system, 6–27
  - AdvFS, A–11
  - AdvFS utilities, 6–29
  - ATM, A–8
  - BIND/DNS, A–9
  - bootable tape, A–12
  - class scheduler, A–7
  - classschl, 3–26
  - configuring mail, A–7
  - creating an UFS file system, 6–40
  - dismount a file system, 6–30
  - displaying mounted file system, 6–30
  - dop, A–10
  - /etc/fstab, 6–33
  - EVM viewer, A–7
  - file system utilities, 6–29
  - file systems, A–11
  - gated, A–8
  - hardware administration, A–6
  - hardware management, 5–4, 5–10
  - host equivalence, A–8
  - host file, A–8
  - installing on a client, 1–30
  - invoking, 1–15
  - iostat, A–8
  - IP router, A–8
  - LAT, A–10
  - LSM, A–11
  - LSM utilities, 6–29
  - monitoring and tuning, A–7
  - mount an NFS, 6–37
  - mounting file system, 6–33
  - mounting file system types, 6–33
  - mounting single file system, 6–33
  - network administration, A–8
  - network gateways, A–9
  - network interface, A–8
  - network interface daemon, A–10
  - NFS, A–9
  - NIS, A–10
  - NIS accounts, A–6
  - NTP, A–9
  - PPP, A–9
  - printers, A–10
  - routed, A–8
  - rwhod, A–8
  - security, A–10
  - sharing (exporting) a file system, 6–35
  - shutdown command, A–13
  - shutting down the system, 2–24
  - SLIP, A–9
  - software management, A–11
  - static routes, A–8
  - storage, A–11
  - Storage option, 6–29
  - sysman -cli, 1–18
  - system setup, 1–12, 7–6
  - UFS, A–11
  - unsharing a file system, 6–36
  - vmstat command, A–7
- sysman -cli**
  - using, 1–18

**sysman -cli command**, 1–14

**SysMan Menu**, 1–14

Accounts, 7–7

class scheduler, 3–26

class scheduling, 3–25

hardware management, 5–10

iostat command, 3–26

shut down, 2–2

shut down for backups, 9–10

sys\_check, 11–12

viewing

cluster, 5–13

CPU data, 5–15

devices, 5–13

hardware hierarchy, 5–11

vmstat command, 3–26

**SysMan Mnu**

class scheduling, 3–34

**SysMan Station**, 1–1, 1–14

CDE front panel icon, 1–11

hardware, 1–23

hardware management, 5–10

hardware tasks, 5–16

hosts, 1–23

invoking, 1–21

menu options, 1–26

monitoring devices, 5–15

network, 1–23

options, 1–26

Physical\_Fileystems view, 6–29

power management, 3–41

shut down, 2–2

storage, 1–23

viewing

device properties, 5–16

devices, 6–40

hardware data, 5–15

**system**

clock, setting, 2–21

crash, 2–9

dump created during, 14–7

logging by system, 14–6

crash dump information created

during reboot, 14–3

crash dump logging in, 14–6

customizing environment, 3–1

detecting failure, 11–14

detecting fan failure, 11–14

hardware failure, 2–9

initialization, 2–18

initialization files, 3–2

logging, 14–6

monitoring, 11–1, 11–10

monitoring utilities, 11–3

performance, 11–10

power off, 2–27

recovery from crash, 2–9

startup files, 3–2

testing, 11–1

**System Admin**

monitoring tools, 11–7

**system administration**

Application Manager, 1–1

command-line, 1–1

configuration utilities, A–2t

configuring system, tools for, 1–13

daily admin applications, A–3t

Daily Administration tools folder,  
A–2

during configuration, A–2

file system applications, A–5t

methods, 1–1

Microsoft Windows, 1–1

monitoring applications, A–4t,  
A–6t

Monitoring/Tuning tools, A–2

PCs, 1–1

remote, 1–1

reviewing logged events, 13–41

serial console, 1–1

software\_management applications,  
A–5t

storage applications, A–5t

Storage Management, A–2

SysMan, 1–1

- SysMan Station, 1-1
- system setup, 1-1, 1-14
- tools, 1-1, A-1
- tools and utilities, A-2
- tuning applications, A-4t
- Windows NT, 1-1
- X11 applications, 1-1
- system administration tools**, 1-12
- System Communication Architecture (SCA)**, C-1
- system configuration**
  - dynamic, 4-6
  - static, 4-18
- system configuration file**
  - pseudodevice entry, 5-53
- system environment**
  - remote, 1-32
- system event**, 12-1, 13-1
  - errors, 12-1
  - reporting, 12-1
- system exercisers**, 11-19
  - diagnostics, 11-19
  - getting help, 11-19
  - log files, 11-20
  - requirements for, 11-19
  - using uerf command with, 11-20
- system files**
  - EVM, 13-8
  - hardware database, 5-7
- system logs, reviewing**, 13-41
- system memory**, 11-20
- system monitoring**, 11-8
- system partition**, D-2
- system privileges**
  - sharing, A-10
- System Reference Manual**, 2-10
- System Server MIB daemon**, 11-14
- system setup**, 1-5
  - account manager, 7-6
  - dxaccounts, 7-6
  - overview, 1-14

- system shut down**, 2-1
  - automatic reboot, 2-28
  - degraded performance, 2-23
  - during high threshold levels, 11-14
  - emergency, 2-29n
  - environmental monitoring, 2-23
  - fastboot, 2-30
  - fasthalt command, 2-30
  - file system corruption, 2-23
  - for errors or events, 2-22
  - for troubleshooting, 2-22
  - from multiuser mode, 2-23
  - fsck command, 2-5
  - fsck warning, 2-28
  - graphical interface, 2-24
  - halt flag, 2-29
  - methods and options, 2-2
  - related system files, 2-4
  - related utilities, 2-5
  - system halt, 2-27
  - using SysMan, 2-2
- system startup**, 2-6, 5-44
- system tuning**, 4-5
- System\_Admin folder**, 1-9
- SYSV\_PTY**, 5-55

## T

---

- tacct file errors**
  - correcting with acctmrg, 10-39
- tape**
  - adding automatically, 5-4
  - adding tape drives manually, 5-55
  - backup, 9-1
  - bootable, 9-5
  - deleting, 5-29
  - device name, 5-37
  - device special files, 5-40, 5-42
  - device special files, converting, 5-42
  - failed or crashed, 5-34

- name, 5–30
- QIC, 5–40
- shared, 5–31
- swapping, 5–33
- tz (SCSI), 5–3
- viewing, 5–26
- worldwide identifier, 5–31
- tape drive**, 11–23
  - not installed, 9–26
- tapex exerciser**, 11–23
- tar command**, 9–5, 9–30
- target kernel**, 4–4
- /tcb/files/auth.db file**, 7–10
- TCP/IP printing**, 8–14, 8–15
- terminal communications system**
  - testing with cmx, 11–24
- terminal line**
  - enabling root logins on, 3–9
- terminals**, 3–8
- terminfo database**, 3–8
- TERMINFO environment variable**, 3–8
- testing the system**, 11–1
- tic command**, 3–8
- time**
  - setting, 2–21
- time zone**, 3–21
  - SVID, 3–21
- tip connection**, 1–32
- TL895**, 5–31
- tools**, A–2
- total accounting record**, 10–29
- Tower of Hanoi**, 9–5
- transactions**, 5–28
- troubleshooting**
  - boot operations, 2–21
  - devices, 5–33
  - event management (EVM), 13–49
  - file system, 6–1
  - files and file system, 6–49
  - printer, 8–42
  - shutdown, 2–22
  - sys\_check, 11–11

- tunefs command**, 6–49
- tuning**
  - class scheduler, A–7
  - file system, 6–1
- turnacct shell script**, 10–23
- tx**, 5–40

## U

---

- uerf command**
  - using with system exercisers, 11–20
- UFS**, 6–8
  - checking a file system, 6–45
  - creating a file system, 6–1
  - file system
    - creating, 6–16, 6–17
    - creating using SysMan, 6–40
    - creating with diskconfig, 5–61
  - file system on LSM volume, 6–40
  - file system structure, 6–8
  - monitoring and tuning, 6–1
  - quota, 6–1
  - quota files, 7–11
  - setting file system quotas, 6–41
  - sparse files, 7–11
  - structure, 6–8
  - troubleshooting, 6–1
  - version, 6–6
- UFS file system**
  - SysMan utilities, A–11
- UID**, 7–10, 7–43
  - defaults, 7–13
  - duplicate, 7–26
  - in passwd file, 7–11
  - limits, 7–12, 7–13
  - maximum number, 7–10
  - /usr/include/limits.h, 7–12
- ulimit**, 3–47
- umount command**, 6–20, 6–23
  - SysMan alternative, 6–30
- UNIX file system**, 6–8
- unmounting file system**, 6–23



- unsharing file system**, 6–36
- update installation**, 1–11
- upgrades**
  - shutdown, 2–22
- USB**, 8–2
- user**, 7–14
  - account, 7–15
  - name, 7–11
- user accounts**
  - password, A–6
  - SysMan, A–6
- user identifier**, 7–10
- user page table**
  - including in partial crash dump, 14–14
- user\_id**, 7–12
- useradd command**, 7–8
- userdel command**, 7–8
- usermod**
  - passwd, 7–6
- usermod command**, 7–8
- /usr file system**
  - restoring, 9–29
- /usr/include/limits.h file**, 7–12, 7–15
- /usr/skel file**, 7–10
- utilities for devices**, 5–57
- utmp file structure**, 10–15
- uugetty, setting**, 1–33

## V

---

- /var file system**
  - restoring, 9–29
- /var/adm logs**, 7–10
- /var/adm/crash directory**, 14–5
- /var/adm/messages**, 11–7, 12–18
- /var/adm/wtmp**
  - login log file, 3–15
- /var/yp/src file**, 7–10
- viewing**
  - cluster, 5–13

- CPU information, 5–15
- devices, 5–13
- system (devices) hierarchy, 5–11
- vipw command**, 7–5
- virtual memory**
  - address space, 3–47
  - data limit, 3–47
  - description, 3–41
  - per-process, 3–46
  - stack limit, 3–47
  - swap space, 3–46
- vm subsystem**, 3–46
- vm-swap-eager**, 3–46
- vmcore.n file**, 14–5
- vmstat command**, 3–26, 11–4, A–7
  - SysMan, A–7
- vmunix**, 2–5
- vmunix.n file**, 14–5

## W

---

- wall command**, 2–18, 2–23
- WBEM**, 1–28
- web-based management**, 1–28
- web/java mode**
  - launching online help, 1–5
- who command**, 7–6, 11–5
- Windows**, 1–15
- Windows 2000 Single Sign-On**, 7–41
- Windows NT domain**
  - ASU, 7–9
- worldwide identifier**, 5–8
- worldwide support**, 3–21
- wtmp file**
  - correcting with fwtmp command, 10–16
- wtmpfix command**, 10–15
- WWID**, 5–8
  - database, 5–8
  - device attributes, 5–31

device identifier, 5–31  
media changer, 5–31  
**wwpsmf**, 8–41

## **X**

---

**X11 applications**, A–1

## **Y**

---

**year**  
setting, 2–21