System Administration Guide: Basic Administration

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817–0798–10
April 2003

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, docs.sun.com, AnswerBook, AnswerBook2, AutoClient, JumpStart, Sun Ray, Sun Blade, PatchPro, Sun Cobalt, SunOS, Solstice, Solstice AdminSuite, Solstice DiskSuite, Solaris Solve, Java, JavaStation, OpenWindows, NFS, iPlanet, Netra and Solaris are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. DLT is claimed as a trademark of Quantum Corporation in the United States and other countries.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

Federal Acquisitions: Commercial Software–Government Users Subject to Standard License Terms and Conditions.

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 2003 Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées du système Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, docs.sun.com, AnswerBook, AnswerBook2, AutoClient, JumpStart, Sun Ray, Sun Blade, PatchPro, Sun Cobalt, SunOS, Solstice, Solstice AdminSuite, Solstice DiskSuite, Solaris Solve, Java, JavaStation, DeskSet, OpenWindows, NFS et Solaris sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. Quantum Corporation riclame DLT comme sa marque de fabrique aux Etats-Unis et dans d'autres pays.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REPONDRE A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.

030224@5533

Contents

Preface 27

1 Solaris Management Tools (Roadmap) 31
What's New in Solaris Management Tools? 31
Matrix of Solaris Management Tools Support 32
Feature Descriptions for Solaris 9 Management Tools 33
Feature Descriptions for Solaris 8 Management Tools 34
Feature Descriptions for Previous Solaris Management Tools 36
Availability Solaris Management Commands 36
Solaris 9 System Management Commands 37
Solaris 8 System Management Commands 38
Descriptions for Previous Solaris Management Commands 38
For More Information About Solaris Management Tools 39

2 Working With the Solaris Management Console (Tasks) 41
Solaris Management Console (Overview) 41
What Is the Solaris Management Console? 41
Solaris Management Console Tools 42
Why Use the Solaris Management Console? 44
Organization of the Solaris Management Console 45
Changing the Solaris Management Console Window 45
Solaris Management Console Documentation 46
How Much Role-Based Access Control? 46
Becoming Superuser (root) or Assuming a Role 47
▼ How to Become Superuser (root) or Assume a Role 48

Using the Solaris Management Tools With RBAC (Task Map) 49
If You Are the First to Log In to the Console 51
Creating the Primary Administrator Role 51
▼ How to Create the First Role (Primary Administrator) 52
▼ How to Assume the Primary Administrator Role 53
Starting the Solaris Management Console 54
▼ How to Start the Console as Superuser or as a Role 54
Using the Solaris Management Tools in a Name Service Environment (Task Map) 55
RBAC Security Files 56
Prerequisites for Using the Solaris Management Console in a Name Service Environment 57
Management Scope 58
The `/etc/nsswitch.conf` File 58
▼ How to Create a Toolbox for a Specific Environment 58
▼ How to Add a Tool to a Toolbox 60
▼ How to Start the Solaris Management Console in a Name Service Environment 61
Adding Tools to the Solaris Management Console 61
▼ How to Add a Legacy Tool to a Toolbox 62
▼ How to Install an Unbundled Tool 62
Troubleshooting the Solaris Management Console 63
▼ How to Troubleshoot the Solaris Management Console 63

3 Managing Users and Groups Topics 65

4 Managing User Accounts and Groups (Overview) 67
What's New in Managing Users and Groups? 67
Solaris Management Console Tools Suite 68
Solaris Directory Services 68
Managing Users and Resources With Projects 68
What Are User Accounts and Groups? 69
Guidelines for Managing User Accounts 70
Name Services 70
User (Login) Names 70
User ID Numbers 71
Passwords 73
Password Aging 74

Home Directories 75
User's Work Environment 75
Guidelines for Managing Groups 76
Tools for Managing User Accounts and Groups 77
What You Can Do With Solaris User Management Tools 78
Managing Home Directories With the Solaris Management Console 81
Modify User Accounts 81
Delete User Accounts 82
Add Customized User Initialization Files 82
Administer Passwords 82
Disable User Accounts 83
Where User Account and Group Information Is Stored 83
Fields in the `passwd` File 83
Fields in the `shadow` File 85
Fields in the `group` File 86
Customizing a User's Work Environment 88
Using Site Initialization Files 90
Avoid Local System References 90
Shell Features 91
Shell Environment 92
The `PATH` Variable 94
Locale Variables 95
Default File Permissions (`umask`) 96
Examples of User and Site Initialization Files 97
Example—Site Initialization File 98

5 Managing User Accounts and Groups (Tasks) 101
Setting Up User Accounts (Task Map) 101
User Information Data Sheet 102
▼ How to Customize User Initialization Files 103
▼ How to Share a User's Home Directory 104
▼ How to Mount a User's Home Directory 105
Maintaining User Accounts (Task Map) 107
Solaris User Registration 107
Accessing Solaris Solve 108
Troubleshooting Solaris User Registration Problems 108
▼ How to Restart Solaris User Registration 109

▼ How To Disable User Registration 110

6 Managing Server and Client Support Topics 111

7 Managing Server and Client Support (Overview) 113
What's New in Server and Client Management? 113
Diskless Client Support 113
Where to Find Server and Client Tasks 114
What Are Servers, Clients, and Appliances? 114
What Does Client Support Mean? 115
Overview of System Types 116
Servers 116
Standalone Systems 117
Diskless Clients 117
AutoClient Systems 118
Appliances 118
Guidelines for Choosing System Types 118
Diskless Client Management Overview 119
OS Server and Diskless Client Support Information 120
Diskless Client Management Features 120
Disk Space Requirements for OS Servers 123

8 Managing Diskless Clients (Tasks) 125
Managing Diskless Clients (Task Map) 125
Managing Diskless Clients 126
▼ How to Prepare for Adding Diskless Clients 128
▼ How to Add OS Services For Diskless Client Support 129
▼ How to Add a Diskless Client 131
▼ How to Boot a Diskless Client 132
▼ How to Delete Diskless Client Support 133
▼ How to Delete OS Services for Diskless Clients 133
Patching Diskless Client OS Services 134
Displaying OS Patches for Diskless Clients 134
▼ How to Add an OS Patch for a Diskless Client 135
Troubleshooting Diskless Client Problems 137

9 Shutting Down and Booting a System Topics 141

10 Shutting Down and Booting a System (Overview) 143
What's New in Shutting Down and Booting a System? 143
PXE Network Boot 144
Where to Find Shutting Down and Booting Tasks 144
Shutting Down and Booting Terminology 145
Guidelines for Shutting Down a System 145
Guidelines for Booting a System 146
Booting a System From the Network 147
When to Shut Down a System 147
When to Boot a System 148

11 Run Levels and Boot Files (Tasks) 151
Run Levels 151
▼ How to Determine a System's Run Level 152
The `/etc/inittab` File 153
Example—Default `inittab` File 154
What Happens When the System Is Brought to Run Level 3 155
Run Control Scripts 156
Run Control Script Summaries 157
Using a Run Control Script to Stop or Start Services 160
▼ How to Use a Run Control Script to Stop or Start a Service 161
Adding a Run Control Script 161
▼ How to Add a Run Control Script 162
Disabling a Run Control Script 162
▼ How to Disable a Run Control Script 162
x86: Boot Files 163

12 Shutting Down a System (Tasks) 165
Shutting Down the System 165
System Shutdown Commands 166
User Notification of System Down Time 167
▼ How to Determine Who Is Logged in to a System 167
▼ How to Shut Down a Server 167
▼ How to Shut Down a Standalone System 171
Turning Off Power to All Devices 172

▼ How to Turn Off Power to All Devices 173

13 SPARC: Booting a System (Tasks) 175

SPARC: Booting a System (Task Map) 175

SPARC: Using the Boot PROM 177

▼ SPARC: How to Find the PROM Revision for a System 177

▼ SPARC: How to Identify Devices on the System 177

▼ SPARC: How to Change the Default Boot Device 179

▼ SPARC: How to Reset the System 180

SPARC: Booting a System 181

▼ SPARC: How to Boot a System to Run Level 3 (Multiuser Level) 181

▼ SPARC: How to Boot a System to Run Level S (Single-User Level) 182

▼ SPARC: How to Boot a System Interactively 183

▼ SPARC: How to Boot a System From the Network 184

▼ SPARC: How to Stop the System for Recovery Purposes 186

▼ SPARC: How to Boot a System for Recovery Purposes 186

▼ SPARC: How to Boot the System With the Kernel Debugger (`kadb`) 188

SPARC: Forcing a Crash Dump and Rebooting the System 189

▼ SPARC: How to Force a Crash Dump and Reboot the System 189

14 x86: Booting a System (Tasks) 191

x86: Booting a System (Task Map) 191

x86: Booting the Solaris Device Configuration Assistant 192

▼ x86: How to Boot the Solaris Device Configuration Assistant 193

x86: Booting a System 193

▼ x86: How to Boot a System to Run Level 3 (Multiuser Level) 193

▼ x86: How to Boot a System to Run Level S (Single-User Level) 194

▼ x86: How to Boot a System Interactively 195

▼ x86: How to Boot a System From the Network 197

▼ x86: How to Stop a System for Recovery Purposes 198

▼ x86: How to Boot a System for Recovery Purposes 198

▼ x86: How to Boot a System With the Kernel Debugger (`kadb`) 203

x86: Forcing a Crash Dump and Rebooting the System 204

▼ x86: How to Force a Crash Dump and Reboot the System 204

15 The Boot Process (Reference) 207

SPARC: The Boot PROM 207

SPARC: The Boot Process 208
x86: The PC BIOS 208
x86: Boot Subsystems 209
x86: Booting the Solaris Release 210
x86: Screens Displayed During the Device Identification Phase 211
x86: Menus Displayed During the Boot Phase 212
x86: The Boot Process 214

16 Managing Removable Media Topics 217

17 Managing Removable Media (Overview) 219
What's New in Managing Removable Media? 219
Where to Find Managing Removable Media Tasks 220
Removable Media Features and Benefits 220
Comparison of Automatic and Manual Mounting 221
What You Can Do With Volume Management 222

18 Accessing Removable Media (Tasks) 223
Accessing Removable Media (Task Map) 223
Accessing Removable Media (Overview) 224
Using Removable Media Names 224
Guidelines for Accessing Removable Media Data 226
▼ How to Add a New Removable Media Drive 226
▼ How to Stop and Start Volume Management (`vold`) 227
▼ How to Access Information on Removable Media 228
▼ How to Copy Information From Removable Media 229
▼ How to Play a Musical CD or DVD 229
▼ How to Find Out If Removable Media Is Still in Use 230
▼ How to Eject Removable Media 231
Accessing Removable Media on a Remote System (Task Map) 232
▼ How to Make Local Media Available to Other Systems 232
▼ How to Access Removable Media on Remote Systems 235

19 Formatting Removable Media (Tasks) 239
Formatting Removable Media (Task Map) 239
Formatting Removable Media Overview 240

Formatting Removable Media Guidelines 240
Removable Media Hardware Considerations 241
▼ How to Load a Removable Media 242
▼ How to Format Removable Media (`rmformat`) 244
▼ How to Format Removable Media for Adding a File System 244
▼ How to Check a File System on Removable Media 246
▼ How to Repair Bad Blocks on Removable Media 247
Applying Read or Write and Password Protection to Removable Media 247
▼ How to Enable or Disable Write Protection on Removable Media 247
▼ How to Enable or Disable Read or Write Protection and a Password on Iomega Media 248

20 Writing CDs (Tasks) 251
Working with Audio and Data CDs 251
CD Media Commonly Used Terms 252
Writing Data and Audio CDs 253
Restricting User Access to Removable Media with RBAC 254
▼ How to Restrict User Access to Removable Media with RBAC 254
▼ How to Identify a CD Writer 254
▼ How to Check the CD Media 255
Creating a Data CD 256
▼ How to Create an ISO 9660 File System for a Data CD 256
▼ How to Create a Multi-Session Data CD 257
Creating an Audio CD 258
▼ How to Create an Audio CD 259
▼ How to Extract an Audio Track on a CD 260
▼ How to Copy a CD 261
▼ How to Erase CD-RW Media 261

21 Managing Software Topics 263

22 Managing Software (Overview) 265
What's New in Software Management? 265
Signed Patches 265
Solaris Product Registry 3.0 266
`prodreg` Command Enhancements 266
Patch Analyzer 266

Solaris Management Console Patch Manager 266
Where to Find Software Management Tasks 267
Overview of Software Packages 267
Tools for Managing Software Packages 268
Adding or Removing a Software Package 268
Key Points for Adding or Removing Software Packages 269
Guidelines for Removing Packages 269
Avoiding User Interaction When Adding Packages 270
Using an Administration File 270
Using a Response File 271

23 Managing Software (Tasks) 273
Commands for Managing Software Packages 274
Adding Software With the Solaris Web Start Program 274
▼ How to Install Software With the Solaris Web Start Program 275
Adding and Removing Software With the Solaris Product Registry (GUI Tasks) 276
▼ How to View Installed or Uninstalled Software Information With the Product Registry (GUI Tasks) 277
▼ How to Install Software With the Product Registry (GUI Tasks) 278
▼ How to Uninstall Software With the Product Registry (GUI Tasks) 279
Verifying and Removing Software With the Solaris Product Registry Command-Line Interface (`prodreg`) 279
▼ How to View Installed or Uninstalled Software Information With the Product Registry (`prodreg` CLI Tasks) 280
▼ How to View Software Attributes With the Product Registry (`prodreg` CLI Tasks) 283
▼ How to Check Dependencies Between Software Components (`prodreg` CLI Tasks) 285
▼ How to Identify Damaged Software Products (`prodreg` CLI Tasks) 286
▼ How to Uninstall Software in the Solaris Product Registry (`prodreg` CLI Tasks) 289
▼ How to Uninstall Damaged Software With the Product Registry (`prodreg` CLI Tasks) 293
▼ How to Reinstall Damaged Software Components (`prodreg` CLI Tasks) 295
Adding and Removing Software Packages With Admintool 297
▼ How to Add Software Packages With Admintool 298
▼ How to Remove Software Packages With Admintool 299
Adding and Removing Software Packages With Package Commands 300
▼ How to Add Software Packages (`pkgadd`) 300

Adding a Software Package to a Spool Directory 302
▼ How to List Information About All Installed Packages (pkginfo) 304
▼ How to Check the Integrity of Installed Software Packages (pkgchk) 305
Removing Software Packages 306
▼ How to Remove Software Packages (pkgrm) 306

24 Managing Solaris Patches (Overview) 309
What Is a Patch? 309
What Is a Signed Patch? 310
Accessing Solaris Patches 312
Solaris Patch Numbering 312
Tools for Managing Solaris Patches 313
Solaris Patch Management Tools for Signed Patches 314

25 Managing Solaris Patches (Tasks) 317
High-Level View of Managing Patches in the Solaris Environment (Task Map) 317
Identifying Disk Space Requirements for Patches 318
Selecting Signed or Unsigned Patches for Your Environment 319
Preparation for Managing Signed Patches (Task Map) 319
Using the Solaris Patch Management Tools 319
How to Verify Package Requirements for Signed Patch Tools 321
▼ How to Download and Install the Solaris Patch Management Tools 321
▼ How to Import Sun Certificates Into the Keystore 323
▼ How to Change the Keystore Password 324
▼ How to Set Up Your Patch Environment 324
Managing Signed Patches on a Solaris System (Task Map) 326
▼ How to Download and Add a Signed Patch on a Solaris System 326
▼ How to Remove a Signed Patch on a Solaris System 329
Identifying Problems With Signed Patches 330
Viewing Patch Tool Log Files 330
▼ How to Resolve a Sequestered Patch 331
Managing Unsigned Solaris Patches (Task Map) 332
Displaying Information About Unsigned Solaris Patches 332
▼ How to Display Information About Solaris Patches 333
Adding an Unsigned Solaris Patch 333
▼ How to Download an Unsigned Solaris Patch 334
▼ How to Add a Unsigned Solaris Patch 334

Removing an Unsigned Solaris Patch 335
▼ How to Remove an Unsigned Solaris Patch 335

26 Managing Devices Topics 337

27 Managing Devices (Tasks) 339
Where to Find Device Management Tasks 339
About Device Drivers 340
Automatic Configuration of Devices 340
Features and Benefits of Autoconfiguration 341
What You Need for Unsupported Devices 341
Displaying Device Configuration Information 342
`driver not attached` Message 342
Identifying a System's Devices 343
▼ How to Display System Configuration Information 343
▼ How to Display Device Information 345
Adding a Peripheral Device to a System 346
▼ How to Add a Peripheral Device 346
▼ How to Add a Device Driver 348

28 Dynamically Configuring Devices (Tasks) 349
Dynamic Reconfiguration and Hot-Plugging 349
Attachment Points 350
x86: Detaching PCI Adapter Cards 352
SCSI Hot-Plugging With the `cfgadm` Command (Task Map) 353
SCSI Hot-Plugging With the `cfgadm` Command 354
▼ How to Display Information About SCSI Devices 354
▼ How to Unconfigure a SCSI Controller 355
▼ How to Configure a SCSI Controller 355
▼ How to Configure a SCSI Device 356
▼ How to Disconnect a SCSI Controller 357
▼ How to Connect a SCSI Controller 358
▼ SPARC: How to Add a SCSI Device to a SCSI Bus 358
▼ SPARC: How to Replace an Identical Device on a SCSI Controller 359
▼ SPARC: How to Remove a SCSI Device 360
SPARC: Troubleshooting SCSI Configuration Problems 361
▼ How to Resolve a Failed SCSI Unconfigure Operation 363

PCI Hot-Plugging With the `cfgadm` Command (Task Map) 363
x86: PCI Hot-Plugging With the `cfgadm` Command 364
▼ x86: How to Display PCI Slot Configuration Information 364
▼ x86: How to Remove a PCI Adapter Card 365
▼ x86: How to Add a PCI Adapter Card 365
x86: Troubleshooting PCI Configuration Problems 366
Reconfiguration Coordination Manager (RCM) Script Overview 367
What Is an RCM Script? 368
What Can an RCM Script Do? 368
How Does the RCM Script Process Work? 368
RCM Script Tasks 369
Application Developer RCM Script (Task Map) 370
System Administrator RCM Script (Task Map) 370
Naming an RCM Script 371
Installing or Removing an RCM Script 371
▼ How to Install an RCM Script 372
▼ How to Remove an RCM Script 372
▼ How to Test an RCM Script 372
Tape Backup RCM Script Example 373

29 Using USB Devices (Overview/Tasks) 377
Overview of USB Devices 377
Commonly Used USB Acronyms 379
USB Bus Description 379
About USB in the Solaris Environment 381
USB Keyboards and Mouse Devices 382
USB Host Controller and Root Hub 383
SPARC: USB Power Management 383
Guidelines for USB Cables 384
Using USB Mass Storage Devices (Task Map) 384
Using USB Mass Storage Devices 385
Using Non-Compliant USB Mass Storage Devices 385
Hot-Plugging USB Devices 386
▼ How to Add a Hot-Pluggable USB Mass Storage Device With `vold` Running 386
▼ How to Add a Hot-Pluggable USB Mass Storage Device Without `vold` Running 387

▼ How to Remove a Hot-Pluggable USB Mass Storage Device With `vold` Running 387
▼ How to Remove a Hot-Pluggable USB Mass Storage Device Without `vold` Running 387
Mounting USB Mass Storage Devices With or Without `vold` Running 388
How to Mount or Unmount a USB Mass Storage Device With `vold` Running 389
How to Mount or Unmount a USB Mass Storage Device Without `vold` Running 390
▼ How to Add a Hot-Pluggable USB Camera 390
Using USB Audio Devices (Task Map) 392
Using USB Audio Devices 392
Hot-Plugging Multiple USB Audio Devices 393
▼ How to Add Hot-Pluggable USB Audio Devices 394
▼ How to Identify Your System's Primary Audio Device 394
▼ How to Change the Primary USB Audio Device 395
▼ How to Remove Unused USB Audio Device Links 397
Troubleshooting USB Audio Device Problems 397
Solving USB Speaker Problems 398
Hot-Plugging USB Devices With the `cfgadm` Command (Task Map) 398
Hot-Plugging USB Devices With the `cfgadm` Command 399
▼ How to Display USB Device Information 400
▼ How to Unconfigure a USB Device 401
▼ How to Configure a USB Device 402
▼ How to Logically Disconnect a USB Device 402
▼ How to Logically Connect a USB Device 402
▼ How to Logically Disconnect a USB Device Subtree 403
▼ How to Reset a USB Device 403

30 Accessing Devices (Overview) 405
Accessing Devices 405
How Device Information Is Created 405
How Devices Are Managed 406
Device Naming Conventions 406
Logical Disk Device Names 407
Specifying the Disk Subdirectory 407
Specifying the Slice 408
x86: Disks With Direct Controllers 408

SPARC: Disks With Bus-Oriented Controllers 409
x86: Disks With SCSI Controllers 409
Logical Tape Device Names 410
Logical Removable Media Device Names 411

31 Managing Disks Topics 413

32 Managing Disks (Overview) 415
What's New in Disk Management in the Solaris 9 4/03 Release? 415
Overview of the EFI Disk Label 415
What's New in Disk Management in the Solaris 9 Release? 419
Solaris Volume Manager and Soft Partitioning 419
Where to Find Disk Management Tasks 420
Overview of Disk Management 420
Disk Terminology 420
About Disk Slices 421
SPARC: Disk Slices 422
x86: Disk Slices 423
Using Raw Data Slices 425
Slice Arrangements on Multiple Disks 425
Determining Which Slices to Use 425
The `format` Utility 426
When to Use the `format` Utility 427
Guidelines for Using the `format` Utility 428
Formatting a Disk 429
About Disk Labels 430
Partition Table 430
Displaying Partition Table Information 431
Dividing a Disk Into Slices 432
Using the Free Hog Slice 433

33 Administering Disks (Tasks) 435
Administering Disks (Task Map) 435
Identifying Disks on a System 436
▼ How to Identify the Disks on a System 436
Formatting a Disk 438
▼ How to Determine if a Disk is Formatted 439

▼ How to Format a Disk 439
Displaying Disk Slices 441
▼ How to Display Disk Slice Information 441
Creating and Examining a Disk Label 443
▼ How to Label a Disk 443
▼ How to Examine a Disk Label 445
Recovering a Corrupted Disk Label 446
▼ How to Recover a Corrupted Disk Label 447
Adding a Third-Party Disk 449
Creating a `format.dat` Entry 449
▼ How to Create a `format.dat` Entry 450
Automatically Configuring SCSI Disk Drives 450
▼ How to Automatically Configure a SCSI Drive 451
Repairing a Defective Sector 452
▼ How to Identify a Defective Sector by Using Surface Analysis 453
▼ How to Repair a Defective Sector 454
Tips and Tricks for Managing Disks 455
Debugging `format` Sessions 455
Label Multiple Disks by Using the `prtvtoc` and `fmthard` Commands 455

34 SPARC: Adding a Disk (Tasks) 457
SPARC: Adding a System Disk or a Secondary Disk (Task Map) 457
SPARC: Adding a System Disk or a Secondary Disk 458
▼ SPARC: How to Connect a System Disk and Boot 459
▼ SPARC: How to Connect a Secondary Disk and Boot 459
▼ SPARC: How to Create Disk Slices and Label a Disk 460
▼ SPARC: How to Create File Systems 465
▼ SPARC: How to Install a Boot Block on a System Disk 466

35 x86: Adding a Disk (Tasks) 467
x86: Adding a System Disk or a Secondary Disk (Task Map) 467
x86: Adding a System or Secondary Disk 468
▼ x86: How to Connect a System Disk and Boot 468
▼ x86: How to Connect a Secondary Disk and Boot 469
x86: Guidelines for Creating an `fdisk` Partition 470
▼ x86: How to Create a Solaris `fdisk` Partition 471
▼ x86: How to Create Disk Slices and Label a Disk 476

▼ x86: How to Create File Systems 478
▼ x86: How to Install a Boot Block on a System Disk 479

36 The `format` Utility (Reference) 481
Recommendations and Requirements for Using The `format` Utility 481
Format Menu and Command Descriptions 482
The `partition` Menu 484
x86: The `fdisk` Menu 485
The `analyze` Menu 486
The `defect` Menu 487
The `format.dat` File 488
Contents of the `format.dat` File 489
Syntax of the `format.dat` File 489
Keywords in the `format.dat` File 489
Partition or Slice Tables (`format.dat`) 492
Specifying an Alternate Data File for the `format` utility 492
Rules for Input to `format` Commands 493
Specifying Numbers to `format` Commands 493
Specifying Block Numbers to `format` Commands 493
Specifying `format` Command Names 494
Specifying Disk Names to `format` Commands 495
Getting Help on the `format` Utility 495

37 Managing File Systems Topics 497

38 Managing File Systems (Overview) 499
What's New in File Systems? 499
Extended File Attributes 499
UFS Snapshots 500
Improved UFS Direct I/O Concurrency 500
Improved `mkfs` Performance 501
New `labelit` Options for UDF File Systems 501
Where to Find File System Management Tasks 502
Overview of File Systems 502
Types of File Systems 503
Disk-Based File Systems 503
Network-Based File Systems 504

Virtual File Systems 504
Commands for File System Administration 507
How File System Commands Determine the File System Type 508
Manual Pages for Generic and Specific Commands 508
The Default Solaris File Systems 509
Swap Space 510
The UFS File System 510
UFS Logging 511
Planning UFS File Systems 512
UFS Direct Input/Output (I/O) 512
Mounting and Unmounting File Systems 513
The Mounted File System Table 515
The Virtual File System Table 516
The NFS Environment 517
Automounting or AutoFS 517
Determining a File System's Type 518
▼ How to Determine a File System's Type 518

39 Creating File Systems (Tasks) 521
Creating a UFS File System 521
Default Parameters for the `newfs` Command 522
▼ How to Create a UFS File System 522
Creating a Temporary File System (TMPFS) 524
▼ How to Create a TMPFS File System 524
Creating a Loopback File System (LOFS) 525
▼ How to Create an LOFS File System 526

40 Mounting and Unmounting File Systems (Tasks) 529
Overview of Mounting File Systems 529
Commands for Mounting and Unmounting File Systems 530
Commonly Used Mount Options 531
Field Descriptions for the `/etc/vfstab` File 532
Mounting File Systems 534
▼ How to Determine Which File Systems Are Mounted 534
▼ How to Add an Entry to the `/etc/vfstab` File 535
▼ How to Mount a File System (`/etc/vfstab` File) 536
▼ How to Mount a UFS File System (`mount` Command) 537

▼ How to Mount a UFS File System Without Large Files (`mount` Command) 538
▼ How to Mount an NFS File System (`mount` Command) 539
▼ x86: How to Mount a PCFS (DOS) File System From a Hard Disk (`mount` Command) 540
Unmounting File Systems 541
Prerequisites for Unmounting File Systems 542
▼ How to Verify a File System is Unmounted 542
▼ How to Stop All Processes Accessing a File System 542
▼ How to Unmount a File System 543

41 Using The CacheFS File System (Tasks) 545
High-Level View of Using the CacheFS File System (Task Map) 545
Overview of the CacheFS File System 546
How a CacheFS File System Works 546
CacheFS File System Structure and Behavior 547
Creating and Mounting a CacheFS File System (Task Map) 548
▼ How to Create the Cache 549
Mounting a File System in the Cache 549
▼ How to Mount a CacheFS File System (`mount`) 550
▼ How to Mount a CacheFS File System (`/etc/vfstab`) 552
▼ How to Mount a CacheFS File System (AutoFS) 553
Maintaining a CacheFS File System (Task Map) 553
Maintaining a CacheFS File System 554
Modifying a CacheFS File System 554
▼ How to Display Information About a CacheFS File System 555
Consistency Checking of a CacheFS File System 556
▼ How to Specify Cache Consistency Checking on Demand 556
▼ How to Delete a CacheFS File System 556
▼ How to Check the Integrity of a CacheFS File System 558
Packing a Cached File System (Task Map) 559
Packing a CacheFS File System 559
▼ How to Pack Files in the Cache 560
▼ How to Display Packed Files Information 561
Using Packing Lists 562
▼ How to Create a Packing List 562
▼ How to Pack Files in the Cache With a Packing List 563
Unpacking Files or Packing Lists From the Cache 563

▼ How to Unpack Files or Packing Lists From the Cache 563
Troubleshooting `cachefspack` Errors 564
Collecting CacheFS Statistics (Task Map) 568
Collecting CacheFS Statistics 568
How to Set Up CacheFS Logging 570
How to Locate the CacheFS Log File 570
How to Stop CacheFS Logging 571
How to View the Working Set (Cache) Size 571
Viewing CacheFS Statistics 572
How to View CacheFS Statistics 572

42 Configuring Additional Swap Space (Tasks) 575
About Swap Space 575
Swap Space and Virtual Memory 576
Swap Space and the TMPFS File System 576
Swap Space as a Dump Device 577
How Do I Know If I Need More Swap Space? 577
Swap-Related Error Messages 578
TMPFS-Related Error Messages 578
How Swap Space Is Allocated 578
The `/etc/vfstab` File 579
Planning for Swap Space 579
Monitoring Swap Resources 580
Adding More Swap Space 581
Creating a Swap File 581
▼ How to Create a Swap File and Make It Available 582
Removing a Swap File From Use 583
▼ How to Remove Unneeded Swap Space 583

43 Checking UFS File System Consistency (Tasks) 585
File System Consistency 585
How the File System State Is Recorded 586
What the `fsck` Command Checks and Tries to Repair 588
Why Inconsistencies Might Occur 588
The UFS Components That Are Checked for Consistency 588
The `fsck` Summary Message 594
Interactively Checking and Repairing a UFS File System 594

▼ How to See If a File System Needs Checking 595
▼ How to Check File Systems Interactively 595
Preening UFS File Systems 596
▼ How to Preen a UFS File System 597
Fixing a UFS File System That the `fsck` Command Cannot Repair 597
Restoring a Bad Superblock 598
▼ How to Restore a Bad Superblock 598
Syntax and Options for the `fsck` Command 600

44 UFS File System (Reference) 601
Default Directories for root (/) and `/usr` File Systems 601
The Platform-Dependent Directories 609
The Structure of Cylinder Groups for UFS File Systems 609
The Boot Block 610
The Superblock 610
Inodes 610
Data Blocks 612
Free Blocks 612
Custom File System Parameters 613
Logical Block Size 613
Fragment Size 614
Minimum Free Space 615
Rotational Delay 615
Optimization Type 615
Number of Files 616
Maximum UFS File and File System Size 616
Maximum Number of UFS Subdirectories 616
Commands for Creating a Customized File System 617
The `newfs` Command Syntax, Options, and Arguments 617
The Generic `mkfs` Command 619

45 Backing Up and Restoring Files and File Systems Topics 621

46 Backing Up and Restoring File Systems (Overview) 623
What's New in Backing Up and Restoring File Systems? 623
UFS Snapshots 623
Where to Find Backup and Restore Tasks 624

Definition: Backing Up and Restoring File Systems 624
Why You Should Back Up File Systems 625
Planning Which File Systems to Back Up 625
Choosing the Type of Backup 627
Choosing a Tape Device 628
High-Level View of Backing Up and Restoring File Systems (Task Map) 629
Guidelines for Scheduling Backups 630
How Often Should You Do Backups? 630
Using Dump Levels to Create Incremental Backups 630
Sample Backup Schedules 632
Example—Daily Cumulative, Weekly Cumulative Backups 632
Example—Daily Cumulative, Weekly Incremental Backups 633
Example—Daily Incremental, Weekly Cumulative Backups 634
Example—Monthly Backup Schedule for a Server 635
Suggestions for Scheduling Backups 638

47 Backing Up Files and File Systems (Tasks) 641
Backing Up Files and File System (Task Map) 641
Preparing for File System Backups 642
▼ How to Find File System Names 642
▼ How to Determine the Number of Tapes Needed for a Full Backup 643
Backing Up a File System 643
▼ How to Backup a File System to Tape 644

48 Using UFS Snapshots (Tasks) 651
Using UFS Snapshots (Task Map) 651
UFS Snapshots Overview 652
Why Use UFS Snapshots? 652
UFS Snapshots Performance Issues 653
Creating and Deleting UFS Snapshots 653
▼ How to Create a UFS Snapshot 654
▼ How to Display UFS Snapshot Information 654
Deleting a UFS Snapshot 655
▼ How to Delete a UFS Snapshot 655
Backing Up a UFS Snapshot 656
▼ How to Create a Full Backup of a UFS Snapshot (`ufsdump`) 656
▼ How to Create an Incremental Backup of a UFS Snapshot (`ufsdump`) 657

▼ How to Back Up a UFS Snapshot (tar) 658
Restoring Data From a UFS Snapshot Backup 658

49 Restoring Files and File Systems (Tasks) 659
Restoring Files and File System Backups (Task Map) 659
Preparing to Restore Files and File Systems 660
Determining the File System Name 660
Determining the Type of Tape Device You Need 661
Determining the Tape Device Name 661
Restoring Files and File Systems 661
▼ How to Determine Which Tapes to Use 662
▼ How to Restore Files Interactively 663
▼ How to Restore Specific Files Non-Interactively 665
▼ How to Restore a Complete File System 667
▼ How to Restore the root (/) and /usr File Systems 670

50 UFS Backup and Restore Commands (Reference) 673
How the ufsdump Command Works 673
Determining Device Characteristics 673
Detecting the End of Media 674
Copying Data With ufsdump 674
Role of the /etc/dumpdates File 674
Backup Device (*dump-file*) Argument 675
Specifying Files to Back Up 677
Specifying Tape Characteristics 677
Limitations of the ufsdump Command 677
Options and Arguments for the ufsdump Command 678
Default ufsdump Options 678
Options for the ufsdump Command 678
The ufsdump Command and Security Issues 680
Options and Arguments for the ufsrestore Command 681

51 Copying UFS Files and File Systems (Tasks) 685
Commands for Copying File Systems 685
Copying File Systems Between Disks 687
Making a Literal File System Copy 687
▼ How to Copy a Disk (dd) 688

Copying Directories Between File Systems (cpio Command) 690
▼ How to Copy Directories Between File Systems (cpio) 690
Copying Files and File Systems to Tape 691
Copying Files to Tape (tar Command) 693
▼ How to Copy Files to a Tape (tar) 693
▼ How to List the Files on a Tape (tar) 694
▼ How to Retrieve Files From a Tape (tar) 694
Copying Files to a Tape With the pax Command 695
▼ How to Copy Files to a Tape (pax) 695
Copying Files to Tape With the cpio Command 696
▼ How to Copy All Files in a Directory to a Tape (cpio) 696
▼ How to List the Files on a Tape (cpio) 698
▼ How to Retrieve All Files From a Tape (cpio) 698
▼ How to Retrieve Specific Files From a Tape (cpio) 699
Copying Files to a Remote Tape Device 700
▼ How to Copy Files to a Remote Tape Device (tar and dd) 700
▼ How to Extract Files From a Remote Tape Device 702
Copying Files and File Systems to Diskette 702
Things You Should Know When Copying Files to Diskettes 703
▼ How to Copy Files to a Single Formatted Diskette (tar) 703
▼ How to List the Files on a Diskette (tar) 704
▼ How to Retrieve Files From a Diskette (tar) 704
▼ How to Archive Files to Multiple Diskettes 705
Copying Files With a Different Header Format 706
▼ How to Create an Archive for Older SunOS Releases 706
Retrieving Files Created With the bar Command 707
▼ How to Retrieve bar Files From a Diskette 707

52 Managing Tape Drives (Tasks) 709
Choosing Which Media to Use 709
Backup Device Names 710
Specifying the Rewind Option for a Tape Drive 711
Specifying Different Densities for a Tape Drive 712
Displaying Tape Drive Status 712
▼ How to Display Tape Drive Status 712
Handling Magnetic Tape Cartridges 713
▼ How to Retension a Magnetic Tape Cartridge 713

▼ How to Rewind a Magnetic Tape Cartridge 714
Guidelines for Drive Maintenance and Media Handling 714

Index 715

Preface

System Administration Guide: Basic Administration is part of a set that includes a significant part of the Solaris™ system administration information. This guide contains information for both SPARC® based and x86 based systems.

This book assumes you have

- Installed the SunOS™ 5.9 operating system
- Set up all the networking software that you plan to use

The SunOS 5.9 operating system is part of the Solaris product family, which also includes many features, including the Solaris Common Desktop Environment (CDE). The SunOS 5.9 operating system is compliant with AT&T's System V, Release 4 operating system.

For the Solaris 9 release, new features interesting to system administrators are covered in sections called *What's New in ... ?* in the appropriate chapters.

Note – The Solaris operating environment runs on two types of hardware, or platforms, SPARC and x86. The Solaris operating environment runs on both 64–bit and 32–bit address spaces. The information in this document pertains to both platforms and address spaces unless called out in a special chapter, section, note, bullet, figure, table, example, or code example.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the Solaris 9 release. To use this book, you should have 1-2 years of UNIX® system administration experience. Attending UNIX system administration training courses might be helpful.

How the System Administration Volumes Are Organized

Here is a list of the topics that are covered by the volumes of the System Administration Guides.

Book Title	Topics
System Administration Guide: Basic Administration	User accounts and groups, server and client support, shutting down and booting a system, removable media, managing software (packages and patches), disks and devices, file systems, and backing up and restoring data
System Administration Guide: Advanced Administration	Printing services, terminals and modems, system resources (disk quotas, accounting, and crontabs), system processes, and troubleshooting Solaris software problems
System Administration Guide: IP Services	TCP/IP networks, IPv4 and IPv6, DHCP, IP Security, Mobile IP, and IP Network Multipathing
System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)	DNS, NIS, and LDAP naming and directory services
System Administration Guide: Naming and Directory Services (FNS and NIS+)	FNS and NIS+ naming and directory services
System Administration Guide: Resource Management and Network Services	Resource management, remote file systems, mail, SLP, and PPP
System Administration Guide: Security Services	Auditing, PAM, RBAC, and SEAM

To view license terms, attribution, and copyright statements for open source software included in this release of the Solaris operating environment, the default path is `/usr/share/src/`*freeware-name* or `/usr/sfw/share/src/`*freeware-name*. If the Solaris operating environment has been installed anywhere other than the default location, modify the given path to access the file at the installed location.

Accessing Sun Documentation Online

The docs.sun.com^{SM} Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

What Typographic Conventions Mean

The following table describes the typographic conventions used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`** `Password:`
AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. Do *not* save changes yet.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

General Conventions

Be aware of the following conventions used in this book.

- When following steps or using examples, be sure to type double-quotes (`"`), left single-quotes (`‘`), and right single-quotes (`’`) exactly as shown.
- The key referred to as Return is labeled Enter on some keyboards.
- The root path usually includes the `/sbin`, `/usr/sbin`, `/usr/bin`, and `/etc` directories, so the steps in this book show the commands in these directories without absolute path names. Steps that use commands in other, less common, directories show the absolute paths in the examples.
- The examples in this book are for a basic SunOS software installation without the Binary Compatibility Package installed and without `/usr/ucb` in the path.

Caution – If `/usr/ucb` is included in a search path, it should always be at the end of the search path. Commands like `ps` or `df` are duplicated in `/usr/ucb` with different formats and options from the SunOS commands.

CHAPTER 1

Solaris Management Tools (Roadmap)

This chapter provides a roadmap to Solaris management tools.

- "What's New in Solaris Management Tools?" on page 31
- "Matrix of Solaris Management Tools Support" on page 32
- "Feature Descriptions for Solaris 9 Management Tools" on page 33
- "Feature Descriptions for Solaris 8 Management Tools" on page 34
- "Feature Descriptions for Previous Solaris Management Tools" on page 36
- "Availability Solaris Management Commands" on page 36
- "For More Information About Solaris Management Tools" on page 39

What's New in Solaris Management Tools?

These tools are new or changed in the Solaris 9 release:

- Diskless client support
- Solaris DHCP
- Resource Management
- Solaris Management Console (referred to as the console) tools suite
- Solaris Volume Manager (previously Solstice™ DiskSuite)

The following table provides a brief description of each tool and where to find more information about them.

TABLE 1–1 New or Changed Solaris Management Tools in the Solaris 9 Release

Solaris Administration Tool	Description	For More Information
Diskless Client Support	Provides a command-line interface for managing diskless client systems.	Chapter 8
Resource Management	Enables you to control how applications use available system resources.	*System Administration Guide: Resource Management and Network Services*
Solaris DHCP	Provides improved performance, capacity, and flexibility in managing DHCP in your network.	"About Solaris DHCP (Overview)" in *System Administration Guide: IP Services*
Solaris Management Console [1]	Serves as a launching point for a variety of GUI-based system management tools.	This guide and the console online help
Solaris Volume Manager (previously Solstice™ DiskSuite)	Provides robust storage management in the Solaris 9 release and is launched from the Solaris Management Console. The command-line interface is also available.	*Solaris Volume Manager Administration Guide*

[1] Do not confuse this tool with Sun Management Center (SunMC). For information about the Sun Management Center product, see `http://www.sun.com/solaris/sunmanagementcenter/docs`.

Matrix of Solaris Management Tools Support

This section provides information about tools that are primarily used to manage users, groups, clients, disks, printers, and serial ports.

This table lists the various Solaris management GUI tools and whether they are currently supported.

TABLE 1–2 Matrix of Solaris Management Tool Support

	Solaris 2.6 and Earlier Releases	Solaris 7	Solaris 8	Solaris 9
admintool	Supported	Supported	Supported	Supported
Solstice AdminSuite 2.3	Supported	Supported	Not Supported	Not Supported
Solstice AdminSuite 3.0	Supported (Solaris 2.6 release only)	Supported	Supported	Not Supported
Solaris Management Tools 1.0	Supported	Supported	Supported	Not Supported
Solaris Management Tools 2.0	Not Supported	Not Supported	Supported (Solaris 8 01/01, 4/01, 7/01, 10/01, 2/02 releases only)	Not Supported
Solaris Management Tools 2.1	Not Supported	Not Supported	Not Supported	Supported

If you want to perform administration tasks on a system with a text-based terminal as the console, use Solaris Management Console commands instead. For more information, see Table 1–6.

Feature Descriptions for Solaris 9 Management Tools

This table describes the tools available in the Solaris 9 release.

TABLE 1–3 Feature Descriptions for Solaris 9 Management Tools

Feature or Tool	Supported in admintool?	Supported in Solaris Management Console 2.1
AutoClient Support	No	No
Computers and Networks Tool	No	Yes

TABLE 1–3 Feature Descriptions for Solaris 9 Management Tools *(Continued)*

Feature or Tool	Supported in admintool?	Supported in Solaris Management Console 2.1
Diskless Client Support	No	Yes, a diskless client CLI is available
Disks Tool	No	Yes
Enhanced Disk Tool (Solaris Volume Manager)	No	Yes
Job Scheduler	No	Yes
Log Viewer	No	Yes
Mail Alias Support	No	Yes
Mounts and Shares Tool	No	Yes
Name Service Support	No	For users, groups, and network information only
Patch Tool	No	Yes
Performance Tool	No	Yes
Printer Support	Yes	Solaris Print Manager is available separately
Projects Tool	No	Yes
RBAC Support	No	Yes
RBAC Tool	No	Yes
Serial Port Tool	Yes	Yes
Software Package Tool	Yes	No
System Information Tool	No	Yes
User/Group Tool	Yes	Yes

Feature Descriptions for Solaris 8 Management Tools

This table lists the tools that are available in the Solaris 8 release and various Solaris 8 update releases.

TABLE 1–4 Feature Descriptions for Solaris 8 Management Tools

Feature or Tool	Supported in admintool?	Supported in Solstice AdminSuite 3.0? (Solaris 8 and Solaris 8 6/00 and 10/00 only)	Supported in Solaris Management Console 1.0?	Supported in Solaris Management Console 2.0? (Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02 only)
AutoClient/Diskless Client Support	No	No (but an AutoClient CLI is available separately)	No	No (but a diskless CLI and AutoClient CLI is available separately)
Disks Tool	No	No	No	Yes
Job Scheduler	No	No	No	Yes
Log Viewer	No	Yes	No	Yes
Mail Alias Support	No	Yes	No	Yes
Mounts and Shares Tool	No	Yes	No	Yes
Name Service Support	No	Yes	No	For users, groups, and network information only
Printer Support	Yes	Solaris Print Manager is available	Yes	No, but Solaris Print Manager is available
Software Package Tool	Yes	No	Yes	No
RBAC Support	No	Yes (rights support only)	No	Yes
RBAC Tool	No	RBAC CLI is available separately	No	Yes
Serial Port Tool	Yes	Yes	Yes	Yes
User/Group Tool	Yes	Yes	Yes	Yes

Feature Descriptions for Previous Solaris Management Tools

This table describes the tools that are available in releases prior to the Solaris 8 release.

TABLE 1–5 Feature Descriptions for Previous Solaris Management Tools

Feature or Tool	Supported in admintool?	Supported in Solstice AdminSuite 2.3?	Supported in Solstice AdminSuite 3.0? (Solaris 2.6 only)
AutoClient/Diskless Client Support	No	Yes	No (but an AutoClient CLI is available separately)
Disks Tool	No	Yes	No
Log Viewer	No	No	Yes
Mail Alias Support	No	Yes	Yes
Mounts and Shares Tool	No	Yes	Yes
Name Service Support	No	Yes	Yes
Printer Support	Yes	Yes	Solaris Print Manager is available
RBAC Support	No	No	Yes (rights support only)
RBAC Tool	No	No	RBAC CLI is available separately
Serial Port Tool	Yes	Yes	Yes
User/Group Tool	Yes	Yes	Yes

Availability Solaris Management Commands

This series of tables lists commands that perform the same tasks as the Solaris management tools. For information on diskless client support, see Chapter 8.

Solaris 9 System Management Commands

This table describes the commands that provide the same functionality as the Solaris management tools. You must be superuser or assume an equivalent role to use these commands. Some of these commands are for the local system only. Others commands operate in a name service environment. See the appropriate man page and refer to the `-D` option.

TABLE 1–6 Descriptions for Solaris Management Commands

Command	Description	Man Page
`smc`	Starts the Solaris Management Console	`smc`(1M)
`smcron`	Manages `crontab` jobs	`smcron`(1M)
`smdiskless`	Manages diskless client support	`smdiskless`(1M)
`smexec`	Manages entries in the `exec_attr` database	`smexec`(1M)
`smgroup`	Manages group entries	`smgroup`(1M)
`smlog`	Manages and views WBEM log files	`smlog`(1M)
`smmultiuser`	Manages bulk operations on multiple user accounts	`smmultiuser`(1M)
`smosservice`	Adds OS services and diskless client support	`smosservice`(1M)
`smprofile`	Manages profiles in the `prof_attr` and `exec_attr` databases	`smprofile`(1M)
`smrole`	Manages roles and users in role accounts	`smrole`(1M)
`smserialport`	Manages serial ports	`smserialport`(1M)
`smuser`	Manages user entries	`smuser`(1M)

This table describes the commands you can use to manage RBAC from the command line. You must be superuser or assume an equivalent role to use these commands. These commands cannot be used to manage RBAC information in a name service environment.

TABLE 1–7 RBAC Command Descriptions

Command	Description	References
`auths`	Displays authorizations granted to a user	`auths`(1)
`profiles`	Displays execution profiles for a user	`profiles`(1)
`roleadd`	Adds a new role to the system	`roleadd`(1M)
`roles`	Displays roles granted to a user	`roles`(1)

This table describes the commands you can use to manage users, groups, and RBAC features from the command line. You must be superuser or assume an equivalent role to use these commands. These commands cannot be used to manage user and group information in a name service environment.

TABLE 1–8 Solaris User/Group Command Descriptions

Command	Description	References
`useradd`, `usermod`, `userdel`	Adds, modifies, or removes a user.	`useradd`(1M), `usermod`(1M), `userdel`(1M)
`groupadd`, `groupmod`, `groupdel`	Adds, modifies, or removes a group.	`groupadd`(1M), `groupmod`(1M), `groupdel`(1M)

Solaris 8 System Management Commands

All of the commands listed Table 1–7 and Table 1–8 are available in the Solaris 8 release.

Descriptions for Previous Solaris Management Commands

This table describes the commands that provide equivalent functionality to the Solstice AdminSuite™ 2.3 and Solstice AutoClient™ 2.3 GUI tools. You must be superuser or be a member of the sysadmin group to use these commands.

Note – The Solstice AdminSuite 2.3 and Solstice AutoClient 2.3 command man pages are not available online. You must have access to the Solstice AdminSuite 2.3 and Solstice AutoClient 2.3 software to view these man pages.

All of the commands listed in Table 1–8 are also available in previous Solaris releases.

TABLE 1–9 Descriptions for Solstice AdminSuite™ 2.3/Solstice AutoClient™ 2.1 Commands

Command	Description	For More Information
`admhostadd`, `admhostmod`, `admhostdel`, `admhostls`	Adds, modifies, removes, and lists support for client and server systems set up with the AdminSuite software	*Solstice AdminSuite 2.3 Administration Guide* and *Solstice AutoClient 2.1 Administration Guide*
`admuseradd`, `admusermod`, `admuserdel`, `admuserls`, `admgroupadd`, `admgroupmod`, `admgroupdel`, `admgroupls`	Adds, modifies, removes, and lists users and groups	*Solstice AdminSuite 2.3 Administration Guide*

For More Information About Solaris Management Tools

This table identifies where to find more information about Solaris management tools.

TABLE 1–10 For More Information About Solaris Management Tools

Tool	Availability	For More Information
Solaris Management Console 2.1 suite of tools	Solaris 9 release	This guide and the console online help
Solaris Management Console 2.0 suite of tools	Solaris 8 1/01, 4/01, 7/01, 10/01, and 2/02 releases	The Solaris Management Console online help
Solaris Management Console 1.0 suite of tools	Solaris 2.6, Solaris 7, and Solaris 8 releases	*Solaris Easy Access Server 3.0 Installation Guide*
admintool	Solaris 9, Solaris 8, and previous Solaris releases	`admintool`(1M)

TABLE 1–10 For More Information About Solaris Management Tools *(Continued)*

Tool	Availability	For More Information
AdminSuite 2.3	Solaris 2.4, Solaris 2.5, Solaris 2.5.1, Solaris 2.6, and Solaris 7 releases	*Solstice AdminSuite 2.3 Administration Guide*
AdminSuite 3.0	Solaris 8, Solaris 8 6/00, and Solaris 8 10/00 releases	*Solaris Easy Access Server 3.0 Installation Guide*
AutoClient 3.0.1	Solaris 8 releases	Call your local service provider
	Solaris 9 releases – Consider using the Web Start Flash installation feature	*Solaris 9 Installation Guide*
Diskless Client CLI	Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02, and Solaris 9 releases	Chapter 8

CHAPTER 2

Working With the Solaris™ Management Console (Tasks)

This chapter provides an overview of the Solaris management tools used to perform system administration tasks. Topics include starting the Solaris Management Console (console), setting up Role-Based Access Control (RBAC) to use with the console, and working with the Solaris management tools in a name service environment.

For information on the procedures associated with performing system management tasks with the Solaris Management Console, see:

- "Using the Solaris Management Tools With RBAC (Task Map)" on page 49
- "Using the Solaris Management Tools in a Name Service Environment (Task Map)" on page 55

For information on troubleshooting Solaris Management Console problems, see "Troubleshooting the Solaris Management Console" on page 63.

Solaris Management Console (Overview)

The following sections provide information about the Solaris Management Console.

What Is the Solaris Management Console?

The Solaris Management Console is a container for GUI-based management tools that are stored in collections referred to as *toolboxes*. The console includes a default toolbox with many basic management tools, including tools for managing users, projects, and cron jobs; for mounting and sharing file systems; and for managing disks and serial ports. For a brief description of each Solaris management tool, see Table 2–1.

You can always add tools to the existing toolbox, or you can create new toolboxes.

The Solaris Management Console has three primary components:

- The Solaris Management Console Client

  Called *console*, this is the visible interface and contains the GUI tools used to perform management tasks.

- The Solaris Management Console Server

  This component is located either on the same machine as the console or remotely, and provides all the *back end* functionality that allows management through the console.

- The Solaris Management Console Toolbox Editor

  This application, which looks similar to the console, is used to add or modify toolboxes, to add tools to a toolbox, or to extend the scope of a toolbox. For example, you would add a toolbox to manage a name service domain.

The default toolbox is visible when you start the console.

Solaris Management Console Tools

This table describes the tools included in the default Solaris Management Console toolbox and provides cross-references to background information for each tool.

TABLE 2–1 Solaris Management Console Tool Suite

Category	Tool	Description	For More Information
System Status	System Information	Monitors and manages system information such as date, time, and timezone.	"Displaying and Changing System Information (Tasks)" in *System Administration Guide: Advanced Administration*
	Log Viewer	Monitors and manages the Solaris Management Console tools log and system logs.	"Troubleshooting Software Problems (Overview)" in *System Administration Guide: Advanced Administration*
	Processes	Monitors and manages system processes.	"Processes and System Performance" in *System Administration Guide: Advanced Administration*

TABLE 2–1 Solaris Management Console Tool Suite *(Continued)*

Category	Tool	Description	For More Information
	Performance	Monitors system performance.	"Managing System Performance (Overview)" in *System Administration Guide: Advanced Administration*
System Configuration	Users	Manages users, rights, roles, groups, and mailing lists.	"What Are User Accounts and Groups?" on page 69 and "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*
	Projects	Creates and manages entries in the `/etc/project` database.	"Projects and Tasks" in *System Administration Guide: Resource Management and Network Services*
	Computers and Networks	Creates and monitors computer and network information.	Solaris Management Console online help
	Patches	Manages patches.	Chapter 24
Services	Scheduled Jobs	Creates and manages scheduled `cron` jobs.	"Ways to Automatically Execute System Tasks" in *System Administration Guide: Advanced Administration*
Storage	Mounts and Shares	Mounts and shares file systems.	Chapter 38
	Disks	Creates and manages disk partitions.	Chapter 32
	Enhanced Storage	Creates and manages volumes, hot spare pools, state database replicas, and disk sets.	*Solaris Volume Manager Administration Guide*
Devices and Hardware	Serial Ports	Sets up terminals and modems.	"Managing Terminals and Modems (Overview)" in *System Administration Guide: Advanced Administration*

Context–sensitive help is available after you start a tool. For broader, more in-depth online information than the context help provides, see the expanded help topics, which you can reach from the console Help menu.

Why Use the Solaris Management Console?

The console provides a set of tools with many benefits for administrators. The console does the following:

- Supports all experience levels

  Inexperienced administrators can complete tasks by using the graphical interface, which includes dialog boxes, wizards, and context help. Experienced administrators will find that the console provides a convenient, secure alternative to using `vi` to manage hundreds of configuration parameters spread across tens or hundreds of systems.

- Controls user access to the system

  Although any user can access the console by default, only superuser can make changes in the initial configuration. As described in "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*, it is possible to create special user accounts called *roles* that can be assigned to users, typically administrators, who are permitted to make specific system changes.

  The key benefit of RBAC is that roles can be limited to only those tasks that are necessary for doing their jobs. RBAC is *not* required for using the Solaris management tools. You can run all tools as superuser without making any changes.

- Provides a command line interface

  If preferred, administrators can operate the Solaris management tools through a command-line interface (CLI). Some commands are written specifically to mimic the GUI tool functions, such as the commands for managing users. These new commands are listed in Table 1–6, with the names and brief descriptions of each command. There is also a man page for each command.

  For those Solaris management tools that have no special commands, such as the Mounts and Shares tools, use the standard UNIX commands.

For in-depth information about how RBAC works, its benefits, and how to apply those benefits to your site, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

To learn more about using RBAC with the Solaris management tools, see "Using the Solaris Management Tools With RBAC (Task Map)" on page 49.

Organization of the Solaris Management Console

In the following figure, the console is shown with the Users Tool open.

The main part of the console consists of three panes:

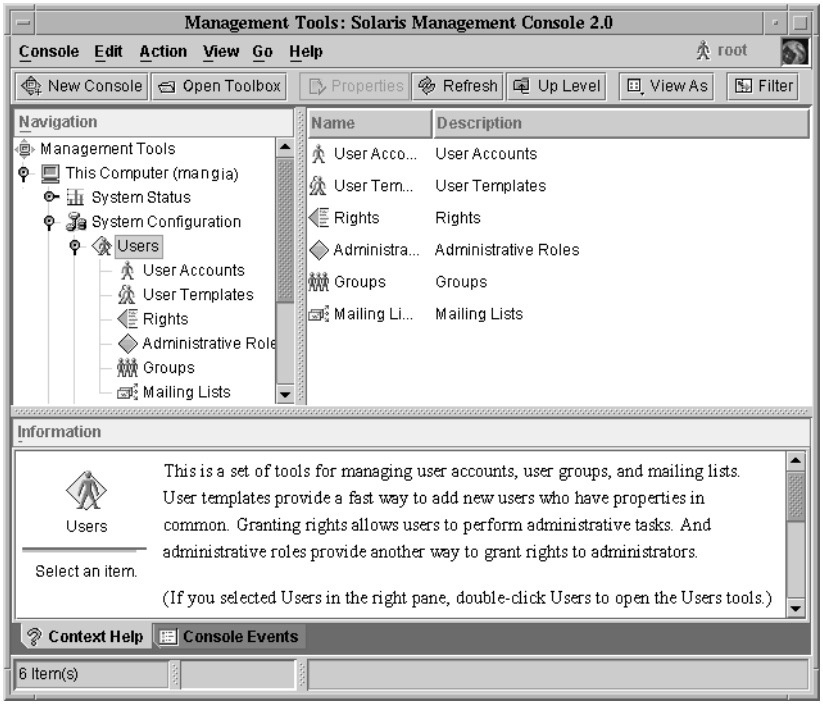


FIGURE 2–1 Solaris Management Console – Users Tool

- Navigation pane (at the left) – For accessing tools (or sets of tools), folders, or other toolboxes. Icons in the navigation pane are called nodes and are expandable if they are folders or toolboxes.
- View pane (at the right) – For viewing information related to the node selected in the navigation pane, shows either the contents of the selected folder, subordinate tools, or data associated with the selected tool.
- Information pane (at the bottom) – For displaying context-sensitive help or console events.

Changing the Solaris Management Console Window

The layout of the console window is highly configurable. You can use the following features to change the console window layout:

- View menu – Use the Show option in the View menu to hide or display the optional bars and panes. The other options in the View menu control the display of nodes in the view pane.
- Console menu – Use the Preferences option to set the following: the initial toolbox, the orientation of panes, clicking or double-clicking for selection, text or icons in the tool bar, fonts, default tool loading, authentication prompts, and advanced logins.
- Context Help or Console Events toggles – Use the icons at the bottom of the information pane to toggle between the display of context-sensitive help and console events.

Solaris Management Console Documentation

The main source of documentation for using the console and its tools is the online help system. Two forms of online help are available: context-sensitive help and expanded help topics.

- Context-sensitive help responds to your use of the console tools.

  Clicking the cursor on tabs, entry fields, radio buttons, and so forth, causes the appropriate help to appear in the Information pane. You can close, or reopen the Information pane by clicking the question mark button on dialog boxes and wizards.
- Expanded help topics are available from the Help menu or by clicking cross reference links in some context-sensitive help.

  These topics appear in a separate viewer and contain more in-depth information than is provided by the context help. Topics include overviews of each tool, explanations of how each tool works, files used by a specific tool, and troubleshooting.

For a brief overview of each tool, refer to Table 2–1.

How Much Role-Based Access Control?

As described in “Why Use the Solaris Management Console?” on page 44, a major advantage of using the Solaris management tools is the ability to use Role-Based Access Control (RBAC). RBAC provides administrators with access to just the tools and commands they need to perform their jobs.

Depending on your security needs, you can use varying degrees of RBAC, as follows:

RBAC Approach	Description	For More Information
No RBAC	Allows you to perform all tasks as superuser. You can log in as yourself. When you select a Solaris management tool, you enter root as the user and the root password.	"How to Become Superuser (root) or Assume a Role" on page 48
Root as a Role	Eliminates anonymous root logins and prevents users from logging in as root. This approach requires users to log in as themselves before they assume the root role. Note that you can apply this technique whether or not you are using other roles.	"Making a Role" in *System Administration Guide: Security Services*
Single Role Only	Uses the Primary Administrator role, which is roughly equivalent to having root access only.	"Creating the Primary Administrator Role" on page 51
Suggested Roles	Uses three roles that are easily configured: Primary Administrator, System Administrator, and Operator. These roles are appropriate for organizations with administrators at different levels of responsibility whose job capabilities roughly fit the suggested roles.	"Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*
Custom Roles	You can add your own roles, depending on your organization's security needs.	"Planning for RBAC" in *System Administration Guide: Security Services*

Becoming Superuser (root) or Assuming a Role

Most administration tasks, such as adding users, file systems, or printers, require that you first log in as root (UID=0) or assume a role if you are using RBAC. The root account, also known as the *superuser* account, is used to make system changes and can override user file protection in emergency situations.

The superuser account and roles should be used only to perform administrative tasks to prevent indiscriminate changes to the system. The security problem associated with the superuser account is that a user has complete access to the system even when performing minor tasks.

In a non-RBAC environment, you can either log into the system as superuser or use the `su` command to change to the superuser account. If RBAC is implemented, you can assume roles through the console or use `su` and specify a role.

When you use the console to perform administration tasks, you can do one of the following:

- Log into the console as yourself and then supply the root user name and password.
- Log into the console as yourself and then assume a role.

A major benefit of RBAC is that roles can be created to give limited access to specific functions only. If you are using RBAC, you can run restricted applications by assuming a role rather than becoming superuser.

For step-by-step instructions on creating the Primary Administrator role, see "How to Create the First Role (Primary Administrator)" on page 52. For an overview on configuring RBAC to use roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

▼ How to Become Superuser (root) or Assume a Role

Become superuser or assume a role by using one of the following methods. Each method requires that you know either the superuser password or the role password.

1. Select one of the following to become superuser.
   - Log in as a user, start the Solaris Management Console, select a Solaris management tool, and then log in as root.

     This method enables to you perform any management task from the console.

     For information on starting the Solaris Management Console, see "How to Start the Solaris Management Console in a Name Service Environment" on page 61.
   - Log in as superuser on the system console.

     ```
     hostname console: root
     Password: root-password
     #
     ```

     The pound sign (#) is the Bourne shell prompt for the superuser account.

     This method provides complete access to all system commands and tools.
   - Log in as a user, and then change to the superuser account by using the `su` command at the command line.

```
% su
Password: root-password
#
```

This method provides complete access to all system commands and tools.

- Log in remotely as superuser. This method is not enabled by default. You must modify the `/etc/default/login` file to remotely log in as superuser on the system console. For information on modifying this file, see "Securing Machines (Tasks)" in *System Administration Guide: Security Services*.

  This method provides complete access to all system commands and tools.

2. Select one of the following to assume a role.

   - Log in as user, and then change to a role by using the `su` command at the command line.

     ```
     % su role
     Password: role-password
     $
     ```

     This method provides access to all the commands and tools the role has access to.

   - Log in as a user, start the Solaris Management Console, select a Solaris management tool, and then assume a role.

     For information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 54.

     This method provides access to the Solaris management tools that the role has access to.

Using the Solaris Management Tools With RBAC (Task Map)

This task map describes the tasks to do if you want to use the Role-Based Access Control (RBAC) security features rather than use the superuser account to perform administration tasks.

Note – The information in this chapter describes how to use the console with RBAC. RBAC overview and task information is included to show you how to initially setup RBAC with the console.

For detailed information on RBAC and using it with other applications, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

Task	Description	For Instructions
1. Start the console	If your user account is already set up, start the console as yourself, and then log in to the console as root. If you do not have a user account set up, become superuser first, and then start the console.	"How to Start the Console as Superuser or as a Role" on page 54
2. Add a user account for yourself	Add a user account for yourself if you do not have one already.	Solaris Management Console online help
3. Create the Primary Administrator role	Create the Primary Administrator role and add yourself to this role.	"How to Create the First Role (Primary Administrator)" on page 52
4. Assume the Primary Administrator role	Assume the Primary Administrator role after you have created this role.	"How to Assume the Primary Administrator Role" on page 53
5. (Optional) Make root a role	Make root a role and add yourself to the root role so that no other user can use the `su` command to become root.	"Making a Role" in *System Administration Guide: Security Services*
6. (Optional) Create other administrative roles	Create other administrative roles and grant the appropriate rights to each role. Then, add the appropriate users to each role.	"How to Create a Role by Using the Administrative Roles Tool" in *System Administration Guide: Security Services*

The following sections provide overview information and step-by-step instructions for using the Solaris Management Console and the RBAC security features.

If You Are the First to Log In to the Console

If you are the first administrator to log in to the console, start the console as a user (yourself), and then log in as superuser. This method gives you complete access to all the console tools.

Here are the general steps, depending on whether or not you are using RBAC:

- *Without RBAC* – If you choose not to use RBAC, continue working as superuser. All other administrators will also need root access to perform their jobs.
- *With RBAC* – You'll need to do the following:
  - Set up your user account, if you do not already have one.
  - Create the role called Primary Administrator.
  - Assign the Primary Administrator right to the role you are creating.
  - Assign your user account to this role.

    For step-by-step instructions on creating the Primary Administrator role, see "How to Create the First Role (Primary Administrator)" on page 52.

    For an overview on configuring RBAC to use roles, see "Configuring RBAC (Task Map)" in *System Administration Guide: Security Services*.

Creating the Primary Administrator Role

An administrative role is a special user account. Users who assume a role are permitted to perform a pre-defined set of administrative tasks.

The Primary Administrator role is permitted to perform all administrative functions, similar to superuser.

If you are superuser, or a user assuming the Primary Administrator role, you can define which tasks other administrators are permitted to perform. With the help of the Add Administrative Role wizard, you can create a role, grant rights to the role, and then specify which users are permitted to assume that role. A right is a named collection of commands, or authorizations, for using specific applications or for performing specific functions within an application, and other rights, whose use can be granted or denied by an administrator.

You are prompted for the following information when you create the Primary Administrator role:

TABLE 2–2 Item Descriptions for Adding a Role by Using the Console

Item	Description
Role Name	Selects the name an administrator uses to log in to a specific role.
Full Name	Provides a full, descriptive name of this role. (Optional)
Description	Further description of this role.
Role ID Number	Selects the identification number assigned to this role. This number is the same as the set of identifiers for UIDs.
Role Shell	Selects the shell that runs when a user logs into a terminal or console window and assumes a role in that window.
Create a role mailing list	Creates a mailing list with the same name as the role, if checked. You can use this list to send email to everyone assigned to the role.
Role Password and Confirm Password	Sets and confirms the role password and password.
Available Rights and Granted Rights	Assigns rights to this role by choosing from the list of Available Rights and adding them to the list of Granted Rights.
Select a home directory	Selects the home directory server where this role's private files will be stored.
Assign users to this role	Adds specific users to the role so they can assume the role to perform specific tasks.

For detailed information about Role-Based Access Control, and how to use roles to create a more secure environment, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

▼ How to Create the First Role (Primary Administrator)

This procedure describes how to create the Primary Administrator role and then assign it to your user account. This procedure assumes that your user account is already created.

1. **Start the console as yourself.**

```
% /usr/sadm/bin/smc &
```

For additional information on starting the console, see “How to Start the Console as Superuser or as a Role” on page 54.

See the console online help if you need to create a user account for yourself.

2. **Click This Computer icon in the Navigation pane.**
3. **Click System Configuration->Users->Administrative Roles.**
4. **Click Action->Add Administrative Role.**

   The Add Administrative Role wizard opens.
5. **Create the Primary Administrator role with the Administrative Role wizard by following these steps.**

   a. **Identify the role name, full role name, description, role ID number, role shell, and whether you want to create a role mailing list. Click Next.**

   b. **Set and confirm the role password. Click Next.**

   c. **Select the Primary Administrator right from the Available Rights column and add it to Granted Rights column. Click Next.**

   d. **Select the home directory for the role. Click Next.**

   e. **Assign yourself to the list of users who can assume the role. Click Next.**

   If necessary, see Table 2–2 for a description of the role items.
6. **Click Finish.**

▼ How to Assume the Primary Administrator Role

After you have created the Primary Administrator role, log in to the console as yourself, and then assume the Primary Administrator role.

When you assume a role, you take on all the attributes of that role, including the rights. At the same time, you relinquish all of your own user properties.

1. **Start the console.**

   ```
   % /usr/sadm/bin/smc &
   ```

   For information on starting the console, see “How to Start the Console as Superuser or as a Role” on page 54.
2. **Log in with your user name and password.**

   A list shows which roles you are permitted to assume.
3. **Log in to the Primary Administrator role and provide the role password.**

Starting the Solaris Management Console

The following procedure describes how to start the console and gain access to the Solaris management tools.

▼ How to Start the Console as Superuser or as a Role

If you start the console as a user, with your own user account, you have limited access to the Solaris management tools. For greater access, you can log in as yourself and then as one of the roles you are allowed to assume. If you are permitted to assume the role of Primary Administrator, you then have access to all the Solaris management tools, equivalent to that of superuser.

1. **Verify that you are in a window environment, such as the CDE environment.**
2. **Start the console in one of the following ways.**
   - From the command line, type:

     ```
     % /usr/sadm/bin/smc &
     ```

     It might take a minute or two for the console to come up the first time.
   - From the Tools menu of the CDE front panel.
   - By double-clicking a Solaris Management Console icon in CDE's Applications Manager or File Manager.

   The Solaris Management Console window is displayed.

Note – Open a console in your window environment to display the Solaris Management Console start-up messages. Do not attempt to start the Solaris Management Console server manually before starting the Solaris Management Console. The server starts automatically when you start the Solaris Management Console. For information on troubleshooting console problems, see "Troubleshooting the Solaris Management Console" on page 63.

---

3. **Double-click the This Computer icon under the Management Tools icon in the Navigation pane.**

   A list of categories is displayed.
4. **(Optional) Select the appropriate toolbox.**

If you want to use a toolbox other than the default toolbox, select the appropriate toolbox from the Navigation pane. Or, select Open Toolbox from the console menu and load the toolbox you want.

For information about using different toolboxes, see "How to Create a Toolbox for a Specific Environment" on page 58.

5. **Double-click the category icon to access a particular tool.**

   Use the online help to identify how to perform a specific task.

6. **Double-click the tool icon.**

   A popup Log-In window is displayed.

7. **Decide if you want to the tool as superuser or as a role.**

   - If you are logging in as superuser and will be working as superuser, select step 8.
   - If you are logging in as yourself and will be assuming the Primary Administrator role, select steps 9 and 10.

8. **If you are logging in as superuser, enter the root password.**

9. **If you are logging in as yourself, backspace over the root user name. Then supply your user ID and user password.**

   A list of roles you can assume is displayed.

10. **Select the Primary Administrator role, or an equivalent role, and supply the role password.**

    For step-by-step instructions on creating the Primary Administrator role, see "How to Create the First Role (Primary Administrator)" on page 52.

    The main tool menu is displayed.

Using the Solaris Management Tools in a Name Service Environment (Task Map)

By default, the Solaris management tools are set up to operate in a local environment. For example, the Mounts and Shares tool enables you to mount and share directories on specific systems, but not in a NIS or NIS+ environment. However, you can manage information with the Users and Computers and Networks tools in a name service environment.

To work with a console tool in a name service environment, you need to create a name service toolbox, and then add the tool to that toolbox.

Task	Description	For Instructions
1. Verify prerequisites	Verify you have completed the prerequisites before attempting to use the console in a name service environment.	"Prerequisites for Using the Solaris Management Console in a Name Service Environment" on page 57
2. Create a toolbox for the name service	Use the New Toolbox wizard to create a toolbox for your name service tools.	"How to Create a Toolbox for a Specific Environment" on page 58
3. Add a tool to the name service toolbox	Add the Users tool, or any other name service tool, to your name service toolbox.	"How to Add a Tool to a Toolbox" on page 60
4. Select the toolbox just created	Select the toolbox you just created to manage name service information.	"How to Start the Solaris Management Console in a Name Service Environment" on page 61

RBAC Security Files

The RBAC security files that work with the Solaris Management Console are created when you upgrade to or install the Solaris 9 release. If you do not install the Solaris Management Console packages, the RBAC security files are installed without the necessary data for using RBAC. For information on the Solaris Management Console packages, see "Troubleshooting the Solaris Management Console" on page 63.

The RBAC security files in the Solaris 9 release are included in your name service so that you can use the Solaris Management Console tools in a name service environment.

The security files on a local server are populated into a name service environment as part of a standard upgrade by the `ypmake`, `nispopulate`, or equivalent LDAP commands. The following name services are supported:

- NIS
- NIS+
- LDAP
- files

Note – The `projects` database is not supported in the NIS+ environment.

The RBAC security files are created when you upgrade to or install the Solaris 9 release.

This table briefly describes the pre-defined security files that are installed on a Solaris 9 system.

TABLE 2–3 RBAC Security Files

Local File Name	Table or Map Name	Description
`/etc/user_attr`	`user_attr`	Associates users and roles with authorizations and rights profiles.
`/etc/security/auth_attr`	`auth_attr`	Defines authorizations and their attributes and identifies associated help files.
`/etc/security/prof_attr`	`prof_attr`	Defines rights profiles, lists the rights profiles assigned authorizations and identifies associated help files.
`/etc/security/exec_attr`	`exec_attr`	Defines the privileged operations assigned to a rights profile.

For unusual upgrade cases, you might have to use the `smattrpop` command to populate RBAC security files in the following instances:

- When creating or modifying rights profiles, or
- When you need to include users and roles by customizing the `usr_attr` file.

For more information, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

Prerequisites for Using the Solaris Management Console in a Name Service Environment

The following table identifies what you need to do before you can use the Solaris Management Console in a name service environment.

Prerequisite	For More Information
Install the Solaris 9 release.	*Solaris 9 Installation Guide*
Set up your name service environment.	*System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*
Select your management scope.	"Management Scope" on page 58

Prerequisite	For More Information
Make sure your/etc/nsswitch.conf file is configured so that you can access your name service data.	"The /etc/nsswitch.conf File" on page 58

Management Scope

The Solaris Management Console uses the term *management scope* to refer to the name service environment that you want to use with the selected management tool. The management scope choices for the Users and Computers and Networks tools are LDAP, NIS, NIS+, or files.

The management scope that you select during a console session should correspond to the primary name service identified in the /etc/nsswitch.conf file.

The /etc/nsswitch.conf File

The /etc/nsswitch.conf file on each system specifies the policy for name service lookups (where data is read from) on that system.

Note – You must make sure that the name service accessed from the console, which you specify through the console Toolbox Editor, appears in the search path of the /etc/nsswitch.conf file. If the specified name service does not appear there, the tools might behave in unexpected ways, resulting in errors or warnings.

When using the Solaris managements tools in a name service environment, you might impact many users with a single operation. For example, if you delete a user in the NIS name service, that user is deleted on all systems that are using NIS.

If different systems in your network have different /etc/nsswitch.conf configurations, unexpected results might occur. So, all systems to be managed with the Solaris management tools should have a consistent name service configuration.

▼ How to Create a Toolbox for a Specific Environment

Applications for administering the Solaris operating environment are called tools, and those tools are stored in collections referred to as *toolboxes*. A toolbox can be located on a local server, where the console is located, or on a remote machine.

Use the Toolbox Editor to add a new toolbox, to add tools to an existing toolbox, or to change the scope of a toolbox. For example, to change the domain from local files to a name service.

Note – You can start the Toolbox Editor as a normal user. However, if you plan to make changes and save them to the default console toolbox (`/var/sadm/smc/toolboxes`), you must start the Toolbox Editor as `root`.

---

1. **Start the Toolbox Editor.**

   ```
   # /usr/sadm/bin/smc edit &
   ```

2. **Select Open from the Toolbox menu.**
3. **Select the This Computer icon in the Toolboxes: window.**
4. **Click Open.**
   The This Computer toolbox opens in the window.
5. **Select the This Computer icon again in the Navigation pane.**
6. **Select Add Folder from the Action menu.**
7. **Use the Folder wizard to add a new toolbox for your name service environment.**
   a. **Name and Description – Provide a name in the Full Name window. Click Next.**
      For example, “NIS tools” for the NIS environment.
   b. **Provide a description in the Description window. Click Next.**
      For example, “tools for NIS environment.”
   c. **Icons – Use the default value for the Icons. Click Next.**
   d. **Management Scope – Select Override.**
   e. **Select your name service under the Management Scope pull-down menu.**
   f. **Add the name service master name in the Server: field, if necessary.**
   g. **Add the domain managed by the server in the Domain: field.**
   h. **Click Finish.**
      The new toolbox appears in the left Navigation pane.
8. **Select the new toolbox icon.**
9. **Select Save As from the Toolbox menu.**

10. **Enter the toolbox path name in the Local Toolbox Filename: dialog box. Use the `.tbx` suffix.**

    `/var/sadm/smc/toolboxes/this_computer/`*toolbox-name*`.tbx`

11. **Click Save.**

    The new toolbox appears in the Navigation pane in the console window.

Where to Go From Here

After you have created a name service toolbox, you can put a name service tool into it. For more information, see "How to Add a Tool to a Toolbox" on page 60.

▼ How to Add a Tool to a Toolbox

In addition to the default tools that ship with the console, additional tools that can be launched from the console are being developed. As these tools become available, you can add one or more tools to an existing toolbox.

You can also create a new toolbox, for either local management or network management, and then add tools to the new toolbox.

1. **Become superuser or assume an equivalent role.**
2. **Start the Toolbox Editor, if necessary.**

   `# /usr/sadm/bin/smc edit &`

3. **Select the toolbox.**

   If you want to work in a name service, select the toolbox you just created in the Toolbox Editor.

   For more information, see "How to Create a Toolbox for a Specific Environment" on page 58.

4. **Select Add Tool from the Action menu.**
5. **Use the Add Tool wizard to add the new tool.**

   a. **Server Selection – Add the name service master in the Server: window. Click Next.**

   b. **Tools Selection – Select the tool you want to add from the Tools: window. Click Next.**

      If this tool box is a name service toolbox, choose a tool you want to work in a name service environment. For example, the Users Tools.

   c. **Name and Description – Accept the default values. Click Next.**

d. **Icons – Accept the default values, unless you have created custom icons. Click Next.**

e. **Management Scope – Accept the default value "Inherit from Parent." Click Next.**

f. **Tool Loading – Accept the default "Load tool when selected." Click Finish.**

6. **Select Save from the Toolbox menu to save the updated toolbox.**
   The Local Toolbox window is displayed.

▼ How to Start the Solaris Management Console in a Name Service Environment

After you have created a name service toolbox and have added tools to it, you can start the Solaris Management Console and open that toolbox to manage a name service environment.

1. **Verify that the following prerequisites are met.**

   a. **Be sure the system you are logged into is configured to work in a name service environment.**

   b. **Verify that the `/etc/nsswitch.conf` file is configured to match your name service environment.**

2. **Start the Solaris Management Console.**
   For more information, see "How to Start the Console as Superuser or as a Role" on page 54.

3. **Select the toolbox you created for the name service, which appears in the Navigation pane.**
   For information on creating a toolbox for a name service, see "How to Create a Toolbox for a Specific Environment" on page 58.

Adding Tools to the Solaris Management Console

This section describes how to add legacy tools or unbundled tools to the console. If you want to add authentication to these tools, see "Securing Legacy Applications" in *System Administration Guide: Security Services*.

▼ How to Add a Legacy Tool to a Toolbox

A legacy tool is any application that was not designed specifically as a Solaris management tool. You can add three types of legacy tool applications, X applications, command-line interface, and HTML, to a console toolbox. Each tool you add to a toolbox can then be launched from the Solaris Management Console.

1. **Become superuser or assume an equivalent role.**
2. **Start the Solaris Management Console Toolbox Editor, if necessary.**

   ```
   # /usr/sadm/bin/smc edit &
   ```

3. **Open the toolbox to which you want to add the legacy application.**
   The toolbox selected is opened in the Toolbox Editor.
4. **Select the node in the toolbox to which you want to add the legacy application.**
   A legacy application can be added to the top node of a toolbox or to another folder.
5. **Click Action->Add Legacy Application.**
   The first panel of the Legacy Application Wizard: General is displayed.
6. **Follow the instructions in the wizard.**
7. **Save the toolbox in the Editor.**

▼ How to Install an Unbundled Tool

Follow this procedure if you want to add a new tool package that can be launched from the console.

1. **Become superuser or assume an equivalent role.**
2. **Install the new tool package.**

   ```
   # pkgadd ABCDtool
   ```

3. **Restart the console so that it recognizes the new tool.**
   a. **Stop the console server.**

      ```
      # /etc/init.d/init.wbem stop
      ```

   b. **Start the console server.**

      ```
      # /etc/init.d/init.wbem start
      ```

4. **Start the console to verify that the new tool is displayed.**
   For more information, see "How to Start the Console as Superuser or as a Role" on page 54.

Troubleshooting the Solaris Management Console

Before using this troubleshooting procedure, make sure the following packages are installed:

```
SUNWmc          Solaris Management Console 2.1 (Server Components)
SUNWmcc         Solaris Management Console 2.1 (Client Components)
SUNWmccom       Solaris Management Console 2.1 (Common Components)
SUNWmcdev       Solaris Management Console 2.1 (Development Kit)
SUNWmcex        Solaris Management Console 2.1 (Examples)
SUNWwbmc        Solaris Management Console 2.1 (WBEM Components)
```

These packages provide the basic Solaris Management Console launcher. You must install the `SUNWprog` cluster to use the Solaris Management Console and all of its tools.

▼ How to Troubleshoot the Solaris Management Console

The client and the server are started automatically when you start the Solaris Management Console.

If the console is visible and you are having trouble running the tools, it might be that the server is not running. Or, the server might be in a problem state that can be resolved by stopping and restarting it.

1. **Become superuser or assume an equivalent role.**
2. **Determine whether the console server is running.**

   ```
   # /etc/init.d/init.wbem status
   ```

   If the console server is running, you should see a message like the following:

   ```
   SMC server version 2.1.0 running on port 898.
   ```

3. **If the console server is not running, start it.**

   ```
   # /etc/init.d/init.wbem start
   ```

   After a short time, you should see a message like the following:

   ```
   SMC server is ready.
   ```

4. **If the server is running and you are still having problems, stop the console server and then restart it.**

a. Stop the console server.

```
# /etc/init.d/init.wbem stop
```

You should see a message like the following:

```
Shutting down SMC server on port 898.
```

b. Start the console server.

```
# /etc/init.d/init.wbem start
```

CHAPTER 3

Managing Users and Groups Topics

This topic map lists the chapters that provide information on managing users and groups.

Chapter 4	Provides overview information about managing user accounts and groups.
Chapter 5	Provides step-by-step instructions for managing user accounts and groups.

CHAPTER 4

Managing User Accounts and Groups (Overview)

This chapter provides guidelines and planning information for managing user accounts and groups. This chapter also includes information about customizing the user's work environment.

This is a list of the overview information in this chapter.

- "What's New in Managing Users and Groups?" on page 67
- "What Are User Accounts and Groups?" on page 69
- "Guidelines for Managing User Accounts" on page 70
- "Guidelines for Managing Groups" on page 76
- "Tools for Managing User Accounts and Groups" on page 77
- "Where User Account and Group Information Is Stored" on page 83
- "Customizing a User's Work Environment" on page 88

For step-by-step instructions on managing user accounts and groups, see Chapter 5.

What's New in Managing Users and Groups?

This section describes new features for managing users and groups in the Solaris 9 release.

Solaris Management Console Tools Suite

The Solaris Management tools suite, available from the Solaris Management Console, enable you to manage all user and group features. For information on using the Solaris Management Console, see Chapter 2. For information on performing specific user and group management tasks, see "What You Can Do With Solaris User Management Tools" on page 78.

Solaris Directory Services

You can manage user and group information in a LDAP (Lightweight Directory Access Protocol) directory service with the iPlanet™ Directory Server, as well as other LDAP directory servers. Managing user and group information is also available in the NIS, NIS+, or files environment.

For information on setting up LDAP, see *System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)*.

For information on managing users and groups with iPlanet Directory Server, see *iPlanet Directory Server 5.1 Administrator's Guide* at the `http://docs.iplanet.com` site.

Managing Users and Resources With Projects

In the Solaris 9 release, users and groups can be members of a *project*, an identifier that indicates a workload component that can be used as the basis of system usage or resource allocation chargeback. Projects are part of the Solaris resource management feature that is used to manage system resources.

Users need to be a member of a project to successfully log in to a system running the Solaris 9 release. By default, users are a member of the `group.staff` project when the Solaris 9 release is installed and no other project information is configured.

User project information is stored in the `/etc/project` file, which can be stored on the local system (files), the NIS name service, or the LDAP directory service. You can use the Solaris Management Console to manage project information.

The `/etc/project` file must exist for users to log in successfully, but requires no administration if you are not using projects.

For more information on using or setting up projects, see "Projects and Tasks" in *System Administration Guide: Resource Management and Network Services*.

What Are User Accounts and Groups?

One basic system administration task is to set up a user account for each user at a site. A typical user account includes the information a user needs to log in and use a system, without having the system's root password. User account information has the following components:

Component	Description
User name	A name that a user uses to log in to a system. This name is also known as a login name.
Password	A secret combination of characters that a user must enter with a user name to gain access to a system.
User's home directory	A directory that is usually the user's current directory at login. The user's home directory typically contains most of the user's files.
User initialization files	Shell scripts that control how the user's working environment is set up when a user logs in to a system.

Also, when you set up a user account, you can add the user to predefined groups of users. A typical use of groups is to set up group permissions on a file and directory, which allows access only to users who are part of that group.

For example, you might have a directory containing confidential files that only a few users should be able to access. You could set up a group called `topsecret` that includes the users working on the `topsecret` project. And, you could set up the `topsecret` files with read permission for the `topsecret` group. That way, only the users in the `topsecret` group would be able to read the files.

A special type of user account, called a *role*, is used to give selected users special privileges. For more information, see "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services*.

A user or group can be a member of one or more projects. A project is an identifier that is used to chargeback system resources. For information on using projects, see "Projects and Tasks" in *System Administration Guide: Resource Management and Network Services*.

Guidelines for Managing User Accounts

The following sections describe some guidelines and planning information for creating user accounts.

Name Services

If you are managing user accounts for a large site, you might want to consider using a name or directory service such as LDAP, NIS, or NIS+. A name or directory service enables you to store user account information in a centralized manner instead of storing user account information in every system's `/etc` files. When using a name or directory service for user accounts, users can move from system to system using the same user account without having site-wide user account information duplicated on every system. Using a name or directory service also promotes centralized and consistent user account information.

User (Login) Names

User names, also called login names, let users access their own systems and remote systems that have the appropriate access privileges. You must choose a user name for each user account you create.

Keep the following guidelines in mind when creating user or role names:

- Be unique within your organization, which might span multiple domains
- Contain from two to eight letters and numerals. The first character should be a letter and at least one character should be a lowercase letter.

Even though user names can include a period (.), underscore (_), or hyphen (-), using these characters is not recommended because they can cause problems with some software products.

Consider establishing a standard way of assigning user names so they are easier for you to track. Also, names should be easy for users to remember. A simple scheme when selecting a user name is to use the first name initial and first seven letters of the user's last name. For example, Ziggy Ignatz becomes `zignatz`. If this scheme results in duplicate names, you can use the first initial, middle initial, and the first six characters of the user's last name. For example, Ziggy Top Ignatz becomes `ztignatz`. If this scheme still results in duplicate names, consider using the following scheme:

- The first initial, middle initial, first five characters of the user's last name,
- and the number 1, or 2, or 3, and so on, until you have a unique name.

Note – Each new user name must be distinct from any mail aliases known to the system or to an NIS or NIS+ domain. Otherwise, mail might be delivered to the alias rather than to the actual user.

User ID Numbers

Associated with each user name is a user identification (UID) number. The user UID identifies the user name to any system on which the user attempts to log in. And, the user UID is used by systems to identify the owners of files and directories. If you create user accounts for a single individual on a number of different systems, always use the same user name and user ID. In that way, the user can easily move files between systems without ownership problems.

UID numbers must be a whole number less than or equal to 2147483647. UID numbers are required for both regular user accounts and special system accounts. The following table lists the UID numbers reserved for user accounts and system accounts.

TABLE 4–1 Reserved UID Numbers

User ID Numbers	Use or Login Accounts	Description
0 - 99	root, `daemon`, `bin`, `sys`, and so on.	System accounts
100 - 2147483647	Regular users	General purpose accounts
60001 and 65534	`nobody` and `nobody4`	Anonymous users
60002	`noaccess`	Non-trusted users

Although UID numbers 0 through 99 are reserved, you can add a user with one of these numbers. However, do not use 0 through 99 for regular user accounts. By definition, root always has UID 0, `daemon` has UID 1, and pseudo-user `bin` has UID 2. In addition, you should give `uucp` logins and pseudo user logins, like `who`, `tty`, and `ttytype`, low UIDs so they fall at the beginning of the `passwd` file.

As with user (login) names, you should adopt a scheme to assign unique UIDs. Some companies assign unique employee numbers, and administrators add a number to the employee number to create a unique UID number for each employee.

To minimize security risks, you should avoid reusing the UIDs from deleted accounts. If you must reuse a UID, “wipe the slate clean” so the new user is not affected by attributes set for a former user. For example, a former user might have been denied access to a printer by being included in a printer deny list, but that attribute might not be appropriate for the new user.

Using Large User IDs and Group IDs

UIDs and GIDs can be assigned up to the maximum value of a signed integer, or 2147483647.

However, UIDs and GIDs over 60000 do not have full functionality and are incompatible with many Solaris features, so avoid using UIDs or GIDs over 60000.

The following table describes interoperability issues with Solaris products and previous Solaris releases.

TABLE 4–2 Interoperability Issues for UIDs or GIDs Over 60000

Category	Product or Command	Issues or Cautions
NFS™ Interoperability	SunOS™ 4.0 NFS software and compatible releases	NFS server and client code truncates large UIDs and GIDs to 16 bits. This situation can create security problems if systems running SunOS 4.0 and compatible releases are used in an environment where large UIDs and GIDs are being used. Systems running SunOS 4.0 and compatible releases require a patch to avoid this problem.
Name Service Interoperability	NIS name service and file-based name service	Users with UIDs greater than 60000 can log in or use the `su` command on systems running the Solaris 2.5 and compatible releases, but their UIDs and GIDs will be set to 60001 (nobody).
	NIS+ name service	Users with UIDs greater than 60000 are denied access on systems running Solaris 2.5 and compatible releases and the NIS+ name service.

TABLE 4–3 Large UID or GID Limitation Summary

UID or GID	Limitations
60003 or greater	■ Users in this category logging into systems running Solaris 2.5 and compatible releases and the NIS or files name service get a UID and GID of `nobody`.

TABLE 4–3 Large UID or GID Limitation Summary *(Continued)*

UID or GID	Limitations
65535 or greater	■ Systems running Solaris 2.5 and compatible releases with the NFS version 2 software see UIDs in this category truncated to 16 bits, creating possible security problems. ■ Users in this category using the `cpio` command with the default archive format to copy a file see an error message for each file. And, the UIDs and GIDs are set to `nobody` in the archive. ■ SPARC based systems: Users in this category running SunOS 4.0 and compatible applications see `EOVERFLOW` returns from some system calls, and their UIDs and GIDs are mapped to `nobody`. ■ x86 based systems: Users in this category running SVR3-compatible applications will probably see `EOVERFLOW` return codes from system calls. ■ x86 based systems: If users in this category attempt to create a file or directory on a mounted System V file system, the System V file system returns an `EOVERFLOW` error.
100000 or greater	■ The `ps -l` command displays a maximum five-digit UID so the printed column won't be aligned when they include a UID or GID larger than 99999.
262144 or greater	■ Users in this category using the `cpio` command with the `-H odc` format or the `pax -x cpio` command to copy files see an error message returned for each file. And, the UIDs and GIDs are set to `nobody` in the archive.
1000000 or greater	■ Users in this category using the `ar` command have their UIDs and GIDs set to `nobody` in the archive.
2097152 or greater	■ Users in this category using the `tar` command, the `cpio -H ustar` command, or the `pax -x tar` command have their UIDs and GIDs set to `nobody`.

Passwords

Although user names are publicly known, passwords must be kept secret and known only to users. Each user account should be assigned a password, which is a combination of six to eight letters, numbers, or special characters. You can set a user's password when you create the user account and have the user change it when logging in to a system for the first time.

To make your computer systems more secure, ask users to change their passwords periodically. For a high level of security, you should require users to change their passwords every six weeks. Once every three months is adequate for lower levels of security. System administration logins (such as root and sys) should be changed monthly, or whenever a person who knows the root password leaves the company or is reassigned.

Many breaches of computer security involve guessing a legitimate user's password. You should make sure that users avoid using proper nouns, names, login names, and other passwords that a person might guess just by knowing something about the user.

Good choices for passwords include the following:

- Phrases (`beammeup`)
- Nonsense words made up of the first letters of every word in a phrase. For example, `swotrb` for SomeWhere Over The RainBow.
- Words with numbers or symbols substituted for letters. For example, `sn00py` for snoopy.

Do not use these choices for passwords:

- Your name, forwards, backwards, or jumbled
- Names of family members or pets
- Car license numbers
- Telephone numbers
- Social Security numbers
- Employee numbers
- Names related to a hobby or interest
- Seasonal themes, such as Santa in December
- Any word in the dictionary

Password Aging

If you are using NIS+ or the `/etc` files to store user account information, you can set up password aging on a user's password. Starting in the Solaris 9 Update 2 release, password aging is also supported in the LDAP directory service.

Password aging enables you to force users to change their passwords periodically or to prevent a user from changing a password before a specified interval. If you want to prevent an intruder from gaining undetected access to the system by using an old and inactive account, you can also set a password expiration date when the account becomes disabled. You can set password aging attributes with the `passwd` command or the Solaris Management Console's Users Tool.

Home Directories

The home directory is the portion of a file system allocated to a user for storing private files. The amount of space you allocate for a home directory depends on the kinds of files the user creates, large or small, and the number of files created.

A home directory can be located either on the user's local system or on a remote file server. In either case, by convention the home directory should be created as `/export/home/`*username*. For a large site, you should store home directories on a server. Use a separate file system for each `/export/home`*n* directory to facilitate backing up and restoring home directories. For example, `/export/home1`, `/export/home2`.

Regardless of where their home directory is located, users usually access their home directories through a mount point named `/home/`*username*. When AutoFS is used to mount home directories, you are not permitted to create any directories under the `/home` mount point on any system. The system recognizes the special status of `/home` when AutoFS is active. For more information about automounting home directories, see "Task Overview for Autofs Administration" in *System Administration Guide: Resource Management and Network Services*.

To use the home directory anywhere on the network, you should always refer to the home directory as `$HOME`, not as `/export/home/`*username*. The latter is machine-specific. In addition, any symbolic links created in a user's home directory should use relative paths (for example, `../../../x/y/x`), so the links will be valid no matter where the home directory is mounted.

User's Work Environment

Besides having a home directory to create and store files, users need an environment that gives them access to the tools and resources they need to do their work. When a user logs in to a system, the user's work environment is determined by initialization files that are defined by the user's startup shell, such as the C, Korn, or Bourne shell.

A good strategy for managing the user's work environment is to provide customized user initialization files, such as `.login`, `.cshrc`, `.profile`, in the user's home directory. For detailed information about customizing user initialization files for users, see "Customizing a User's Work Environment" on page 88. After you create the customized user initialization files, you can add them to a user's home directory when you create a new user account.

A recommended one-time task is to set up *skeleton* directories on a server. You can use the same server where the user's home directories are stored. The skeleton directories enable you to store customized user initialization files for different types of users.

Note – Do not use system initialization files, such as `/etc/profile` or `/etc/.login`, to manage a user's work environment, because they reside locally on systems and are not centrally administered. For example, if AutoFS is used to mount the user's home directory from any system on the network, you would have to modify the system initialization files on each system to ensure a consistent environment when a user moved from system to system.

Another way to customize user accounts is through role-based access control. See "Role-Based Access Control (Overview)" in *System Administration Guide: Security Services* for more information.

Guidelines for Managing Groups

A *group* is a collection of users who can share files and other system resources. For example, a set of users that are working on the same project could be formed into a group. A group is traditionally known as a UNIX group.

Each group must have a name, a group identification (GID) number, and a list of user names that belong to the group. A GID identifies the group internally to the system. The two types of groups that a user can belong to are:

- Primary group – Specifies a group that the operating system assigns to files created by the user. Each user must belong to a primary group.
- Secondary groups – Specifies one or more groups to which a user also belongs. Users can belong to up to 15 secondary groups.

Sometimes a user's secondary group is not important. For example, ownership of files reflect the primary group, not any secondary groups. Other applications, however, might rely on a user's secondary memberships. For example, a user has to be a member of the sysadmin group (group 14) to use the Admintool software, but it doesn't matter if group 14 is his or her current primary group.

The `groups` command lists the groups that a user belongs to. A user can have only one primary group at a time. However, a user can temporarily change the user's primary group, with the `newgrp` command, to any other group in which the user is a member.

When adding a user account, you must assign a primary group for a user or accept the default group, `staff` (group 10). The primary group should already exist. If the primary group does not exist, specify the group by a GID number. User names are not added to primary groups. If user names were, the list might become too long. Before you can assign users to a new secondary group, you must create the group and assign it a GID number.

Groups can be local to a system or can be managed through a name service. To simplify group administration, you should use a name service like NIS or a directory service like LDAP, which enables you to centrally manage group memberships.

Tools for Managing User Accounts and Groups

The following table lists the recommended tools for managing users and groups. These tools are all included in the Solaris Management Console suite of tools. For information about starting and using the Solaris Management Console, see Chapter 2.

TABLE 4–4 Tools for Managing Users and Groups

Solaris Management Tool	Is Used To	Task Information
Users	Manage users.	Solaris Management Console Online Help
User Templates	Create a set of attributes for a specific kind of user like students, engineers, or instructors.	Solaris Management Console Online Help
Rights	Manage RBAC rights.	Solaris Management Console Online Help
Administrative Roles	Manage RBAC administrative roles.	Solaris Management Console Online Help
Groups	Manage group information.	Solaris Management Console Online Help
Projects	Manage project information.	Solaris Management Console Online Help
Mailing Lists	Manage mailing lists.	Solaris Management Console Online Help

For information on the Solaris management commands that can be used to manage user accounts and groups if you are not using the Solaris Management Console, see Table 1–6. These commands provide the same functionality as the Solaris management tools, including authentication and name service support.

What You Can Do With Solaris User Management Tools

The Solaris user management tools enable you to manage user accounts on a local system or in a name service environment.

This table describes the tasks you can do with Users Tool's User Accounts feature.

TABLE 4–5 User Account Management Tasks

Task	Description	Background Information
Add a user	You can add a user to the local system or name service.	"What Are User Accounts and Groups?" on page 69 and "Guidelines for Managing User Accounts" on page 70
Create a user Template	You can create a template of pre-defined user attributes for creating users of the same group, such a users, contractors, or engineers.	Same as above
Add a user with a user template	You can add a user with a template so that user attributes are pre-defined.	Same as above
Clone a user template	Clone a user template if you would like to use a similar set of pre-defined user attributes. Then, change only some of the attributes as needed.	Same as above
Set up user properties	You can set up user properties in advance of adding users such as whether a user template is used when adding a user and whether the home directory or mail box is deleted by default when removing a user.	Same as above

TABLE 4–5 User Account Management Tasks *(Continued)*

Task	Description	Background Information
Add multiple users	You can add multiple users to the local system or name service by specifying a text file, typing each name, or automatically generating a series of user names.	Same as above
View or change user properties	You can view or change user properties like login shell, password, or password options.	Same as above
Assign rights to users	You can assign rights to users that will allow them to perform specific administration tasks.	Same as above
Remove a user	You can remove the user from the local system or the name service and optionally specify whether the user's home directory or mail is removed. The user is also removed from any groups or roles.	Same as above

TABLE 4–6 User Rights Management Tasks

Task	Description	Background Information
Grant a right	You can grant a user a right to run a specific command or application that was previously only available to an administrator.	"RBAC Rights Profiles" in *System Administration Guide: Security Services*
View or change existing rights Properties	You can view or change existing rights.	Same as above
Add an authorization	You can add an authorization, which is a discrete right granted to a role or a user.	"RBAC Authorizations" in *System Administration Guide: Security Services*
View or change an authorization	You can view or change existing authorizations.	Same as above

TABLE 4–7 User Role Management Tasks

Task	Description	Background Information
Add an administrative role	You can add a role that someone would use to perform a specific administrative task.	"RBAC Roles" in *System Administration Guide: Security Services*

TABLE 4–7 User Role Management Tasks *(Continued)*

Task	Description	Background Information
Assign rights to an administrative role	You can assign specific rights to a role that enable someone to perform a task.	Same as above
Change an administrative role	You can add or remove rights from a role.	Same as above

TABLE 4–8 Group Management Tasks

Task	Description	
Add a group	Add a group to the local system or name service so that the group name is available before you add the user.	"Guidelines for Managing Groups" on page 76
Add a user to a group	Add a user to a group if the user needs access to group-owned files.	Same as above
Remove a user from a group	You can remove a user from a group if the user no longer requires group file access.	Same as above

TABLE 4–9 Project Management Tasks

Task	Description	Background Information
Create or clone a project	You can create a new project or clone an existing project if it has attributes similar to what you need for the new project.	Solaris Management Console online help
Modify or view project attributes	You can view or change existing project attributes.	Solaris Management Console online help
Delete a project	You can remove a project if it is no longer used.	Solaris Management Console online help

TABLE 4–10 Mailing List Management Tasks

Task	Description	Background Information
Create a mailing list	You can create a mailing list, which is a list of names for sending email messages.	Solaris Management Console online help

TABLE 4–10 Mailing List Management Tasks *(Continued)*

Task	Description	Background Information
Change a mailing list name	You can make changes to the mailing list after it is created.	Solaris Management Console online help
Remove a mailing list	You can remove a mailing list if it is no longer used.	Solaris Management Console online help

Managing Home Directories With the Solaris Management Console

Keep the following in mind when using the Solaris Management Console tools to manage user home directories:

- If you use the Users Tool's Add User Wizard to add a user account and you specify the user's home directory as `/export/home/`*username*, the home directory is automatically setup to be automounted, and the following entry is added to the `passwd` file:

  ```
  /home/username
  ```

- The only way you can use Users Tool to set up a user account that does not automount the home directory is to set up a user account template that disables this feature. Then, you can add users with this template. There is no way to disable this feature with the Add User Wizard.
- You can use the `smuser add` command with the `-x autohome=N` option to add a user without automounting the user's home directory. However, there is no option to the `smuser delete` command to remove the home directory after the user is added. You would have to remove the user and the user's home directory with the Users Tool

Modify User Accounts

Unless you define a user name or UID number that conflicts with an existing one, you should never need to modify a user account's login name or UID number. Use the following steps if two user accounts have duplicate user names or UID numbers:

- If two user accounts have duplicate UID numbers, use the Users Tool to remove one account and re-add it with a different UID number. You cannot use the Users Tool to modify a UID number of an existing user account.
- If two user accounts have duplicate user names, use the Users Tool to modify one of the accounts and change the user name.

If you do use the Users Tool to change a user name, the home directory's ownership is changed, if a home directory exists for the user.

One part of a user account that you can change is a user's group memberships. Select Properties from Users Tool's Action menu to add or delete a user's secondary groups. Alternatively, you can use the Groups Tool to directly modify a group's member list.

You can also modify the following parts of a user account:

- Description (comment)
- Login shell
- Passwords and password options
- Home directory and home directory access
- Rights and roles

Delete User Accounts

When you delete a user account with the Users Tool, the software deletes the entries in the `passwd` and `group` files. In addition, you can delete the files in the user's home directory and mail directory.

Add Customized User Initialization Files

Although you cannot create customized user initialization files with the Users Tool, you can populate a user's home directory with user initialization files located in a specified "skeleton" directory. You can do this by creating a user template with the User Templates tool and specifying a skeleton directory from which to copy user initialization files.

You can customize the user initialization templates in the `/etc/skel` directory and then copy them to users' home directories.

Administer Passwords

You can use Users Tool for password administration, which includes the following capabilities:

- Specifying a normal password for a user account
- Enabling users to create their own passwords during their first login
- Disabling or locking a user account
- Specifying expiration dates and password aging information.

Note – Password aging is not supported by the NIS name service.

Disable User Accounts

Occasionally, you might need to temporarily or permanently disable a login account. Disabling or locking a user account means that an invalid password, `*LK*`, is assigned to the user account, preventing future logins.

The easiest way to disable a user account is to lock the password for an account with Users Tool. You can also enter an expiration date in the account availability section of the User Properties screen to set a limit on how long the account is active.

Other ways to disable a user account is to set up password aging or to change the user's password.

Where User Account and Group Information Is Stored

Depending on your site policy, you can store user account and group information in a name service or a local system's `/etc` files. In the NIS+ name service, information is stored in tables, in the NIS name service, information is stored in maps, and in the LDAP directory service, information is stored in indexed database files.

Note – To avoid confusion, the location of the user account and group information is generically referred to as a *file* rather than as a *database*, *table* or *map*.

Most of the user account information is stored in the `passwd` file. However, password encryption and password aging is stored in the `passwd` file when using NIS or NIS+ and in the `/etc/shadow` file when using `/etc` files. Password aging is not available when using NIS.

Group information is stored in the `group` file.

Fields in the `passwd` File

The fields in the `passwd` file are separated by colons and contain the following information:

username:*password*:*uid*:*gid*:*comment*:*home-directory*:*login-shell*

For example:

```
kryten:x:101:100:Kryten Series 4000 Mechanoid:/export/home/kryten:/bin/csh
```

The following table describes the `passwd` file fields.

TABLE 4–11 Fields in the `passwd` File

Field Name	Description
username	Contains the user or login name. User names should be unique and consist of 1-8 letters (A-Z, a-z) and numerals (0-9). The first character must be a letter, and at least one character must be a lowercase letter.
password	Contains an `x`, a placeholder for the encrypted password. The encrypted password is stored in the `shadow` file.
uid	Contains a user identification (UID) number that identifies the user to the system. UID numbers for regular users should range from 100 to 60000. All UID numbers should be unique.
gid	Contains a group identification (GID) number that identifies the user's primary group. Each GID number must be a whole number between 0 and 60002. 60001 and 60002 are assigned to `nobody` and `noaccess`. 65534 is assigned to `nobody4`.
comment	Usually contains the full name of the user. This field is informational only. It is sometimes called the GECOS field because it was originally used to hold the login information needed to submit batch jobs to a mainframe running GECOS (General Electric Computer Operating System) from UNIX systems at Bell Labs.
home-directory	Contains the user's home directory path name.
login-shell	Contains the user's default login shell, such as `/bin/sh`, `/bin/csh` or `/bin/ksh`. Table 4–18 contains a description of shell features.

Default `passwd` File

The default Solaris `passwd` file contains entries for standard daemons, processes usually started at boot time to perform some system-wide task, such as printing, network administration, and port monitoring.

```
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
adm:x:4:4:Admin:/var/adm:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
```

```
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x Nobody:/:
```

TABLE 4–12 Default `passwd` File Entries

User Name	User ID	Description
`root`	`0`	Superuser account.
`daemon`	`1`	Umbrella system daemon associated with routine system tasks.
`bin`	`2`	Administrative daemon associated with running system binaries to perform some routine system task.
`sys`	`3`	Administrative daemon associated with system logging or updating files in temporary directories.
`adm`	`4`	Administrative daemon associated with system logging.
`lp`	`71`	Line printer daemon.
`uucp`	`5`	Daemon associated with `uucp` functions.
`nuucp`	`6`	Daemon associated with `uucp` functions.
`smmsp`	`25`	Sendmail message submission program daemon.
`listen`	`37`	Network listener daemon.
`nobody`	`60001`	Assigned to users or software processes that do not need nor should have any special permissions.
`noaccess`	`60002`	Assigned to a user or a process that needs access to a system through some application but without actually logging in.
`nobody4`	`65534`	SunOS 4.0 or 4.1 version of the `nobody` user account.

Fields in the `shadow` File

The fields in the `shadow` file are separated by colons and contain the following information:

username : *password* : *lastchg* : *min* : *max* : *warn* : *inactive* : *expire*

For example:

```
rimmer:86Kg/MNT/dGu.:8882:0::5:20:8978
```

The following table describes the `shadow` file fields.

TABLE 4–13 Fields in the `shadow` File

Field Name	Description
username	Contains the user or login name.
password	Might contain the following entries: a 13-character encrypted user password; the string `*LK*`, which indicates an inaccessible account; or the string `NP`, which indicates no password for the account.
lastchg	Indicates the number of days between January 1, 1970, and the last password modification date.
min	Contains the minimum number of days required between password changes.
max	Contains the maximum number of days the password is valid before the user is prompted to specify a new password.
inactive	Contains the number of days a user account can be inactive before being locked.
expire	Contains the absolute date when the user account expires. Past this date, the user cannot log in to the system.

Fields in the `group` File

The fields in the `group` file are separated by colons and contain the following information:

group-name:*group-password*:*gid*:*user-list*

For example:

```
bin::2:root,bin,daemon
```

The following table describes the `group` file fields.

TABLE 4–14 Fields in the `group` File

Field Name	Description
group-name	Contains the name assigned to the group. For example, members of the chemistry department in a university might be called `chem`. Group names can have a maximum of eight characters.

TABLE 4–14 Fields in the `group` File *(Continued)*

Field Name	Description
group-password	Usually contains an asterisk or is empty. The *group-password* field is a relic of earlier versions of UNIX. If a group has a password, the `newgrp` command prompts users to enter the password. However, no utility exists to set the password.
gid	Contains the group's GID number. It must be unique on the local system, and should be unique across the entire organization. Each GID number must be a whole number between 0 and 60002. Numbers under 100 are reserved for system default group accounts. User defined groups can range from 100 to 60000. 60001 and 60002 are reserved and assigned to `nobody` and `noaccess`, respectively.
user-list	Contains a comma-separated list of user names, representing the user's secondary group memberships. Each user can belong to a maximum of 15 secondary groups.

Default `group` file

The default Solaris `group` file contains the following system groups that support some system-wide task, such as printing, network administration, and electronic mail. Many of these groups having corresponding entries in the `passwd` file.

```
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
smmsp::25:smmsp
sysadmin::14:root
nobody::60001:
noaccess::60002:
nogroup::65534:
```

TABLE 4–15 Default `group` File Entries

Group Name	Group ID	Description
`root`	`0`	Superuser group.
`other`	`1`	Optional group.

TABLE 4–15 Default `group` File Entries *(Continued)*

Group Name	Group ID	Description
`bin`	`2`	Administrative group associated with running system binaries.
`sys`	`3`	Administrative group associated with system logging or temporary directories.
`adm`	`4`	Administrative group associated with system logging.
`uucp`	`5`	Group associated with `uucp` functions.
`mail`	`6`	Electronic mail group.
`tty`	`7`	Group associated with tty devices.
`lp`	`8`	Line printer group.
`nuucp`	`9`	Group associated with `uucp` functions.
`staff`	`10`	General administrative group.
`daemon`	`12`	Group associated with routine system tasks.
`sysadmin`	`14`	Administrative group associated with Admintool and Solstice AdminSuite tools.
`smmsp`	25	Sendmail message submission program daemon.
`nobody`	`60001`	Group assigned to users or software processes that do not need nor should have any special permissions.
`noaccess`	`60002`	Group assigned to a user or a process that needs access to a system through some application but without actually logging in.
`nogroup`	`65534`	Group assigned to a user who not a member of a known group.

Customizing a User's Work Environment

Part of setting up a user's home directory is providing user initialization files for the user's login shell. A *user initialization file* is a shell script that sets up a work environment for a user after the user logs in to a system. Basically, you can perform any task in a user initialization file that you can do in a shell script. However, its primary job is to define the characteristics of a user's work environment, such as a user's search path, environment variables, and windowing environment. Each login shell has its own user initialization file or files, which are listed in the following table.

TABLE 4–16 User Initialization Files for Bourne, C, and Korn Shells

Shell	User Initialization File	Purpose
Bourne	`$HOME/.profile`	Defines user's environment at login
C	`$HOME/.cshrc`	Defines user's environment for all C shells and is invoked after login shell
	`$HOME/.login`	Defines user's environment at login
Korn	`$HOME/.profile`	Defines user's environment at login
	`$HOME/$ENV`	Defines user's environment at login in the file and is specified by the Korn shell's `ENV` environment variable

The Solaris environment provides default user initialization files for each shell in the `/etc/skel` directory on each system, as shown in the following table.

TABLE 4–17 Default User Initialization Files

Shell	Default File
C	`/etc/skel/local.login`
	`/etc/skel/local.cshrc`
Bourne or Korn	`/etc/skel/local.profile`

You can use these files as a starting point and modify them to create a standard set of files that provide the work environment common to all users. Or, you can modify them to provide the working environment for different types of users. For step-by-step instructions on how to create sets of user initialization files for different types of users, see "How to Customize User Initialization Files" on page 103.

When you use the Users Tool to create a new user account and select the create home directory option, the following files are created, depending on which login shell is selected:

Shell	Files Created
C	The `/etc/skel/local.cshrc` and the `/etc/skel/local.login` files are copied into the user's home directory and are renamed `.cshrc` and `.login`.

Shell	Files Created
Bourne and Korn	The `/etc/skel/local.profile` file is copied into the user's home directory and renamed `.profile`.

If you use the `useradd` command to add a new user account and specify the `/etc/skel` directory by using the `-k` and `-m` options, all three `/etc/skel/local*` and `/etc/skel/.profile` files are copied into the user's home directory. At this point, you will need to rename them to whatever is appropriate for the user's login shell.

Using Site Initialization Files

The user initialization files can be customized by both the administrator and the user. This important feature can be accomplished with centrally located and globally distributed user initialization files, called site initialization files. Site initialization files enable you to continually introduce new functionality to the user's work environment, while enabling the user to customize the user's initialization file.

When you reference a site initialization file in a user initialization file, all updates to the site initialization file are automatically reflected when the user logs in to the system or when a user starts a new shell. Site initialization files are designed for you to distribute site-wide changes to users' work environments that you did not anticipate when you added the users.

Any customization that can be done in a user initialization file can be done in a site initialization file. These files typically reside on a server, or set of servers, and appear as the first statement in a user initialization file. Also, each site initialization file must be the same type of shell script as the user initialization file that references it.

To reference a site initialization file in a C-shell user initialization file, place a line similar to the following at the beginning of the user initialization file:

```
source /net/machine-name/export/site-files/site-init-file
```

To reference a site initialization file in a Bourne- or Korn-shell user initialization file, place a line similar to the following at the beginning of the user initialization file:

```
. /net/machine-name/export/site-files/site-init-file
```

Avoid Local System References

You should not add specific references to the local system in the user's initialization file. You want the instructions in a user initialization file to be valid regardless of the system to which the user logs in. For example:

- To make a user's home directory available anywhere on the network, always refer to the home directory with the variable `$HOME`. For example, use `$HOME/bin` instead of `/export/home/`*username*`/bin`. `$HOME` works when the user logs in to another system and the home directories are automounted.
- To access files on a local disk, use global path names, like `/net/`*system-name/directory-name*. Any directory referenced by `/net/`*system-name* can be mounted automatically on any system on which the user logs in, assuming the system is running AutoFS.

Shell Features

The following table lists basic shell features that each shell provides, which can help you determine what you can and can't do when creating user initialization files for each shell.

TABLE 4–18 Basic Features of Bourne, C, and Korn Shells

Feature	Bourne	C	Korn
Known as the standard shell in UNIX	Yes	No	No
Compatible syntax with Bourne shell	-	No	Yes
Job control	Yes	Yes	Yes
History list	No	Yes	Yes
Command-line editing	No	Yes	Yes
Aliases	No	Yes	Yes
Single-character abbreviation for login directory	No	Yes	Yes
Protection from overwriting (`noclobber`)	No	Yes	Yes
Setting to ignore Control-d (`ignoreeof`)	No	Yes	Yes
Enhanced `cd`	No	Yes	Yes
Initialization file separate from `.profile`	No	Yes	Yes
Logout file	No	Yes	No

Shell Environment

A shell maintains an environment that includes a set of variables defined by the `login` program, the system initialization file, and the user initialization files. In addition, some variables are defined by default. A shell can have two types of variables:

- Environment variables – Variables that are exported to all processes spawned by the shell. Their settings can be seen with the `env` command. A subset of environment variables, like `PATH`, affects the behavior of the shell itself.
- Shell (local) variables – Variables that affect only the current shell. In the C shell, a set of these shell variables have a special relationship to a corresponding set of environment variables. These shell variables are `user`, `term`, `home`, and `path`. The value of the environment variable counterpart is initially used to set the shell variable.

In the C shell, you use the lowercase names with the `set` command to set shell variables and use uppercase names with the `setenv` command to set environment variables. If you set a shell variable, the shell sets the corresponding environment variable and vice versa. For example, if you update the `path` shell variable with a new path, the shell also updates the `PATH` environment variable with the new path.

In the Bourne and Korn shells, you can use the uppercase variable name equal to some value to set both shell and environment variables. You also have to use the `export` command to activate the variables for any subsequently executed commands.

For all shells, you generally refer to shell and environment variables by their uppercase names.

In a user initialization file, you can customize a user's shell environment by changing the values of the predefined variables or by specifying additional variables. The following table shows how to set environment variables in a user initialization file.

TABLE 4–19 Setting Environment Variables in a User Initialization File

Set a User's Environment Variables for The Shell Type	Line to Add to the User Initialization File
C shell	`setenv` *VARIABLE value* Example: `setenv MAIL /var/mail/ripley`
Bourne or Korn shell	*VARIABLE*=*value*`; export` *VARIABLE* Example: `MAIL=/var/mail/ripley;export MAIL`

The following table describes environment and shell variables that you might want to customize in a user initialization file. For more information about variables that are used by the different shells, see sh(1), ksh(1), or csh(1).

TABLE 4–20 Shell and Environment Variable Descriptions

Variable	Description
CDPATH, or cdpath in the C shell	Sets a variable used by the cd command. If the target directory of the cd command is specified as a relative path name, the cd command first looks for the target directory in the current directory (“.”). If the target is not found, the path names listed in the CDPATH variable are searched consecutively until the target directory is found and the directory change is completed. If the target directory is not found, the current working directory is left unmodified. For example, the CDPATH variable is set to /home/jean, and two directories exist under /home/jean, bin and rje. If you are in the /home/jean/bin directory and type cd rje, you change directories to /home/jean/rje, even though you do not specify a full path.
history	Sets history for the C shell.
HOME, or home in the C shell	Sets the path to the user’s home directory.
LANG	Sets the locale.
LOGNAME	Defines the name of the user currently logged in. The default value of LOGNAME is set automatically by the login program to the user name specified in the passwd file. You should only need to refer to, not reset, this variable.
LPDEST	Sets the user’s default printer.
MAIL	Sets the path to the user’s mailbox.
MANPATH	Sets the hierarchies of man pages available.
PATH, or path in the C shell	Specifies, in order, the directories that the shell searches to find the program to run when the user types a command. If the directory is not in the search path, users must type the complete path name of a command. The default PATH is automatically defined and set as specified in .profile (Bourne or Korn shell) or .cshrc (C shell) as part of the login process. The order of the search path is important. When identical commands exist in different locations, the first command found with that name is used. For example, suppose that PATH is defined in Bourne and Korn shell syntax as PATH=/bin:/usr/bin:/usr/sbin:$HOME/bin and a file named sample resides in both /usr/bin and /home/jean/bin. If the user types the command sample without specifying its full path name, the version found in /usr/bin is used.

TABLE 4–20 Shell and Environment Variable Descriptions *(Continued)*

Variable	Description
`prompt`	Defines the shell prompt for the C shell.
`PS1`	Defines the shell prompt for the Bourne or Korn shell.
`SHELL`, or shell in the C shell	Sets the default shell used by `make`, `vi`, and other tools.
`TERMINFO`	Specifies the path name for an unsupported terminal that has been added to the `terminfo` file. Use the `TERMINFO` variable in `/etc/profile` or `/etc/.login`. When the `TERMINFO` environment variable is set, the system first checks the `TERMINFO` path defined by the user. If it does not find a definition for a terminal in the `TERMINFO` directory defined by the user, it searches the default directory, `/usr/share/lib/terminfo`, for a definition. If the system does not find a definition in either location, the terminal is identified as "dumb."
`TERM`, or term in the C shell	Defines the terminal. This variable should be reset in `/etc/profile` or `/etc/.login`. When the user invokes an editor, the system looks for a file with the same name as the definition of this environment variable. The system searches the directory referenced by `TERMINFO` to determine the terminal characteristics.
`TZ`	Sets the time zone, which is used to display dates, for example, in the `ls -l` command. If `TZ` is not set in the user's environment, the system setting is used. Otherwise, Greenwich Mean Time is used.

The `PATH` Variable

When the user executes a command by using the full path, the shell uses that path to find the command. However, when users specify only a command name, the shell searches the directories for the command in the order specified by the `PATH` variable. If the command is found in one of the directories, the shell executes the command.

A default path is set by the system, but most users modify it to add other command directories. Many user problems related to setting up the environment and accessing the right version of a command or a tool can be traced to incorrectly defined paths.

Setting Path Guidelines

Here are some guidelines for setting up efficient `PATH` variables:

- If security is not a concern, put the current working directory (.) first in the path. However, including the current working directory in the path poses a security risk that you might want to avoid, especially for superuser.

- Keep the search path as short as possible. The shell searches each directory in the path. If a command is not found, long searches can slow down system performance.
- The search path is read from left to right, so you should put directories for commonly used commands at the beginning of the path.
- Make sure directories are not duplicated in the path.
- Avoid searching large directories, if possible. Put large directories at the end of the path.
- Put local directories before NFS™ mounted directories to lessen the chance of "hanging" when the NFS server does not respond and to reduce unnecessary network traffic.

Examples—Setting a User's Default Path

The following examples show how to set a user's default path to include the home directory and other NFS mounted directories. The current working directory is specified first in the path. In a C-shell user initialization file, you would add the following:

```
set path=(. /usr/bin $HOME/bin /net/glrr/files1/bin)
```

In a Bourne- or Korn-shell user initialization file, you would add the following:

```
PATH=.:/usr/bin:/$HOME/bin:/net/glrr/files1/bin
export PATH
```

Locale Variables

The `LANG` and `LC` environment variables specify the locale-specific conversions and conventions for the shell, like time zones, collation orders, and formats of dates, time, currency, and numbers. In addition, you can use the `stty` command in a user initialization file to set whether the system will support multibyte characters.

`LANG` sets all possible conversions and conventions for the given locale. If you have special needs, you can set various aspects of localization separately through these `LC` variables: `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, `LC_NUMERIC`, `LC_MONETARY`, and `LC_TIME`.

The following table describes some of the values for the `LANG` and `LC` environment variables.

TABLE 4–21 Values for LANG and LC Variables

Value	Locale
de	German
fr	French
iso_8859_1	English and European
it	Italian
japanese	Japanese
korean	Korean
sv	Swedish
tchinese	Taiwanese

Examples—Setting the Locale Using the LANG Variables

The following examples show how to set the locale by using the LANG environment variables. In a C-shell user initialization file, you would add the following:

```
setenv LANG DE
```

In a Bourne- or Korn-shell user initialization file, you would add the following:

```
LANG=DE; export LANG
```

Default File Permissions (umask)

When you create a file or directory, the default file permissions assigned to the file or directory are controlled by the *user mask*. The user mask is set by the umask command in a user initialization file. You can display the current value of the user mask by typing umask and pressing Return.

The user mask contains the following octal values:

- The first digit sets permissions for the user
- The second sets permissions for group
- The third sets permissions for other, also referred to as "world"

Note that if the first digit is zero, it is not displayed. For example, if umask is set to 022, 22 is displayed.

To determine the umask value you want to set, subtract the value of the permissions you want from 666 (for a file) or 777 (for a directory). The remainder is the value to use with the umask command. For example, suppose you want to change the default mode for files to 644 (rw-r--r--). The difference between 666 and 644 is 022, which is the value you would use as an argument to the umask command.

You can also determine the `umask` value you want to set by using the following table, which shows the file and directory permissions that are created for each of the octal values of `umask`.

TABLE 4–22 Permissions for `umask` Values

`umask` Octal Value	File Permissions	Directory Permissions
0	`rw-`	`rwx`
1	`rw-`	`rw-`
2	`r--`	`r-x`
3	`r--`	`r--`
4	`-w-`	`-wx`
5	`-w-`	`-w-`
6	`--x`	`--x`
7	`---` (none)	`---` (none)

The following line in a user initialization file sets the default file permissions to `rw-rw-rw-`.

```
umask 000
```

Examples of User and Site Initialization Files

The following sections provide examples of user and site initialization files that you can use to start customizing your own initialization files. Many of the examples use system names and paths that you need to change for your particular site.

Example—`.profile` File

```
1 PATH=$PATH:$HOME/bin:/usr/local/bin:/usr/ccs/bin:.
2 MAIL=/var/mail/$LOGNAME
3 NNTPSERVER=server1
4 MANPATH=/usr/share/man:/usr/local/man
5 PRINTER=printer1
6 umask 022
7 export PATH MAIL NNTPSERVER MANPATH PRINTER
```

1. Defines the user's shell search path.
2. Defines the path to the user's mail file.
3. Defines the user's Usenet news server.
4. Defines the user's search path for man pages.
5. Defines the user's default printer.

6. Sets the user's default file creation permissions.
7. Sets the listed environment variables.

Example—`.cshrc` File

```
1 set path=($PATH $HOME/bin /usr/local/bin /usr/ccs/bin)
2 setenv MAIL /var/mail/$LOGNAME
3 setenv NNTPSERVER server1
4 setenv PRINTER printer1
5 alias h history
6 umask 022
7 source /net/server2/site-init-files/site.login
```

1. Defines the user's shell search path.
2. Defines the path to the user's mail file.
3. Defines the user's Usenet news server.
4. Defines the user's default printer.
5. Creates an alias for the `history` command. The user will need to type only `h` to run the `history` command.
6. Sets the user's default file creation permissions.
7. Sources the site initialization file.

Example—Site Initialization File

The following shows an example site initialization file in which a user can choose a particular version of an application.

```
# @(#)site.login
main:
echo "Application Environment Selection"
echo ""
echo "1. Application, Version 1"
echo "2. Application, Version 2"
echo ""
echo -n "Type 1 or 2 and press Return to set your
application environment: "

set choice = $<

if ( $choice !~ [1-2] ) then
goto main
endif

switch ($choice)

case "1":
setenv APPHOME /opt/app-v.1
```

```
breaksw

case "2":
setenv APPHOME /opt/app-v.2
endsw
```

This site initialization file could be referenced in a user's `.cshrc` file (C shell users only) with the following line:

```
source /net/server2/site-init-files/site.login
```

In this line, the site initialization file is named `site.login` and is located on a server named `server2`. This line also assumes that the automounter is running on the user's system.

CHAPTER 5

Managing User Accounts and Groups (Tasks)

This chapter describes how to set up and maintain user accounts and groups.

For information on the procedures associated with setting up and maintaining user accounts and groups, see "Setting Up User Accounts (Task Map)" on page 101 and "Maintaining User Accounts (Task Map)" on page 107.

For background information about managing user accounts and groups, see Chapter 4.

Setting Up User Accounts (Task Map)

Task	Description	For Instructions
1. Start the Solaris Management Console launcher	Start the Solaris Management Console launcher to access the User Accounts and Groups Tools.	"How to Start the Console as Superuser or as a Role" on page 54 or "How to Start the Solaris Management Console in a Name Service Environment" on page 61
2. (Optional) Customize User Initialization Files	You can set up user initialization files (`.cshrc`, `.profile`, `.login`), so you can provide new users with consistent environments.	"How to Customize User Initialization Files" on page 103
3. (Optional) Add a Group	You can add groups to help administer users by using the Groups Tool.	See Solaris Management Console online help

Task	Description	For Instructions
4. (Optional) Set up a User Template	You can create a user template so you don't have to manually add all similar user properties.	See Solaris Management Console online help
5. Add a User	You can add a user account by using the Users Tool.	See Solaris Management Console online help
6. (Optional) Add Rights or a Role to a User	You can add rights or a role to a user so the user can perform a specific command or task.	See Solaris Management Console online help
7. Share the User's Home Directory	You must share the user's home directory so the directory can be remotely mounted from the user's system.	"How to Share a User's Home Directory" on page 104
8. Mount the User's Home Directory	You must mount the user's home directory on the user's system.	"How to Mount a User's Home Directory" on page 105

User Information Data Sheet

You might find it useful to create a form like the one that follows to gather information about users before adding their accounts.

Item	Description
User Name:	
Role Name:	
Profiles or Authorizations:	
User Name:	
UID:	
Primary Group:	
Secondary Groups:	
Comment:	
Default Shell:	
Password Status and Aging:	
Home Directory Server Name:	

Home Directory Path Name:	
Mounting Method:	
Permissions on Home Directory:	
Mail Server:	
Department Name:	
Department Administrator:	
Manager:	
Employee Name:	
Employee Title:	
Employee Status:	
Employee Number:	
Start Date:	
Add to These Mail Aliases:	
Desktop System Name:	

▼ How to Customize User Initialization Files

1. **Become superuser or assume an equivalent role on the system where the users' home directories are created and shared.**

2. **Create a skeleton directory for each type of user.**

   # **mkdir /***shared-dir***/skel/***user-type*

shared-dir	The name of a directory that is available to other systems on the network.
user-type	The name of a directory to store initialization files for a type of user.

3. **Copy the default user initialization files into the directories you created for different types of users.**

   # **cp /etc/skel/local.cshrc /***shared-dir***/skel/***user-type***/.cshrc**
   # **cp /etc/skel/local.login /***shared-dir***/skel/***user-type***/.login**
   # **cp /etc/skel/local.profile /***shared-dir***/skel/***user-type***/.profile**

Note – If the account has profiles assigned to it, then the user has to launch a special version of the shell called a profile shell to use commands (with any security attributes) that are assigned to the profile. There are three profile shells corresponding to the types of shells: `pfsh` (Bourne shell), `pfcsh` (C shell), and `pfksh` (Korn shell).

4. **Edit the user initialization files for each user type and customize them based on your site's needs.**

   For a detailed description on the ways to customize the user initialization files, see "Customizing a User's Work Environment" on page 88.

5. **Set the permissions for the user initialization files.**

   ```
   # chmod 744 /shared-dir/skel/user-type/.*
   ```

6. **Verify that the permissions for the user initialization files are correct.**

   ```
   # ls -la /shared-dir/skel/*
   ```

Example—Customizing User Initialization Files

The following example shows how to customize the C-shell user initialization file in the `/export/skel/enduser` directory designated for a particular type of user. For an example of a `.cshrc` file, see "Example—.cshrc File" on page 98.

```
# mkdir /export/skel/enduser
# cp /etc/skel/local.cshrc /export/skel/enduser/.cshrc

(Edit .cshrc file)
# chmod 744 /export/skel/enduser/.*
```

▼ How to Share a User's Home Directory

1. **Become superuser or assume an equivalent role on the system that contains the home directory.**

2. **Verify that the `mountd` daemon is running.**

   ```
   # ps -ef | grep mountd
   root   176     1  0   May 02 ?        0:19 /usr/lib/nfs/mountd
   ```

   The `/usr/lib/nfs/mountd` line shows whether the `mountd` daemon is running.

3. **If the `mountd` daemon is not running, start it.**

   ```
   # /etc/init.d/nfs.server start
   ```

4. **List the file systems that are shared on the system.**

```
# share
```

5. **Select one of the following based on whether the file system containing the user's home directory is already shared.**

   a. **If the user's home directory is already shared, go to the verification step below.**

   b. **If the user's home directory is not shared, go to Step 6.**

6. **Edit the `/etc/dfs/dfstab` file and add the following line.**

   ```
   share -F nfs /file-system
   ```

   file-system is the file system containing the user's home directory that you need to share. By convention, the file system is `/export/home`.

7. **Share the file systems listed in the `/etc/dfs/dfstab` file.**

   ```
   # shareall -F nfs
   ```

   This command executes all the `share` commands in the `/etc/dfs/dfstab` file, so you do not have to wait to reboot the system.

8. **Verify that a user's home directory is shared, as follows:**

   ```
   # share
   ```

Where to Go From Here

If the user's home directory is not located on the user's system, you have to mount the user's home directory from the system where it is located. For detailed instructions, see "How to Mount a User's Home Directory" on page 105.

Example—Sharing a User's Home Directory

```
# ps -ef | grep mountd
# /etc/init.d/nfs.server start
# share
# vi /etc/dfs/dfstab

(The line share -F nfs /export/home is added.)
# shareall -F nfs
# share
-               /usr/dist               ro   ""
-               /export/home/user-name  rw   ""
```

▼ How to Mount a User's Home Directory

For information on automounting a home directory, see "Task Overview for Autofs Administration" in *System Administration Guide: Resource Management and Network Services*.

1. **Make sure that the user's home directory is shared.**
   For more information, see "How to Share a User's Home Directory" on page 104.
2. **Log in as superuser on the user's system.**
3. **Edit the `/etc/vfstab` file and create an entry for the user's home directory.**

   system-name`:/export/home/`*user-name* `- /export/home/`*user-name* `nfs - yes rw`

system-name	The name of the system where the home directory is located.
`/export/home/`*user-name*	The name of the user's home directory that will be shared. By convention, `/export/home`*user-name* contains user's home directories. However, this could be a different file system.
`-`	Required placeholders in the entry.
`/export/home/`*user-name*	The name of the directory where the user's home directory will be mounted.

   For more information about adding an entry to the `/etc/vfstab` file, see Chapter 40.
4. **Create the mount point for the user's home directory.**

   `# mkdir -p /export/home/`*user-name*
5. **Mount the user's home directory.**

   `# mountall`

   All entries in the current `vfstab` file (whose `mount at boot` fields are set to `yes`) are mounted.
6. **Verify that the home directory is mounted.**

   `# mount | grep` *user-name*

Example—Mounting a User's Home Directory

```
# vi /etc/vfstab

(The line  venus:/export/home/ripley - /export/home/ripley
nfs - yes rw  is added.)
# mkdir -p /export/home/ripley
# mountall
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/onerror=panic/dev=2200000 ...
/proc on /proc read/write/setuid/dev=3840000 on Wed Feb 28 09:49:07 2001
/dev/fd on fd read/write/setuid/dev=3900000 on Wed Feb 28 09:49:10 2001
/etc/mnttab on mnttab read/write/setuid/dev=3a00000 on Wed Feb 28 09:49:12 2001
```

```
/var/run on swap read/write/setuid/dev=1 on Wed Feb 28 09:49:12 2001
/tmp on swap read/write/setuid/dev=2 on Wed Feb 28 09:49:15 2001
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/onerror=panic/dev= ...
/export/home/ripley on venus:/export/home/ripley remote/read/write/setuid/dev=3a8001e ...
```

Maintaining User Accounts (Task Map)

Task	Description	Instructions
Modify a Group	You can modify a group's name or the users in a group by using the Groups Tool.	See Solaris Management Console online help
Delete a Group	You can delete a group if its no longer needed.	See Solaris Management Console online help
Modify a User Account	*Disable a User Account* You can temporarily disable a user account if it will be needed in the future. *Change a User's Password* You might need to change a user's password if the user forgets it. *Change Password Aging* You can force users to change their passwords periodically with User Account tool's Password Options menu.	See Solaris Management Console online help See Solaris Management Console online help See Solaris Management Console online help
Delete a User Account	You can delete a user account if it is no longer needed.	See Solaris Management Console online help

Solaris User Registration

Solaris User Registration is a tool for getting information about new Solaris releases, upgrade offers, and promotions. This graphical user interface (GUI) automatically starts when you first log into your desktop. The GUI lets you register now, later, or never. The registration process also provides Sun with the user's Solaris version, survey type, platform, hardware, and locale.

Accessing Solaris Solve

Completing the Solaris User Registration process provides access to Solaris Solve[SM], an exclusive web site that offers valuable Solaris product information and solutions—all in one convenient location. It provides a quick and easy method for getting the most recent information on what's happening around the latest Solaris release. Solaris Solve also provides a preview to additional Sun contract and service opportunities.

Basically, the steps for completing Solaris User Registration and accessing Solaris Solve are:

1. Fill in the electronic Solaris User Registration profile.
2. Submit the profile by email or print the profile to fax or mail.
3. Create your login ID and password to access the Solaris Solve site.

   Even if you do not access the Solaris Solve site immediately, we recommend that you create your Solaris Solve login ID and password during the Solaris User Registration process. A Solaris Solve login ID and password should contain 6 to 8 alphanumeric characters without spaces and colons.
4. Access the Solaris Solve site.

Note – Solaris User Registration is not invoked if the system administrator or user is logged in as superuser.

If you choose to register, a copy of the completed form is stored in `$HOME/.solregis/uprops`. If you choose to never register and change your mind later, you can start User Registration by:

- Typing `/usr/dt/bin/solregis` at any command line prompt, or
- Clicking the Registration icon in the Application Manager's desktop tools folder (Common Desktop Environment desktop only)

For more information, see `solregis`(1).

Troubleshooting Solaris User Registration Problems

This section provides troubleshooting tips for solving Solaris User Registration problems.

The following table describes problems that may occur when you try to register, and actions required to resolve these conflicts.

TABLE 5–1 Registration Problem Descriptions and Suggested Resolutions

Problem Description	How to Resolve the Problem
The registration form failed to initialize: Web page window displays and requests user see system administrator to resolve problem that prevents registration setup.	Check for missing registration files.
The form could not be emailed: Dialog box displays and requests user see system administrator to resolve problem.	Check to see if email is configured correctly. Also check if CDE is on user's system since it must be present to email completed registration form. Alternatively, users can print the form and fax or mail it.
The form could not be printed: Dialog box displays and requests user to see system administrator to resolve problem.	Check to see if the printer is configured correctly. Alternatively, the user can email form.
The form could not be saved: Dialog box displays and verifies that registration succeeded; however, the registration information cannot be recalled when updating registration in the future.	Check the user's home directory. Required action depends on the system's configuration.
You forgot your Solaris Solve login ID and password.	Send a mail message describing the problem to `SolarisSolve@sun.com` or see "How to Restart Solaris User Registration" on page 109.
You want to restart the registration process.	"How to Restart Solaris User Registration" on page 109.

▼ How to Restart Solaris User Registration

Use the following procedure to restart the Solaris User Registration process.

1. **Change to the `$HOME/.solregis` directory.**

   ```
   % cd $HOME/.solregis
   ```

2. **Remove the `uprops` file.**

   ```
   % rm uprops
   ```

3. **Restart the registration process.**

   ```
   % /usr/dt/bin/solregis &
   ```

▼ How To Disable User Registration

The following table shows how to disable User Registration before and after installing Solaris software. Before disabling Solaris User Registration, Sun recommends that system administrators register for their organization.

TABLE 5–2 Ways to Disable User Registration

Before or After Solaris Installation	Description	For More Information
Before Solaris software is installed	■ Deselect the `SUNWsregu` package (interactive installation) ■ Modify a custom JumpStart profile to not install the `SUNWsregu` package ■ Create and run a finish script that creates a file named `solregis` in the `/etc/default` directory on one or more systems with the following line in it: `DISABLE=1`	*Solaris 9 Installation Guide* `solregis`(1)
After Solaris software is installed	■ Use the `pkgrm` command to remove the `SUNWsregu` package ■ Add the `solregis` file in the `/etc/default` directory (custom JumpStart installation only)	Chapter 23 *Solaris 9 Installation Guide* `solregis`(1)

CHAPTER 6

Managing Server and Client Support Topics

This topic map lists the chapters that provide information on managing server and client support.

Chapter 7	Provides a high-level overview about managing server and client support on a network. This chapter describes the different system types for which you can add support, and guidelines for choosing a system type for your environment.
Chapter 8	Provides step-by-step instructions for managing diskless client support with the `smosservice` and `smdiskless` commands.

CHAPTER 7

Managing Server and Client Support (Overview)

This chapter describes the management of server and client support on a network, and it provides overview information about each system configuration (referred to as a *system type*) that is supported in the Solaris environment. This chapter also includes guidelines for selecting the appropriate system type to meet your needs.

This is a list of the overview information in this chapter.

- "What's New in Server and Client Management?" on page 113
- "Where to Find Server and Client Tasks" on page 114
- "What Are Servers, Clients, and Appliances?" on page 114
- "What Does Client Support Mean?" on page 115
- "Overview of System Types" on page 116
- "Diskless Client Management Overview" on page 119

For step-by-step instructions about how to manage diskless client support, see Chapter 8.

What's New in Server and Client Management?

This section describes new server and client management features.

Diskless Client Support

In this Solaris 9 release, you can manage diskless clients with the `smosservice` and `smdiskless` commands. Diskless clients are systems with no disks that depend on servers for all their services.

These commands are part of the Solaris Management Console tool suite. You cannot use the Solaris Management Console to manage diskless clients. You can only use the `smosservice` and `smdiskless` commands to manage diskless clients.

For more information on managing diskless clients, see "Diskless Client Management Overview" on page 119 and Chapter 8.

Where to Find Server and Client Tasks

Use this table to find step-by-step instructions for setting up server and client support.

Server/Client Services	For More Information
Install or JumpStart clients	*Solaris 9 Installation Guide*
Diskless client systems in the Solaris 9 environment	"Diskless Client Management Overview" on page 119 and Chapter 8
Diskless client systems and Solstice AutoClient systems in previous Solaris releases	*Solstice AdminSuite 2.3 Administration Guide*
AutoClient 3.0.1 systems in the Solaris 8 or Solaris 9 environments	Call your service provider

What Are Servers, Clients, and Appliances?

Systems on the network can usually be described as one of the following:

System Type	Description
Server	A system that provides services to other systems in its network. There are file servers, boot servers, web servers, database servers, license servers, print servers, installation servers, appliance servers, and even servers for particular applications. This chapter uses the term server to mean a system that provides boot services and file systems for other systems on the network.

System Type	Description
Client	A system that uses remote services from a server. Some clients have limited disk storage capacity, or perhaps none at all, and they have to rely on remote file systems from a server to function. Diskless systems, AutoClient systems, and appliance systems are examples of this type of client. Other clients might use remote services (such as installation software) from a server, but they don't rely on a server to function. A standalone system, which has its own hard disk containing the root (/), `/usr`, and `/export/home` file systems and swap space, is a good example of this type of client.
Sun Cobalt Server Appliance	The Sun Cobalt server appliance provides an integrated set of pre-configured Internet services. Users of the server appliance just need a web browser and an IP address. Administration on the servers is centralized and the appliance users require no client administration. For more information, see `http://www.sun.com/hardware/serverappliances`.
Appliance	A network appliance such as the Sun Ray appliance provides access to applications and the Solaris environment. An appliance gives you centralized server administration and no client administration or upgrades. Sun Ray appliances also provide *hot desking*, which is the ability to instantly access your computing session from any appliance in the server group, exactly where you left off. For more information, see `http://www.sun.com/products/sunray`.

What Does Client Support Mean?

Support for a client means providing software and services to help the client function. Support can include the following:

- Making a system known to the network (host name and Ethernet address information)
- Providing installation services to remotely boot and install a system
- Providing operating system (OS) services and application services to a system with limited disk space or no disk space

Overview of System Types

System types are sometimes defined by how they access the root (/) and /usr file systems, including the swap area. For example, standalone systems and server systems mount these file systems from a local disk, while other clients mount the file systems remotely, relying on servers to provide these services. This table lists some of the characteristics of each system type.

TABLE 7–1 Characteristics of General System Types

System Type	Local File Systems	Local Swap?	Remote File Systems	Network Use	Relative Performance
Server	root (/) /usr /home /opt /export/home /export/root	Yes	– *None* –	High	High
Standalone System	root (/) /usr /export/home	Yes	– *None* –	Low	High
Diskless Client	– *None* –	No	root (/) swap /usr /home	High	Low
AutoClient System	Cached root (/) Cached /usr	Yes	/var	Low	High
Appliance	None	None	None	High	High

Servers

A server system contains the following file systems:

- The root (/) and /usr file systems, plus swap space
- The /export and /export/home file systems, which support client systems and provide home directories for users

- The `/opt` directory or file system for storing application software

Servers can also contain the following software to support other systems:

- Operating system (OS) services for diskless systems or AutoClient systems that are running a different release or clients that are a different platform than the server
- Solaris CD image software and boot software for networked systems to perform remote installations
- JumpStart™ directory for networked systems to perform custom JumpStart installations

Standalone Systems

A *networked standalone system* can share information with other systems in the network, but it can continue to function if detached from the network.

A standalone system can function autonomously because it has its own hard disk that contains the root (`/`), `/usr`, and `/export/home` file systems and swap space. The standalone system thus has local access to operating system software, executables, virtual memory space, and user-created files.

Note – A standalone system requires sufficient disk space to hold its necessary file systems.

A *non-networked standalone system* is a standalone system with all the characteristics listed above, except it is not connected to a network.

Diskless Clients

A *diskless client* has no disk and depends on a server for all its software and storage needs. A diskless client remotely mounts its root (`/`), `/usr`, and `/home` file systems from a server.

A diskless client generates significant network traffic due to its continual need to procure operating system software and virtual memory space from across the network. A diskless client cannot operate if it is detached from the network or if its server malfunctions.

For more overview information about diskless clients, see “Diskless Client Management Overview” on page 119.

AutoClient Systems

An AutoClient system is nearly identical to a diskless client in terms of installation and administration. An AutoClient system has the following characteristics:

- Requires a minimum of a 100-Mbyte local disk for swapping and for caching its individual root (/) file system and the `/usr` file system from a server
- Can be set up so that it continues to access its cache when the server is unavailable
- Relies on a server to access other file systems and software applications
- Contains no permanent data, making it a field-replaceable unit (FRU)

Appliances

An appliance, such as the Sun Ray appliance, is an X display device that requires no administration. There is no CPU, fan, disk, and very little memory. An appliance is connected to a Sun display monitor, but the appliance user's desktop session is run on a server and displayed back to the user. The X environment is setup automatically for the user and has the following characteristics:

- Relies on a server to access other file systems and software applications
- Provides centralized software administration and resource sharing
- Contains no permanent data, making it a field-replaceable unit (FRU)

Guidelines for Choosing System Types

You can determine which system types are appropriate for your environment by comparing each system type based on the following characteristics:

- Centralized Administration
  - Can the system be treated as a field-replaceable unit (FRU)? This means that a broken system can be quickly replaced with a new system without any lengthy backup and restore operations and no loss of system data.
  - Does the system need to be backed up? Large costs in terms of time and resources can be associated with backing up a large number of desktop systems.
  - Can the system's data be modified from a central server?
  - Can the system be installed from a centralized server, quickly and easily, without handling the client system's hardware?
- Performance
  - Does this configuration perform well in desktop usage?
  - Does the addition of systems on a network affect the performance of other systems already on the network?
- Disk Space Usage

- How much disk space is required to effectively deploy this configuration?

This table describes how each system type scores in terms of each category. A ranking of 1 is most efficient. A ranking of 4 is least efficient.

TABLE 7–2 Comparison of System Types

System Type	Centralized Administration	Performance	Disk Usage
Standalone System	4	1	4
Diskless Client	1	4	1
AutoClient System	1	2	2
Appliance	1	1	1

Diskless Client Management Overview

The following sections and Chapter 8 describe how to manage diskless client support in the Solaris 9 release.

A *diskless client* is a system that depends on an *OS server* for its operating system, software, and storage. A diskless client mounts its root (/), `/usr`, and other file systems from its OS server. A diskless client has its own CPU and physical memory and can process data locally. However, a diskless client cannot operate if it is detached from its network or if its OS server malfunctions. A diskless client generates significant network traffic because of its continual need to function across the network.

In previous Solaris releases, diskless clients were managed with the Solstice graphical management tools. In the Solaris 9 release, the diskless client commands, `smosservice` and `smdiskless`, enable you to manage OS services and diskless client support.

OS Server and Diskless Client Support Information

The following table describes which Solaris releases and architecture types are supported by the `smosservice` and `smdiskless` commands.

Architecture Type	Solaris 2.6	Solaris 7	Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02	Solaris 9
SPARC Servers	Supported	Supported	Supported	Supported
x86 Servers	Supported	Supported	Supported	Supported
SPARC Clients	Supported	Supported	Supported	Supported
x86 Clients	Not Supported	Not Supported	Not Supported	Supported

This table describes the combination of OS server-client configurations that are supported by the `smosservice` and `smdiskless` commands.

	Solaris 2.6 Release Support	Solaris 7 Release Support	Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02 Support	Solaris 9 Support
OS Server-Client OS Release	Solaris 2.6–Solaris 2.6	Solaris 7–Solaris 2.6, or 7	Solaris 8 1/01, 4/01, 7/01, 10/01, 2/02–Solaris 2.6, 7, or 8 1/01, 4/01, 7/01, 10/01, 2/02	Solaris 9–Solaris 2.6, 7, 8 1/01, 4/01, 7/01, 10/01, 2/02

Diskless Client Management Features

You can use the `smosservice` and `smdiskless` commands to add and maintain diskless client support on a network. By using a name service, you can manage system information in a centralized manner so that important system information, such as host names, does not have to be duplicated on every system in the network.

You can do the following tasks with the `smosservice` and `smdiskless` commands:

- Add and modify diskless client support
- Add and remove OS services
- Manage diskless client information in the LDAP, NIS, NIS+, or files environment

You can only use the diskless client commands to set up diskless client booting. You cannot use them to set up other services, such as remote installation or profile services. Set up remote installation services by including diskless client specifications in the `sysidcfg` file. For more information, see *Solaris 9 Installation Guide*.

Working With Diskless Client Commands

By writing your own shell scripts and using the commands shown in the following table, you can easily set up and manage your diskless client environment.

TABLE 7–3 Diskless Client Commands

Command	Subcommand	Task
`/usr/sadm/bin/smosservice`		
	`add`	Add OS services
	`delete`	Delete OS services
	`list`	List OS services
	`patch`	Manage OS service patches
`/usr/sadm/bin/smdiskless`		
	`add`	Add a diskless client to an OS server
	`delete`	Delete a diskless client from an OS server
	`list`	List the diskless clients on an OS server
	`modify`	Modify the attributes of a diskless client

You can obtain help on these commands in two ways:

- Use the `-h` option when you type the command, subcommand, and required options. For example, to display the usage statement for `smdiskless add` type the following:

  ```
  % /usr/sadm/bin/smdiskless add -p my-password -u my-user-name -- -h
  ```

- View the `smdiskless`(1M) or `smosservice`(1M) man pages.

Required RBAC Rights for Diskless Client Management

You can use the `smosservice` and `smdiskless` commands as superuser. If you are using Role-Based Access Control (RBAC), you can use of either a subset or all of the diskless client commands, according to the RBAC rights to which they are assigned. The following table lists the RBAC rights that are required to use the diskless client commands.

TABLE 7–4 Required Rights For Diskless Client Management

RBAC Right	Command	Task
Basic Solaris User, Network Management	`smosservice list`	List OS services
	`smosservice patch`	List OS services patches
	`smdiskless list`	List diskless clients
Network Management	`smdiskless add`	Add diskless clients
System Administrator	All commands	All tasks

Adding OS Services

A Solaris OS server is a server that provides operating system (OS) services to support diskless client systems. You can add support for an OS server or convert a standalone system to an OS server with the `smosservice` command.

For each platform group and Solaris release that you want to support, you must add the particular OS service to the OS server. For example, if you want to support SPARC® Sun4m systems running the Solaris 8 release, you must add Sun4m/Solaris 8 OS services to the OS server. You would also still need to add OS services to support SPARC Sun4c systems or x86 based systems that runs the Solaris 8 release, because they are different platform groups.

You must have access to the appropriate Solaris CD or disk image to add OS services.

Adding OS Services When the OS Server Has Been Patched

When adding OS services to an OS server, you might see error messages saying that you have inconsistent versions of the OS running on the server and the OS that you are trying to add. This message occurs when the installed version of the OS has packages that were previously patched and the OS services being added do not have those packages patched (because the patches have been integrated into the packages).

For example, you may have a server that is running the Solaris 7 release. You may also have additional OS services loaded on this server, including the Solaris 2.6 SPARC sun4m OS services that have been patched. If you try to add the Solaris 2.6 SPARC sun4c OS services from a CD-ROM to this server, you could get the following error message:

```
Error: inconsistent revision, installed package appears to have been
patched resulting in it being different than the package on your media.
You will need to backout all patches that patch this package before
retrying the add OS service option.
```

Disk Space Requirements for OS Servers

Before you set up your diskless client environment, make sure you have the required disk space available for each diskless client directory.

In previous Solaris releases, you were prompted about diskless client support during the installation process. In the Solaris 9 release, you must manually allocate an `/export` file system either during installation or create it after installation. See the following table for specific disk space requirements.

TABLE 7–5 Disk Space Requirements for OS Servers

Directory	Required Space in Mbytes
`/export/Solaris_`*version*	10
`/export/exec`	800
`/export/share`	5
`/export/swap/`*diskless_client*	32 (default size)
`/export/dump/`*diskless_client*	32 (default size)
`/export/root/templates/Solaris_`*version*	30
`/export/root/clone/Solaris_`*version*`/`*machine_class*	30 through 60 (depends on machine class)
`/export/root/`*diskless_client*(clone of above)	30 through 60 (depends on machine class)
`/tftpboot/inetboot.`*machine_class*`.Solaris_`*version*	200 Kbytes per *machine_class*`.Solaris_`*version*

CHAPTER 8

Managing Diskless Clients (Tasks)

This chapter describes how to manage diskless clients in the Solaris environment.

For information on the procedures associated with managing diskless clients, see "Managing Diskless Clients (Task Map)" on page 125.

For overview information on managing diskless clients, see Chapter 7.

For information about managing clients with Solstice AdminSuite™ software, see *Solstice AdminSuite 2.3 Administration Guide*.

Managing Diskless Clients (Task Map)

The following table identifies the procedures needed to manage diskless clients.

Task	Description	For Instructions
1. (Optional) Remove existing diskless client support	If you have existing diskless clients that were added with the Solstice AdminSuite product, remove the diskless client support and OS services with the `admhostdel` and `admhostmod` commands before installing the Solaris 9 release.	*Solstice AdminSuite 2.3 Administration Guide*

Task	Description	For Instructions
3. (Optional) Enable Solaris Management Console logging to view diskless client error messages	Choose Log Viewer from the console main window to view diskless client error messages.	"Starting the Solaris Management Console" on page 54
4. Prepare for adding a diskless client	Verify supported releases and identify the *platform*, *mediapath*, and *cluster* (or software group) of each diskless client.	"How to Prepare for Adding Diskless Clients" on page 128
5. Add required OS services to an OS server	Add the OS services for the diskless clients you want to support with the `smosservice` command. You must identify the platform, media path, and each diskless client platform that you want to support.	"How to Add OS Services For Diskless Client Support" on page 129
6. Add a diskless client	Add diskless client support by specifying all required information with the `smdiskless` command.	"How to Add a Diskless Client" on page 131
7. Boot the diskless client	Verify that the diskless client support is successfully added by booting the diskless client.	"How to Boot a Diskless Client" on page 132
8. (Optional) Delete diskless client support	Delete support for a diskless client if it is no longer required.	"How to Delete Diskless Client Support" on page 133
9. (Optional) Delete OS services for a diskless client	Delete OS services for a diskless client if they are no longer needed.	"How to Delete OS Services for Diskless Clients" on page 133
10. (Optional) Patch OS services	Add, delete, list, or synchronize patches for diskless client OS services.	"How to Add an OS Patch for a Diskless Client" on page 135

Managing Diskless Clients

These sections describe the procedures needed to manage diskless clients.

Keep the following key points in mind when managing diskless clients:

- The Solaris installation program doesn't prompt you to set up diskless client support. You must manually create an `/export` partition to support diskless clients. You create the `/export` partition during or after the installation process.
- The `/export` partition must contains a minimum of 800–1000 Mbytes, depending upon the number of clients supported. For specific information, see "Disk Space Requirements for OS Servers" on page 123.
- The name service identified in the `smosservice` or `smdiskless` commands must match the primary name service identified in the `/etc/nsswitch.conf` file. If you don't specify a name service in the `smdiskless` or `smosservice` commands, the default name service is `files`.

After you determine the platform, media path, and cluster for each diskless client, you are ready to add OS services. The following directories are created and populated for each OS service that you add:

- `/export/Solaris_`*version*`/Solaris_`*version_instruction_set*`.all` (symbolic link to `/export/exec/Solaris_`*version*`/Solaris_`*version_instruction_set*`.all`)
- `/export/Solaris_`*version*
- `/export/Solaris_`*version*`/var`
- `/export/Solaris_`*version*`/opt`
- `/export/share`
- `/export/root/templates/Solaris_`*version*
- `/export/root/clone`
- `/export/root/clone/Solaris_`*version*
- `/export/root/clone/Solaris_`*version*`/`*machine_class*

The following default directories are created and populated on the OS server for each diskless client that you add:

- `/export/root/`*diskless_client*
- `/export/swap/`*diskless_client*
- `/tftpboot/`*diskless_client_ipaddress_in_hex*`/export/dump/`*diskless_client* (if you specify the `-x` *dump* option)

Note – You can modify the default locations of the root, `/swap`, and `/dump` directories by using the `-x` option. However, do not create these directories under the `/export` file system.

▼ How to Prepare for Adding Diskless Clients

Make sure that the system intended to be the OS service is running a supported release. Also verify that the combination of OS server release and diskless client release is supported.

When you use the `smosservice add` command to add OS services, you must specify the *platform*, *mediapath*, and *cluster* (or software group) of each diskless client platform that you want to support.

1. **Verify that the intended OS server and diskless client will be running a combination of Solaris releases that are supported.**

   For more information, see "OS Server and Diskless Client Support Information" on page 120.

2. **Identify the diskless client platform by using this format:**

   instruction_set.machine_class.Solaris_version

   For example:

   `sparc.sun4u.Solaris_9`

   The following are the possible platform options:

instruction_set	*machine_class*	*Solaris_version*
sparc	sun4d*, sun4c*, sun4m, sun4u,	Solaris_9, Solaris_8, Solaris_2.7, Solaris_2.6
i386	i86pc	Solaris_9, Solaris_8, Solaris_2.7, Solaris_2.6

   * The sun4c architecture is not supported in the Solaris 8 and Solaris 9 releases. The sun4d architecture is not supported in the Solaris 9 release.

3. **Identify the media path, which is the full path to the disk image that contains the operating system that you want to install for the diskless client.**

   The Solaris 9 operating environment is delivered on multiple CDs. However, you cannot use the `smosservice` command to load OS services from a multiple CD distribution. You must run the scripts that are found on the Solaris 9 software CDs (and optional Language CD) to do the following:

   - Create an install image on a server. For information on setting up an install server, refer to *Solaris 9 Installation Guide*.
   - Load the required OS services from the CD image using one of the following scripts:
     - CD 1 of 2 – `/cdrom/cdrom0/s0/Solaris_9/Tools/setup_install_server`

- CD 2 of 2 – `/cdrom/cdrom0/s0/Solaris_9/Tools/add_to_install_server`
- Language CD – `/cdrom/cdrom0/s0/Solaris_9/Tools/add_to_install_server`

For example, if you are using the `setup_install_server` script from the Solaris 9 Software 1 of 2 SPARC Platform Edition CD on a locally connected CD-ROM device, the syntax looks something like this:

```
# mkdir /export/install/sparc_9
# cd /cd_mount_point/Solaris_9/Tools
# ./setup_install_server /export/install/sparc_9
```

- After the Solaris CD image is installed on the disk, specify the disk image path. For example:

```
/net/export/install/sparc_9
```

4. **Identify the `SUNWCXall` cluster when you add OS services.**

   You must use *the same cluster* for diskless clients that run the same operating environment on the same system (SPARC or IA).

   For example, consider the following diskless clients:

   - `sparc.sun4m.Solaris_9`
   - `sparc.sun4u.Solaris_9`

   To set up these diskless clients, you would need to specify the `SUNWCXall` cluster for each diskless client because the `sun4u` and `sun4m` systems require the `SUNWCXall` cluster. In addition, diskless clients that run the same operating environment (in this situation, Solaris_9) on the same system must use the same cluster.

Note – If you are using a `sun4u` system, or if you are using a system with an accelerated 8-bit color memory frame buffer (cgsix), you *must* specify `SUNWCXall` as the cluster.

▼ How to Add OS Services For Diskless Client Support

Use this procedure to add OS services for a diskless client on the server.

1. **Become superuser or assume an equivalent role on the server.**

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 48.

2. **Verify that the Solaris Management Console server is running and that the diskless client tools are available on the system.**

```
# /usr/sadm/bin/smosservice list -H starbug:898 --
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
was successful.
Platform
--------------------------------------------------------------------------
```

3. **Add the OS services.**

```
# /usr/sadm/bin/smosservice add -H hostname:898 -- -o hostname
-x mediapath=path -x platform=instruction-set.machine-class.Solaris-version
-x cluster=cluster-name -x locale=locale-name
```

`add`	Adds the specified OS service.
`-H` *hostname:898*	Specifies the host name and port to which you want to connect. If you do not specify a port, the system connects to the default port, 898.
`--`	Identifies that the subcommand arguments start after this point.
`-x` *mediapath=path*	Specifies the full path to the Solaris image.
`-x` *platform=instruction-set.machine-class.Solaris-version*	Specifies the instruction architecture, machine class, and the Solaris version to be added.
`-x` *cluster=cluster-name*	Specifies the Solaris cluster to install.
`-x` *locale=locale-name*	Specifies the locale to install.

Note – The installation process can take approximately 45 minutes, depending on the server speed and the OS service configuration you choose.

For more information, see `smosservice`(1M).

4. **(Optional) Continue to add the other OS services.**

5. **When you are finished adding OS services, verify that the OS services were installed.**

```
# /usr/sadm/bin/smosservice list -H hostname:898 --
```

Example—Adding an OS Service for Diskless Client Support

This example describes how to add Solaris 8 OS services on the server `starbug`. The server `starbug` is running the Solaris 9 release.

```
# /usr/sadm/bin/smosservice add -H starbug:898 -- -o starbug
-x mediapath=/net/install/export/sparc_8 -x platform=sparc.sun4u.Solaris_8
-x cluster=SUNWCXall -x locale=en_US
Authenticating as user: root

Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password :: xxx
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
was successful.
```

▼ How to Add a Diskless Client

Use this procedure to add a diskless client after you have added OS services.

1. **Become superuser or assume an equivalent role.**

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 48.

2. **Add the diskless client.**

   ```
   # /usr/sadm/bin/smdiskless add -- -i ip-address -e ethernet-address
   -n client-name -x os=instruction-set.machine-class.Solaris-version
   -x root=/export/root/client-name -x swap=/export/swap/client-name
   -x swapsize=size -x tz=timezone -x locale=locale-name
   ```

`add`	Adds the specified diskless client.
`--`	Identifies that the subcommand arguments start after this pont.
`-i` *ip-address*	Identifies the IP address of the diskless client.
`-e` *ethernet-address*	Identifies the Ethernet address of the diskless client.
`-n` *client-name*	Specifies the name of the diskless client.
`-x os=`*instruction-set.machine-class.Solaris-version*	Specifies the instruction architecture, machine class, OS, and the Solaris version for the diskless client.

`-x root=root=/export/root/`*client-name*	Identifies the root directory for the diskless client.
`-x swap=root=/export/root/`*client-name*	Identifies the swap file for the diskless client.
`-x swapsize=`*size*	Specifies the size of the swap file in Mbytes. The default is 24 Mbytes.
`-x tz=`*timezone*	Specifies the timezone for the diskless client.
`-x locale=`*locale-name*	Specifies the locale to install for the diskless client.

For more information, see `smdiskless`(1M).

3. **(Optional) Continue to use the `smdiskless add` command to add each diskless client.**

4. **Verify that the diskless clients were installed.**

```
# /usr/sadm/bin/smosservice list -H hostname:898 --
```

Examples—Adding a Diskless Client

This example shows how to add a Solaris 8 client, `earth`, from the server `starbug`.

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.27 -e 8:0:20:1f:33:9e
-n earth -x os=sparc.sun4m.Solaris_8 -x root=/export/root/earth
-x swap=/export/swap/earth -x swapsize=64 -x tz=US/Mountain -x locale=en_US
```

This example shows how to add a Solaris 7 client, `earth2`, from the server `starbug`.

```
# /usr/sadm/bin/smdiskless add -- -i 172.20.27.26 -e 8:0:20:1f:32:be
-n earth2 -x os=sparc.sun4m.Solaris_2.7 -x root=/export/root/earth2
-x swap=/export/swap/earth2 -x swapsize=64 -x tz=US/Mountain
```

▼ How to Boot a Diskless Client

1. **Verify the following prerequisites on the OS server:**

   - Confirm that the name service used to add the diskless client and the OS services matches the primary name in the server's `/etc/nsswitch.conf` file.

     Otherwise, the diskless client won't boot.

   - Confirm that the `rpc.bootparamd` daemon is running. If it is not running, start it.

2. **Boot the diskless client.**

```
ok boot net
```

▼ How to Delete Diskless Client Support

1. **Become superuser or assume an equivalent role.**

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 48.

2. **Remove the diskless client support.**

   # **/usr/sadm/bin/smdiskless delete -- -o** *hostname*:898 **-n** *client-name*

3. **Verify that the diskless client support is removed.**

   # **/usr/sadm/bin/smosservice list -H** *hostname*:898 **--**

Example—Deleting Diskless Client Support

This example shows how to delete the diskless client `earth` from the OS server `starbug`.

```
# /usr/sadm/bin/smdiskless delete -- -o starbug -n earth
Authenticating as user: root

Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password ::
Starting SMC server version 2.0.0.
endpoint created: :898
SMC server is ready.
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
was successful.
```

▼ How to Delete OS Services for Diskless Clients

1. **Become superuser or assume an equivalent role.**

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 48.

2. **Remove the OS services for the diskless clients.**

   # **/usr/sadm/bin/smosservice delete -H** *hostname*:898 **--**
   -x rmplatform=*instruction-set.machine-class.Solaris-version*

3. **Verify that the OS services are removed.**

   # **/usr/sadm/bin/smosservice list -H** *hostname*:898 **--**

Example—Deleting OS Services for Diskless Clients

The following example shows how to delete the diskless client OS services (`sparc.all.Solaris_9`) from the server `starbug`.

```
# /usr/sadm/bin/smosservice delete -H starbug:898 --
-x rmplatform=sparc.all.Solaris_9
Authenticating as user: root
Type /? for help, pressing enter accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug:898
was successful.
```

Patching Diskless Client OS Services

You use the `smosservice patch` command to do the following:

- Establish the `/export/diskless/Patches` patch spool directory on an OS server.
- Add patches to the patch spool directory. If the patch you are adding obsoletes an existing patch in the spool, the obsolete patch is moved to `/export/diskless/Patches/Archive`.
- Delete patches from the patch spool directory.
- List the patches in the patch spool directory.
- Synchronize spooled patches out to clients. You must reboot each synchronized client for the client to recognize the patch update.

Note – Keep your OS servers up to date by installing recommended OS patches on a timely basis.

For information on downloading patches, see "How to Download an Unsigned Solaris Patch" on page 334.

Displaying OS Patches for Diskless Clients

Diskless client patches are logged in different directories, depending on the type of patch:

- Kernel patches are logged in the diskless client's `/var/sadm/patch` directory. To display kernel patches, type the following command on the diskless client:

```
% patchadd -p
```

- /usr patches are logged in the OS server's /export/Solaris_*version*/var/patch directory. A directory is created for each patch ID. To display /usr patches, type the following command on the OS server:

```
% patchadd -S Solaris_8 -p
Patch: 111879-01 Obsoletes: Requires: Incompatibles: Packages: SUNWwsr
```

To list all spooled patches by OS and architecture, use the smosservice command with the -P option.

▼ How to Add an OS Patch for a Diskless Client

1. **Become superuser or assume an equivalent role on the server.**

   For more information, see "How to Become Superuser (root) or Assume a Role" on page 48.

2. **Log in to the diskless client system and shut it down.**

```
# init 0
```

3. **Add the patch to a spool directory.**

```
# /usr/sadm/bin/smosservice patch -- -a /var/patches/patch-ID-revision
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
was successful.
```

   If the patch to add depends on another patch, adding the patch fails with the following message:

```
The patch patch-ID-revision could not be added
because it is dependent on other patches which have not yet been spooled.
You must add all required patches to the spool first.
```

4. **Verify the patch is spooled.**

```
# /usr/sadm/bin/smosservice patch -- -P
```

5. **Push the spooled patch to the diskless client.**

```
# /usr/sadm/bin/smosservice patch -- -m -U
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
```

```
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
was successful.
```

Note – Pushing and synchronizing the patch to the diskless client can take up to 90 minutes per patch.

---

6. **Verify the patch is applied to the diskless client.**

```
# /usr/sadm/bin/smosservice patch -- -P
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
Login to starbug as user root was successful.
Download of com.sun.admin.osservermgr.cli.OsServerMgrCli from starbug
was successful.
Patches In Spool Area
Os Rel Arch   Patch Id  Synopsis
--------------------------------------------------------------------------
8      sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr

Patches Applied To OS Services
Os Service                              Patch
--------------------------------------------------------------------------
Solaris_8

Patches Applied To Clone Areas
Clone Area                              Patch
--------------------------------------------------------------------------
Solaris_8/sun4m
```

Example—Adding an OS Patch for a Diskless Client

This example shows how to add a Solaris 8 patch (111879-01) to the diskless client's OS services on the server.

```
# /usr/sadm/bin/smosservice patch -- -a /var/patches/111879-01
Authenticating as user: root
.
.
.
# /usr/sadm/bin/smosservice patch -- -P
Patches In Spool Area
Os Rel Arch   Patch Id  Synopsis
--------------------------------------------------------------------------
8      sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr
.
.
```

```
.
# /usr/sadm/bin/smosservice patch -- -m -U
Authenticating as user: root
.
.
.
# /usr/sadm/bin/smosservice patch -- -P
Authenticating as user: root
.
.
.
Patches In Spool Area
Os Rel Arch   Patch Id  Synopsis
--------------------------------------------------------------------------------
8      sparc  111879-01 SunOS 5.8: Solaris Product Registry patch SUNWwsr

Patches Applied To OS Services
Os Service                              Patch
--------------------------------------------------------------------------------
Solaris_8

Patches Applied To Clone Areas
Clone Area                              Patch
--------------------------------------------------------------------------------
Solaris_8/sun4m
```

Troubleshooting Diskless Client Problems

This section lists some common problems with diskless clients and possible solutions.

Problem

- OS server does not respond to client RARP requests
- OS server does not respond to client `bootparam` requests
- OS server cannot mount diskless client root file system

Solution

In a files environment

- Verify that `files` is listed as the first source for `hosts`, `ethers`, and `bootparams` in the `/etc/nsswitch.conf` file on the OS server.
- Verify that the client's IP address appears in the `/etc/inet/hosts` file.
- Verify that the client's Ethernet address appears in the `/etc/ethers` file.
- Verify that the `/etc/bootparams` file contains the following paths to the client's root and swap areas:

client `root=`*os-server*`:/export/root/`*client* `swap=`*os-server*`:/export/swap/`*client*

The swap size varies depending on whether you specify the `-x` *swapsize* option when you add the diskless client. If you specify the `-x` *dump* option when you add the diskless client, the following line is present:

`dump=`*os-server*`:/export/dump/`*client* `dumpsize=24`

The dump size varies depending on whether you specify the `-x` *dumpsize* option when you add the diskless client.

- Verify that the OS server's IP address appears in the `/export/root/`*client*`/etc/inet/hosts` file.

In a name service environment

- Verify that both the OS server's and the client's Ethernet address and IP address are correctly mapped.
- Verify that the `/etc/bootparams` file contains the paths to the client's root and swap areas, as follows:

client `root=`*os-server*`:/export/root/`*client* `swap=`*os-server*`:/export/swap/`*client* `swapsize=24`

The swap size varies depending on whether you specify the `-x` *swapsize* option when you add the diskless client. If you specify the `-x` *dump* option when you add the diskless client, the following line is present:

`dump=`*os-server*`:/export/dump/`*client* `dumpsize=24`

The dump size varies depending on whether you specify the `-x` *dumpsize* option when you add the diskless client.

Problem
Diskless client panics

Solution

- Verify that the OS server's Ethernet address is correctly mapped to its IP address. If you physically moved a system from one network to another, you might have forgotten to remap the system's new IP address.
- Verify that the client's host name, IP address, and Ethernet address do not exist in the database of another server *on the same subnet* that responds to the client's RARP, TFTP, or `bootparam` requests. Often, test systems are set up to install their OS from an install server. In these cases, the install server answers the client's RARP or `bootparam` request, returning an incorrect IP address. This incorrect address might result in the download of a boot program for the wrong architecture, or a failure to mount the client's root file system.
- Verify that the diskless client's TFTP requests are not answered by an install server (or previous OS server) that transfers an incorrect boot program. If the boot program is of a different architecture, the client immediately panics. If the

boot program loads from a non-OS server, the client might obtain its root partition from the non-OS server and its /usr partition from the OS server. In this situation, the client panics if the root and /usr partitions are of conflicting architectures or versions.

- If you are using both an install server and an OS server, verify that the following entry exists in the /etc/dfs/dfstab file:

  ```
  share -F nfs -o -ro /export/exec/Solaris_version_instruction_set.all/usr
  ```

  Where *version*=2.6, 2.7, or 8, and *instruction_set*=sparc or i386.

- Verify that the diskless client's root, /swap, and /dump (if specified) partitions have share entries:

  ```
  share -F nfs -o rw=client,root=client /export/root/client
  share -F nfs -o rw=client,root=client /export/swap/client
  share -F nfs -o rw=client,root=client /export/dump/client
  ```

- On the OS server, type the following to check which files are shared:

  ```
  % share
  ```

  The OS server must share /export/root/*client* and /export/swap/*client_name* (defaults), or the root, /swap, and /dump partitions you specified when you added the diskless client.

  Verify that the following entries exists in the /etc/dfs/dfstab file:

  ```
  share -F nfs -o ro /export/exec/Solaris_version_instruction_set.all/usr
  share -F nfs -o rw=client,root=client /export/root/client
  share -F nfs -o rw=client,root=client /export/swap/client
  ```

Problem
OS server is not responding to diskless client's RARP request

Solution
From the client's intended OS server, run the snoop command as root by using the client's Ethernet address:

```
# snoop xx:xx:xx:xx:xx:xx
```

Problem
Boot program downloads, but panics early in the process

Solution
Using the snoop command, verify that the intended OS server is answering the client's TFTP and NFS requests.

Problem

- Diskless client hangs
- Incorrect server responds to diskless client's RARP request

Solution
Restart the following daemons on the OS server:

```
# /usr/sbin/rpc.bootparamd
# /usr/sbin/in.rarpd -a
```

CHAPTER 9

Shutting Down and Booting a System Topics

This topic map lists the chapters that provide information on shutting down and booting a system.

Chapter 10	Provides overview information and guidelines for shutting down and booting a system.
Chapter 11	Provides overview information and tasks that are related to run levels and boot files.
Chapter 12	Provides step-by-step instructions for shutting down a system.
Chapter 13	Provides step-by-step instructions for booting a SPARC based system.
Chapter 14	Provides step-by-step instructions for booting an x86 based system.
Chapter 15	Provides a high-level overview of the boot process for both SPARC based and x86 based systems.

CHAPTER 10

Shutting Down and Booting a System (Overview)

This chapter provides guidelines for shutting down and booting a system. The Solaris software environment is designed to run continuously so that electronic mail and network resources are available to users. Occasionally, it is necessary to shut down or reboot a system because of a system configuration change, a scheduled maintenance event, or a power outage.

This is a list of the overview information in this chapter.

- "What's New in Shutting Down and Booting a System?" on page 143
- "Where to Find Shutting Down and Booting Tasks" on page 144
- "Shutting Down and Booting Terminology" on page 145
- "Guidelines for Shutting Down a System" on page 145
- "Guidelines for Booting a System" on page 146
- "Booting a System From the Network" on page 147
- "When to Shut Down a System" on page 147
- "When to Boot a System" on page 148

What's New in Shutting Down and Booting a System?

This section describes new features that are related to shutting down and booting a system in this Solaris release.

PXE Network Boot

You can boot the Solaris 9 Operating Environment (x86 Platform Edition) directly from a network without the Solaris boot diskette on x86 based systems that support the Preboot Execution Environment (PXE) network booting protocol. The PXE network boot is available only for devices that implement the Intel Preboot Execution Environment specification.

You can enable the PXE network boot on the client system by using the BIOS setup program in the system BIOS, the network adapter BIOS, or both. On some systems, you must also adjust the boot device priority list so that a network boot is attempted before a boot from other devices. See the manufacturer's documentation for each setup program, or watch for setup program entry instructions during boot.

Some PXE-capable network adapters have a feature that enables a PXE boot if you type a particular keystroke in response to a brief boot-time prompt. This feature is ideal when you use PXE for an install boot on a system that normally boots from the disk drive because you do not have to modify the PXE settings. If your adapter does not have this feature, disable PXE in the BIOS setup when the system reboots after installation, and the system will boot from the disk drive.

Some early versions of PXE firmware cannot boot the Solaris system. If you have one of these older versions, your system can read the PXE network bootstrap program from a boot server, but the bootstrap will not transmit packets. If this problem occurs, upgrade the PXE firmware on the adapter. Obtain firmware upgrade information from the adapter manufacturer's web site. For more information, see `elxl`(7D) and`iprb`(7D).

For information on booting x86 based systems with or without the boot diskette, see "x86: How to Boot a System From the Network" on page 197.

Where to Find Shutting Down and Booting Tasks

Use these references to find step-by-step instructions for shutting down and booting a system.

Shut Down and Boot Task	For More Information
Shut down a SPARC based system or an x86 based system	Chapter 12

Shut Down and Boot Task	For More Information
Boot a SPARC based system	Chapter 13
Boot an x86 based system	Chapter 14
Manage a SPARC based system with the power management software	`power.conf`(4), `pmconfig`(1M)

Shutting Down and Booting Terminology

This section describes the terminology that is used in shutting down and booting a system.

- **Run levels and init states** – A *run level* is a letter or digit that represents a system state in which a particular set of system services are available. The system is always running in one of a set of well-defined run levels. Run levels are also referred to as *init states* because the `init` process is used to perform transitions between run levels. System administrators use the `init` command to initiate a run-level transition. This book refers to init states as run levels.

  For more information about run levels, see "Run Levels" on page 151.
- **Boot types** – A *boot type* describes how a system is booted. Different boot types include the following:
  - Interactive boot – You are prompted to provide information about how the system is booted, such as the kernel and device path name.
  - Reconfiguration boot – The system is reconfigured to support newly added hardware or new pseudo devices.
  - Recovery boot – The system is hung or an invalid entry is prohibiting the system from booting successfully or from allowing users to log in.

Guidelines for Shutting Down a System

Keep the following in mind when you shut down a system:

- Use the `init` and `shutdown` commands to shut down a system. Both commands perform a clean system shutdown, which means that all system processes and services are terminated normally.

- Use the `shutdown` command to shut down a server, because logged-in users and systems that mount resources from the server are notified before the server is shut down. Additional notification of system shutdowns by electronic mail is also recommended so that users can prepare for system downtime.
- You need superuser privileges to use the `shutdown` or `init` command to shut down a system.
- Both `shutdown` and `init` commands take a run level as an argument. The three most common run levels are as follows:
  - Run level 3 – Means that all system resources are available and users can log in. By default, booting a system brings it to run level 3, which is used for normal day-to-day operations. Also known as multiuser level with NFS resources shared.
  - Run level 6 – Stops the operating system and reboots to the state that is defined by the `initdefault` entry in the `/etc/inittab` file.
  - Run level 0 – Means that the operating system is shut down and it is safe to turn off power. You need to bring a system to run level 0 whenever you move a system, or add or remove hardware.

  Run levels are fully described in Chapter 11.

Guidelines for Booting a System

Keep the following in mind when you boot a system:

- After a system is shut down, it is booted by using the `boot` command at the PROM level on a SPARC based system or by using the `boot` command at the Primary Boot Subsystem Menu on an x86 based system.
- A system can be rebooted by turning the power off and then back on. This method is not a clean shutdown because system services and processes are terminated abruptly. However, turning a system's power off and back on is an alternative for emergency situations.
- SPARC based systems and x86 based systems use different hardware components for booting. These differences are described in Chapter 15.

Booting a System From the Network

You might need to boot a system from the network under the following situations:

- When the system is first installed.
- If the system won't boot from the local disk.
- If the system is a diskless client.

In addition, there are two network configuration boot strategies available:

- RARP (Reverse Address Resolution Protocol and ONC+ RPC Bootparams Protocol)
- DHCP (Dynamic Host Configuration Protocol)

The default network boot strategy is set to RARP.

Use this table if you need information on booting a system over the network.

Network Boot Task	For More Information
Boot a SPARC system or a SPARC diskless client	Chapter 13
Boot an x86 system or an x86 diskless client	Chapter 14
Boot a DHCP client during installation	*Solaris 9 Installation Guide*
Configure a DHCP client with DHCP Manager	*System Administration Guide: IP Services*

When to Shut Down a System

The following table provides a list of system administration tasks and the type of shut down that is needed to initiate the task.

TABLE 10–1 Shutting Down a System

Reason for System Shut Down	Appropriate Run Level	For More Information
To turn off system power due to anticipated power outage	Run level 0, where it is safe to turn off power	Chapter 12
To change kernel parameters in the `/etc/system` file	Run level 6 (reboot the system)	Chapter 12

TABLE 10–1 Shutting Down a System *(Continued)*

Reason for System Shut Down	Appropriate Run Level	For More Information
To perform file system maintenance, such as backing up or restoring system data	Run level S (single-user level)	Chapter 12
To repair a system configuration file such as `/etc/system`	See "When to Boot a System" on page 148	N/A
To add or remove hardware from the system	Reconfiguration boot (also to turn off power when adding or removing hardware)	Chapter 27
To repair an important system file which is causing system boot failure	See "When to Boot a System" on page 148	N/A
To boot the kernel debugger (`kadb`) to track down a system problem	Run level 0, if possible	Chapter 12
To recover from a hung system and you want to force a crash dump	See "When to Boot a System" on page 148	N/A

For examples of shutting down a server or a standalone system, see Chapter 12.

When to Boot a System

The following table provides a list of system administration tasks and the corresponding boot type that is used to complete the task.

TABLE 10–2 Booting a System

Reason for System Reboot	Appropriate Boot Type	Information for SPARC Procedure	Information for x86 Procedure
To turn off system power due to anticipated power outage	Turn system power back on	Chapter 12	Chapter 12
To change kernel parameters in the `/etc/system` file	Reboot the system to run level 3 (multiuser level with NFS resources shared)	"SPARC: How to Boot a System to Run Level 3 (Multiuser Level)" on page 181	"x86: How to Boot a System to Run Level 3 (Multiuser Level)" on page 193

TABLE 10–2 Booting a System *(Continued)*

Reason for System Reboot	Appropriate Boot Type	Information for SPARC Procedure	Information for x86 Procedure
To perform file system maintenance, such as performing a backup or restoring system data	Use Control-D from run level S to bring the system back to run level 3	"SPARC: How to Boot a System to Run Level S (Single-User Level)" on page 182	"x86: How to Boot a System to Run Level S (Single-User Level)" on page 194
To repair a system configuration file such as `/etc/system`	Interactive boot	"SPARC: How to Boot a System Interactively" on page 183	"x86: How to Boot a System Interactively" on page 195
To add or remove hardware from the system	Reconfiguration boot (also to turn on system power after adding or removing hardware)	"SPARC: How to Connect a Secondary Disk and Boot" on page 459	"x86: How to Connect a Secondary Disk and Boot" on page 469
To boot the kernel debugger (`kadb`) to track down a system problem	Booting `kabd`	"SPARC: How to Boot the System With the Kernel Debugger (`kadb`)" on page 188	"x86: How to Boot a System With the Kernel Debugger (`kadb`)" on page 203
To repair an important system file that is causing system boot failure	Recovery boot	"SPARC: How to Boot a System for Recovery Purposes" on page 186	"x86: How to Boot a System for Recovery Purposes" on page 198
To recover from a hung system and you want to force a crash dump	Recovery boot	See example on "SPARC: How to Force a Crash Dump and Reboot the System" on page 189	See example on "x86: How to Force a Crash Dump and Reboot the System" on page 204

For examples of booting a system, see Chapter 13 or Chapter 14.

CHAPTER 11

Run Levels and Boot Files (Tasks)

This chapter provides overview information and tasks that are related to run levels and boot files.

This is a list of the step-by-step instructions in this chapter.

- "How to Use a Run Control Script to Stop or Start a Service" on page 161
- "How to Add a Run Control Script" on page 162
- "How to Disable a Run Control Script" on page 162

This is a list of the overview information in this chapter.

- "Run Levels" on page 151
- "The `/etc/inittab` File" on page 153
- "Run Control Scripts" on page 156
- "x86: Boot Files" on page 163

Run Levels

A system's *run level* (also known as an *init state*) defines what services and resources are available to users. A system can be in only one run level at a time.

The Solaris environment has eight run levels, which are described in the following table. The default run level is specified in the `/etc/inittab` file as run level 3.

TABLE 11–1 Solaris Run Levels

Run Level	Init State	Type	Purpose
0	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system.
s or S	Single-user state	Single-user	To run as a single user with some file systems mounted and accessible.
1	Administrative state	Single-user	To access all available file systems. User logins are disabled.
2	Multiuser state	Multiuser	For normal operations. Multiple users can access the system and all file system. All daemons are running except for the NFS server daemons.
3	Multiuser level with NFS resources shared	Multiuser	For normal operations with NFS resources shared. This is the default run level for the Solaris environment.
4	Alternative multiuser state		Currently unavailable.
5	Power-down state	Power-down	To shut down the operating system so that it is safe to turn off power to the system. If possible, automatically turns off power on systems that support this feature.
6	Reboot state	Reboot	To shut down the system to run level 0, and then reboot to multiuser level with NFS resources shared (or whatever level is the default in the `inittab` file).

▼ How to Determine a System's Run Level

Display run level information by using the `who -r` command.

```
$ who -r
```

Use the `who -r` command to determine a system's current run level for any level, except run level 0.

Example—Determining a System's Run Level

This example displays information a system's current run level and information about previous run levels.

```
$ who -r
 .    run-level 3[1]  Dec 13 10:10[2]   3[3]   0[4] S[5]
$
```

1. Identifies the current run level
2. Identifies the date of last run level change
3. Also identifies the current run level
4. Identifies the number of times the system has been at this run level since the last reboot
5. Identifies the previous run level

The /etc/inittab File

When you boot the system or change run levels with the `init` or `shutdown` command, the `init` daemon starts processes by reading information from the `/etc/inittab` file. This file defines three important items for the `init` process:

- The system's default run level
- What processes to start, monitor, and restart if they terminate
- What actions to take when the system enters a new run level

Each entry in the `/etc/inittab` file has the following fields:

id : *rstate* : *action* : *process*

The following table describes the fields in an `inittab` entry.

TABLE 11–2 Fields Descriptions for the `inittab` File

Field	Description
id	Is a unique identifier for the entry.
rstate	Lists the run levels to which this entry applies.

TABLE 11–2 Fields Descriptions for the `inittab` File *(Continued)*

Field	Description
action	Identifies how the process that is specified in the process field is to be run. Possible values include: `initdefault`, `sysinit`, `boot`, `bootwait`, `wait`, and `respawn`. `initdefault` identifies the default run level. For a description of the other action keywords, see `inittab`(4).
process	Defines the command or script to execute.

Example—Default `inittab` File

The following example shows an annotated default `inittab` file that is installed with the Solaris release:

```
1 ap::sysinit:/sbin/autopush -f /etc/iu.ap
2 ap::sysinit:/sbin/soconfig -f /etc/sock2path
3 fs::sysinit:/sbin/rcS sysinit   >/dev/msglog 2<>/dev/msglog </dev/console
4 is:3:initdefault:
5 p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/...
6 sS:s:wait:/sbin/rcS              >/dev/msglog 2<>/dev/msglog </dev/console
7 s0:0:wait:/sbin/rc0              >/dev/msglog 2<>/dev/msglog </dev/console
8 s1:1:respawn:/sbin/rc1           >/dev/msglog 2<>/dev/msglog </dev/console
9 s2:23:wait:/sbin/rc2             >/dev/msglog 2<>/dev/msglog </dev/console
10 s3:3:wait:/sbin/rc3             >/dev/msglog 2<>/dev/msglog </dev/console
11 s5:5:wait:/sbin/rc5             >/dev/msglog 2<>/dev/msglog </dev/console
12 s6:6:wait:/sbin/rc6             >/dev/msglog 2<>/dev/msglog </dev/console
13 fw:0:wait:/sbin/uadmin 2 0      >/dev/msglog 2<>/dev/msglog </dev/console
14 of:5:wait:/sbin/uadmin 2 6      >/dev/msglog 2<>/dev/msglog </dev/console
15 rb:6:wait:/sbin/uadmin 2 1      >/dev/msglog 2<>/dev/msglog </dev/console
16 sc:234:respawn:/usr/lib/saf/sac -t 300
17 co:234:respawn:/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: "
   -T terminal-type -d /dev/console -l console
-m ldterm,ttcompat
```

1. Initializes STREAMS modules
2. Configures socket transport providers
3. Initializes file systems
4. Defines default run level
5. Describes a power fail shutdown
6. Defines single-user level
7. Defines run level 0
8. Defines run level 1
9. Defines run level 2
10. Defines run level 3
11. Defines run level 5
12. Defines run level 6
13. Defines an unused level, firmware

14. Defines an unused level, off
15. Defines an unused level, reboot
16. Initializes Service Access Controller
17. Initializes console and identifies the terminal type

What Happens When the System Is Brought to Run Level 3

1. The `init` process is started and reads the `/etc/default/init` file to set any environment variables. By default, only the `TIMEZONE` variable is set.
2. Then `init` reads the `inittab` file and does the following:
   a. Identifies the `initdefault` entry, which defines the default run level (3).
   b. Executes any process entries that have `sysinit` in the `action` field so that any special initializations can take place before users login.
   c. Executes any process entries that have a 3 in the `rstate` field, which matches the default run level, 3.

      For a detailed description of how the `init` process uses the `inittab` file, see `init`(1M).

The following table describes the keywords used for run level 3's `action` field.

TABLE 11–3 Run Level 3 Action Keyword Descriptions

Key Word	Description
`powerfail`	Starts the process when the `init` process receives a power failure signal
`respawn`	Starts the process and restarts it when it dies
`wait`	Starts the process and waits for it to finish before going on to the next entry for this run level

The following table describes the processes (or commands) that are executed at run level 3.

TABLE 11–4 Command Descriptions for Run Level 3

Command or Script Name	Description
`/usr/sbin/shutdown`	Shuts down the system. The `init` process runs the `shutdown` command only if the system has received a power fail signal.

TABLE 11–4 Command Descriptions for Run Level 3 *(Continued)*

Command or Script Name	Description
`/sbin/rcS`	Checks and mounts root (/), `/usr`, `/tmp`, `/var`, `/var/adm`, and `/var/run` file systems.
`/sbin/rc2`	Starts the standard system processes and brings the system up into run level 2 (multiuser level).
`/sbin/rc3`	Starts NFS resource sharing for run level 3.
`/usr/lib/saf/sac -t 30`	Starts the port monitors. This process is restarted if it fails.
`/usr/lib/saf/ttymon -g -h -p "`uname -n` console login: " -T` *terminal_type* `-d /dev/console -l console`	Starts the `ttymon` process that monitors the console for login requests. This process is restarted if it fails. The *terminal_type* on a SPARC based system is `sun`. The *terminal_type* on an x86 based system is `AT386`.

Run Control Scripts

The Solaris software environment provides a detailed series of run control (`rc`) scripts to control run-level changes. Each run level has an associated `rc` script that is located in the `/sbin` directory:

- `rc0`
- `rc1`
- `rc2`
- `rc3`
- `rc5`
- `rc6`
- `rcS`

For each `rc` script in the `/sbin` directory, there is a corresponding directory named `/etc/rc`*n*`.d` that contains scripts to perform various actions for that run level. For example, `/etc/rc2.d` contains files that are used to start and stop processes for run level 2.

```
# ls /etc/rc2.d
K03samba*           S20sysetup*         S72slpd*            S88utmpd*
K06mipagent*        S21perf*            S73cachefs.daemon*  S89PRESERVE*
K07dmi*             S30sysid.net*       S73nfs.client*      S89bdconfig@
K07snmpdx*          S40llc2*            S74autofs*          S90wbem*
K16apache*          S42ncakmod*         S74syslog*          S91afbinit*
K21dhcp*            S47pppd*            S74xntpd*           S91gfbinit*
K26sshd*            S69inet*            S75cron*            S91ifbinit*
K27boot.server*     S70sckm*            S75flashprom*       S92volmgt*
```

```
K28kdc*             S70uucp*            S75savecore*        S93cacheos.finish*
K28kdc.master*      S71ldap.client*     S76nscd*            S94ncalogd*
K28nfs.server*      S71rpc*             S77sf880dr*         S95IIim*
README              S71sysid.sys*       S80lp*              S95svm.sync*
S01MOUNTFSYS*       S72autoinstall*     S80spc*             S98efcode*
S05RMTMPFILES*      S72directory@       S85power*           S99audit*
S10lu*              S72inetsvc*         S88sendmail*        S99dtlogin*
```

The `/etc/rc`*n*`.d` scripts are always run in ASCII sort order. The scripts have names of the form:

```
[KS][0-9][0-9]*
```

Files that begin with `K` are run to terminate (kill) a system service. Files that begin with `S` are run to start a system service.

Run control scripts are also located in the `/etc/init.d` directory. These files are linked to corresponding run control scripts in the `/etc/rc`*n*`.d` directories.

The actions of each run control script are summarized in the following section.

Run Control Script Summaries

The following sections summarize the run control scripts that are used to start and stop system services when you change run levels.

The `/sbin/rc0` Script

The `/sbin/rc0` script runs the `/etc/rc0.d` scripts to perform the following tasks:

- Stops system services and daemons
- Terminates all running processes
- Unmounts all file systems

The `/sbin/rc1` Script

The `/sbin/rc1` script runs the `/etc/rc1.d` scripts to perform the following tasks:

- Stops system services and daemons
- Terminates all running user processes
- Unmounts all remote file systems
- Mounts all local file systems if the previous run level was S

The `/sbin/rc2` Script

The `/sbin/rc2` script runs the `/etc/rc2.d` scripts to perform the following tasks, grouped by function:

Local system-related tasks:

- Mounts all local file systems if the previous run level was S
- Enables disk quotas if at least one file system was mounted with the `quota` option
- Saves temporary editor files in the `/usr/preserve` directory
- Removes any files and subdirectories in the `/tmp` directory
- Starts system activity data collecting, system accounting, and system auditing, if configured
- Starts the system logging daemon (`syslogd`), sets the default dump device, and rotates the `/var/adm/messages` file
- Sets the default scheduling class if the `/etc/dispadmin.conf` file exists
- Starts LP print service (`lpsched`) if a local printer is configured and cleans up the print queue
- Configures power management, if appropriate
- Starts the `utmpd` daemon
- Starts the `cron` and `vold` daemons
- Configures serial device stream
- Configures WBEM services
- Syncs volumes, if required, and starts the `mdmonitord` daemon to monitor the physical components of the volumes
- Starts the CDE desktop login process, `dtlogin`, if appropriate

Network service or security-related tasks:

- Configures the network interfaces, sets `ifconfig` netmask, and configures network routing, if appropriate
- Starts network service (`inetd` and `rpcbind`) daemons
- Starts the logical link controller (`llc2`), if configured
- Sets the name service domain name, starts various name services daemons, depending on if the system is configured for a name service, and whether the system is a client or a server
- Starts the `keyserv`, `statd`, `lockd`, and `xntpd` daemons, if appropriate
- Mounts all NFS entries
- Configures the Solaris Network Cache and Accelerator (NCA) and NCA logging, if appropriate
- Starts the Solaris PPP server or client daemons (`pppoed` or `pppd`), if configured
- Starts LDAP cache manager (`ldap_cachemgr`), if configured
- Starts directory server (`slapd`) daemon, if configured
- Starts DNS (`in.named`) daemon, if configured
- Starts Service Location Protocol (`slpd`) daemon, if configured

- Configures system resource controls and system pools if the `/etc/rctladm.conf` and `/etc/pooladm.conf` files exist
- Starts the `cachefsd`, `automount`, and `sendmail` daemons, if appropriate
- Starts the `htt_server` process

Install-related tasks:

- Configures the boot environment for the Live Upgrade software upon system startup or system shutdown
- Checks for the presence of the `/etc/.UNCONFIGURE` file to see if the system should be reconfigured
- Reboots the system from the installation media or a boot server if either `/.PREINSTALL` or `/AUTOINSTALL` exists

Hardware-related tasks:

- Starts the Sun Fire 15000 key management daemon (`sckmd`), if appropriate
- Starts the Sun Fire 880 Dynamic Reconfiguration daemon (`sf880drd`), if appropriate
- Runs the flash PROM update script
- Configures any graphic frame buffers or graphic accelerators
- Runs the FCode interpreter daemon (`efdaemon`), if necessary

Transitions the following services between run-level changes:

- Apache (`tomcat`)
- Boot server (`in.rarpd`), (`rpc.bootparamd`), or (`rpld`)
- DHCP (`in.dhcpd`)
- Kerberos KDC (`krb5kdc`) and Kerberos administration (`kadmind`)
- Mobile IP (`mipagent`)
- NFS server (`nfsd`), (`mountd`), (`nfslogd`), if appropriate
- Samba (`smdb`) and (`nmdb`)
- Secure shell (`sshd`)
- Solstice Enterprise Agents (`dmispd`) and (`snmpXdmid`)

Note – Many of the system services and applications that are started at run level 2 depend on what software is installed on the system.

The `/sbin/rc3` Script

The `/sbin/rc3` script runs the `/etc/rc3.d` scripts to perform the following tasks:

- Starts the Apache server daemon (`tomcat`), if configured
- Starts the DHCP daemon (`in.dhcpd`), if appropriate

- Starts Kerberos KDC (`krb5kdc`) and Kerberos administration (`kadmind`) daemons, if configured
- Starts Mobile IP daemon (`mipagent`), if configured
- Starts the Samba daemons (`smdb` and `nmdb`), if configured
- Starts the secure shell daemon (`sshd`), if appropriate
- Starts the Solstice Enterprise Agents (`dmispd` and `snmpXdmid`)
- Cleans up the `/etc/dfs/sharetab` file
- Starts the NFS server daemons `nfsd`, `mountd`, and `nfslogd`, if appropriate
- If the system is a boot server, starts the `rarpd`, `rpc.bootparamd`, and `rpld` daemons

The `/sbin/rc5` and `/sbin/rc6` Scripts

The `/sbin/rc5` and `/sbin/rc6` scripts run the `/etc/rc0.d/K*` scripts to perform the following tasks:

- Kills all active processes
- Unmounts the file systems

The `/sbin/rcS` Script

The `/sbin/rcS` script runs the `/etc/rcS.d` scripts to bring the system up to run level S. The following tasks are performed by these scripts:

- Establishes a minimal network
- Checks and mounts root (`/`), `/usr`, `/tmp`, `/var`, `/var/adm`, and `/var/run` file systems.
- Sets the system name
- Mounts pseudo file systems (`/proc` and `/dev/fd`)
- Rebuilds the device entries for reconfiguration boots
- Checks and mounts other file systems to be mounted in single-user level

Using a Run Control Script to Stop or Start Services

One advantage of having individual scripts for each run level is that you can run scripts in the `/etc/init.d` directory individually to stop system services without changing a system's run level.

▼ How to Use a Run Control Script to Stop or Start a Service

1. **Become superuser.**
2. **Stop the system service.**

   # **/etc/init.d/***filename* **stop**
3. **Restart the system service.**

   # **/etc/init.d/***filename* **start**
4. **Verify that the service has been stopped or started.**

   # **pgrep -f** *service*

Example—Using a Run Control Script to Stop or Start a Service

For example, you can stop the NFS server daemons by typing the following:

```
# /etc/init.d/nfs.server stop
# pgrep -f nfs
#
```

Then, you can restart the NFS server daemons by typing the following:

```
# /etc/init.d/nfs.server start
# pgrep -f nfs
341
343
347
345
# pgrep -f nfs -d, | xargs ps -fp
 UID   PID  PPID  C    STIME TTY      TIME CMD
  daemon   341     1  0   Aug 21 ?        0:00 /usr/lib/nfs/statd
    root   343     1  0   Aug 21 ?        0:00 /usr/lib/nfs/lockd
    root   347     1  0   Aug 21 ?        0:41 /usr/lib/nfs/nfsd
    root   345     1  0   Aug 21 ?        0:02 /usr/lib/nfs/mountd
```

Adding a Run Control Script

If you want to add a run control script to start and stop a service, copy the script into the `/etc/init.d` directory. Then, create links in the `rc`*n*`.d` directory where you want the service to start and stop.

See the `README` file in each `/etc/rc`*n*`.d` directory for more information on naming run control scripts. The following procedure describes how to add a run control script.

▼ How to Add a Run Control Script

1. **Become superuser.**
2. **Add the script to the `/etc/init.d` directory.**

   ```
   # cp filename /etc/init.d
   # chmod 0744 /etc/init.d/filename
   # chown root:sys /etc/init.d/filename
   ```

3. **Create links to the appropriate `rc`*n*`.d` directory.**

   ```
   # cd /etc/init.d
   # ln filename /etc/rc2.d/Snnfilename
   # ln filename /etc/rcn.d/Knnfilename
   ```

4. **Verify that the script has links in the specified directories.**

   ```
   # ls /etc/init.d/ /etc/rc2.d/ /etc/rcn.d/
   ```

Example—Adding a Run Control Script

The following example shows how to add a run control script for the `xyz` service.

```
# cp xyz /etc/init.d
# chmod 0744 /etc/init.d/xyz
# chown root:sys /etc/init.d/xyz
# cd /etc/init.d
# ln xyz /etc/rc2.d/S100xyz
# ln xyz /etc/rc0.d/K100xyz
# ls /etc/init.d /etc/rc2.d /etc/rc0.d
```

Disabling a Run Control Script

You can disable a run control script by renaming it with an underscore (_) at the beginning of the file name. Files that begin with an underscore or dot are not executed. If you copy a file by adding a suffix to it, both files will be run.

▼ How to Disable a Run Control Script

1. **Become superuser.**
2. **Rename the script by adding an underscore (_) to the beginning of the new file.**

   ```
   # cd /etc/rcn.d
   # mv filename _filename
   ```

3. **Verify that the script has been renamed.**

```
# ls _*
# _filename
```

Example—Disabling a Run Control Script

The following example shows how to rename the `S100datainit` script.

```
# cd /etc/rc2.d
# mv S100datainit _S100datainit
# ls _*
# _S100datainit
```

x86: Boot Files

In addition to the run control scripts and boot files described previously, there are additional boot files that are associated with booting a Solaris x86 system.

TABLE 11–5 x86: Boot Files

File	Description
`/etc/bootrc`	Contains menus and options for booting the Solaris release.
`/boot`	Contains files and directories needed to boot the system.
`mdboot`	DOS executable that loads the first-level bootstrap program (`strap.com`) into memory from disk.
`mdbootbp`	DOS executable that loads the first-level bootstrap program (`strap.com`) into memory from diskette.
`rc.d`	Directory that contains install scripts. Do not modify the contents of this directory.
`solaris`	Directory that contains items for the boot subsystem.
`solaris/boot.bin`	Loads the Solaris kernel or standalone `kadb`. In addition, this executable provides some boot firmware services.
`solaris/boot.rc`	Identifies the Solaris x86 Platform Edition and runs the Device Configuration Assistant in DOS-emulation mode.
`solaris/bootconf.exe`	DOS executable for the Device Configuration Assistant.
`solaris/bootconf.txt`	Text file that contains internationalized messages for Device Configuration Assistant (`bootconf.exe`).

TABLE 11–5 x86: Boot Files *(Continued)*

File	Description
`solaris/bootenv.rc`	Contains variables that are used to set up the boot environment.
`solaris/devicedb`	Directory that contains the `master` file, a database of all possible devices supported with realmode drivers.
`solaris/drivers`	Directory that contains realmode drivers.
`solaris/itup2.exe`	DOS executable run during install time update (ITU) process.
`solaris/machines`	Obsolete directory.
`solaris/nbp`	File associated with network booting.
`solaris/strap.rc`	File that contains instructions on what load module to load and where in memory it should be loaded.
`strap.com`	DOS executable that loads the second-level bootstrap program into memory.

CHAPTER 12

Shutting Down a System (Tasks)

This chapter describes the procedures for shutting down systems. This is a list of the step-by-step instructions in this chapter.

- "How to Determine Who Is Logged in to a System" on page 167
- "How to Shut Down a Server" on page 167
- "How to Shut Down a Standalone System" on page 171
- "How to Turn Off Power to All Devices" on page 173

This is a list of the overview information in this chapter.

- "System Shutdown Commands" on page 166
- "User Notification of System Down Time" on page 167
- "Turning Off Power to All Devices" on page 172

For overview information about system run levels, see Chapter 11.

Shutting Down the System

Solaris software is designed to run continuously so that the electronic mail and network software can work correctly. However, some system administration tasks and emergency situations require that the system is shut down to a level where it is safe to remove power. In some cases, the system needs to be brought to an intermediate level, where not all system services are available, such as the following:

- Adding or removing hardware
- Preparing for an expected power outage
- Performing file system maintenance, such as a backup

For a complete list of system administration tasks that require a system shutdown, see Chapter 10.

For information on using your system's power management features, see *Solaris Common Desktop Environment: User's Guide*.

System Shutdown Commands

The use of the `init` and `shutdown` commands are the primary ways to shut down a system. Both commands perform a *clean shutdown* of the system, which means that all file system changes are written to the disk, and all system services, processes, and the operating system are terminated normally.

The use of a system's stop key sequence or turning a system off and then on are not clean shutdowns because system services are terminated abruptly. However, it is sometimes necessary to use these actions in emergency situations. For instructions on system recovery techniques, see Chapter 13 or Chapter 14.

The following table describes the various shutdown commands and provides recommendations for using them.

TABLE 12–1 Shutdown Commands

Command	Description	When To Use
`shutdown`	An executable shell script that calls the `init` program to shut down the system. The system is brought to run level S by default.	Recommended for servers running at run level 3 because users are notified of the impending shut down. Also notified are the systems that are mounting resources from the server that is being shut down.
`init`	An executable that kills all active processes and syncs the disks before changing run levels.	Recommended for standalone systems when other users will not be affected. Provides a faster system shutdown because users are not notified of the impending shutdown.
`reboot`	An executable that syncs the disks and passes boot instructions to the `uadmin` system call, which, in turn, stops the processor.	Not recommended. Use the `init` command instead.
`halt`	An executable that syncs the disks and stops the processor.	Not recommended because it doesn't execute the `/etc/rc0` script. This script stops all processes, syncs the disks, and unmounts any remaining file systems.

User Notification of System Down Time

When the `shutdown` command is initiated, a warning followed by a final shutdown message is broadcast to all users who are currently logged onto the system and all systems that are mounting resources from the affected system.

For this reason, the `shutdown` command is recommended over the `init` command when you need to shut down a server. When you use either command, you might want to give users more notice by sending them a mail message about any scheduled system shutdown.

Use the `who`(1) command to determine which users on the system need to be notified. This command is also useful for determining a system's current run level. See "How to Determine a System's Run Level" on page 152.

▼ How to Determine Who Is Logged in to a System

1. **Log into the system to be shut down.**
2. **Display logged-in users.**

   ```
   $ who
   ```

Example—Determining Who Is Logged in to a System

The following example shows how to display who is logged in to the system.

```
$ who
holly [1]     console      May  7 07:30
kryten      pts/0  [2]     May  7 07:35      (starbug) [4]
lister      pts/1        May  7 07:40 [3] (bluemidget)
```

1. Identifies the user name of the logged-in user.
2. Identifies the terminal line of the logged-in user.
3. Identifies the date and time that the user logged in.
4. (Optional) Identifies the host name if a user is logged in from a remote system.

▼ How to Shut Down a Server

Use this procedure when you need to shut down a server.

1. **Become superuser.**
2. **Find out if users are logged in to the system.**

   ```
   # who
   ```

A list of all logged-in users is displayed. You might want to send mail or broadcast a message to let users know that the system is being shut down.

3. **Shut down the system.**

```
# shutdown -iinit-level -ggrace-period -y
```

`-i`*init-level*	Brings the system to an init level that is different from the default of S. The choices are 0, 1, 2, 5, and 6.
`-g`*grace-period*	Indicates a time (in seconds) before the system is shut down. The default is 60 seconds.
`-y`	Continues to shut down the system without intervention. Otherwise, you are prompted to continue the shutdown process after 60 seconds.

For more information, see `shutdown`(1M).

4. **If you are asked for confirmation, type `y`.**

```
Do you want to continue? (y or n): y
```

If you used the `shutdown -y` command, you will not be prompted to continue.

5. **Type the superuser password, if prompted.**

```
Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxx
```

6. **After you have finished the system administration tasks, press Control-D to return to the default system run level.**

7. **Use the following table to verify that the system is at the run level that you specified in the `shutdown` command.**

Specified Run Level	SPARC System Prompt	x86 System Prompt
S (single-user level)	#	#
0 (power-down level)	`ok` or `>`	`type any key to continue`
Run level 3 (multiuser level with remote resources shared)	*hostname* `console login:`	*hostname* `console login:`

SPARC: Example—Bringing a Server to Run Level S

In the following example, the `shutdown` is used to bring a SPARC based system to run level S (single-user level) in 3 minutes.

```
# who
root       console      Dec 13 14:30
# shutdown -g180 -y

Shutdown started.    Thu Dec 13 14:30:32 MST 2001

Broadcast Message from root (console) on earth Thu Dec 13 14:30:33...
The system earth will be shut down in 3 minutes
.
.
.
Broadcast Message from root (console) on earth Thu Dec 13 14:30:33...
The system earth will be shut down in 30 seconds
.
.
.
INIT: New run level: S
The system is coming down for administration.  Please wait.
Unmounting remote filesystems: /vol nfs done.
Shutting down Solaris Management Console server on port 898.
Print services stopped.
Dec 13 14:34:00 earth syslogd: going down on signal 15
Killing user processes: done.

INIT: SINGLE USER MODE

Type control-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Entering System Maintenance Mode ...
#
```

SPARC: Example—Bringing a Server to Run Level 0

In the following example, the `shutdown` command is used to bring a SPARC based system to run level 0 in 5 minutes without requiring additional confirmation.

```
# who
root       console      Dec 12 08:08
rimmer     pts/0        Dec 11 14:48    (starbug)
pmorph     pts/1        Dec 13 12:31    (bluemidget)
# shutdown -i0 -g300 -y
Shutdown started.    Thu Dec 13 14:51:39 MST 2001

Broadcast Message from root (console) on earth Thu Dec 13 14:51:39...
The system earth will be shut down in 5 minutes
.
.
.
Changing to init state 0 - please wait
#
INIT: New run level: 0
The system is coming down.  Please wait.
System services are now being stopped.
```

```
.
.
.
The system is down.
syncing file systems... done
Program terminated
Type  help  for more information
ok
```

If you are bringing the system to run level 0 to turn off power to all devices, see "How to Turn Off Power to All Devices" on page 173.

SPARC: Example—Rebooting a Server to Run Level 3

In the following example, the `shutdown` command is used to reboot a SPARC based system to run level 3 in two minutes without requiring additional confirmation.

```
# who
root       console      Dec 12 08:08
rimmer     pts/0        Dec 11 14:48    (starbug)
pmorph     pts/1        Dec 13 12:31    (bluemidget)
# shutdown -i6 -g120 -y
Shutdown started.    Thu Dec 13 15:56:30

Broadcast Message from root (console) on earth Thu Dec 13 15:56:30...
The system earth will be shut down in 2 minutes
.
.
.
Changing to init state 6 - please wait
#
INIT: New run level: 6
The system is coming down.  Please wait.
.
.
.
The system is down.
syncing file systems... done
rebooting...
.
.
.
earth console login:
```

Where to Go From Here

Regardless of why you shut down a system, you'll probably want to return to run level 3 where all file resources are available and users can log in. For instructions on bringing a system back to a multiuser level, see Chapter 13 or Chapter 14.

▼ How to Shut Down a Standalone System

Use this procedure when you need to shut down a standalone system.

1. **Become superuser.**
2. **Shut down the system.**

   `# init` *run-level*

   run-level identifies the new run level.

   For more information, see `init`(1M).

3. **Use the following table to verify that the system is at the run level that you specified in the `init` command.**

Specified Run Level	SPARC System Prompt	x86 System Prompt
S (single-user level)	#	#
2 (multiuser level)	#	#
0 (power-down level)	`ok` or `>`	`type any key to continue`
3 (multiuser level with NFS resources shared)	*hostname* `console login:`	*hostname* `console login:`

x86: Example—Bringing a Standalone System to Run Level 0

In the following example, the `init` command is used to bring an x86 based standalone system to the level where it is safe to turn off power.

```
# init 0
#
INIT: New run level: 0
The system is coming down.  Please wait.
.

.

.
The system is down.
syncing file systems... [11] [10] [3] done
Type any key to continue
```

If you are bringing the system to run level 0 to turn off power to all devices, see "How to Turn Off Power to All Devices" on page 173.

SPARC: Example—Bringing a Standalone System to Run Level S

In the following example, the `init` is used to bring a SPARC based standalone system to run level S (single-user level).

```
# init s
#
INIT: New run level: S
The system is coming down for administration.  Please wait.
Unmounting remote filesystems: /vol nfs done.
Print services stopped.
syslogd: going down on signal 15
Killing user processes: done.
INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Entering System Maintenance Mode
#
```

Where to Go From Here

Regardless of why you shut down the system, you'll probably want to return to run level 3 where all file resources are available and users can log in. For instructions on bringing a system back to a multiuser level, see Chapter 13 or Chapter 14.

Turning Off Power to All Devices

You need turn off power to all system devices is when you do the following:

- Replace or add hardware
- Move the system from one location to another
- Prepare for an expected power outage or natural disaster like an approaching electrical storm

System devices to power down include the CPU, the monitor, and external devices such as disks, tapes, and printers.

Before you turn off power to all system devices, you should shutdown the system cleanly, as described in the preceding sections.

▼ How to Turn Off Power to All Devices

1. **Select one of the following to shut down the system.**
   a. **If shutting down a server, see "How to Shut Down a Server" on page 167.**
   b. **If shutting down a standalone system, see "How to Shut Down a Standalone System" on page 171.**
2. **Turn off the power to all devices after the system is shutdown. If necessary, also unplug the power cables.**
3. **After power can be restored, use the following steps to turn on the system and devices.**
   a. **Plug in the power cables.**
   b. **Turn on the monitor.**
   c. **Turn on disk drives, tape drives, and printers.**
   d. **Turn on the CPU.**
      The system is brought to run level 3.

CHAPTER 13

SPARC: Booting a System (Tasks)

This chapter describes the procedures for using the OpenBoot™ PROM monitor and the procedures for booting a SPARC based system to different run levels.

For information on the procedures associated with booting a SPARC system, see "SPARC: Booting a System (Task Map)" on page 175.

For overview information about the boot process, see Chapter 10. To troubleshoot boot problems, see "What to Do If Rebooting Fails" in *System Administration Guide: Advanced Administration*.

For step-by-step instructions on booting an x86 based system, see Chapter 14.

SPARC: Booting a System (Task Map)

Task	Description	For Instructions
Use the Boot PROM	The boot PROM is used to boot a system. You might need to do one of the following:	
	Identify the PROM revision number.	"SPARC: How to Find the PROM Revision for a System" on page 177
	Identify devices on the system to boot from.	"SPARC: How to Identify Devices on the System" on page 177

Task	Description	For Instructions
	Change the default boot device when a new disk is added or when you need to change the system boot method.	"SPARC: How to Change the Default Boot Device" on page 179
Boot the system	Select one of the following boot methods:	
	Boot to run level 3 - Used after shutting down the system or performing some system hardware maintenance task.	"SPARC: How to Boot a System to Run Level 3 (Multiuser Level)" on page 181
	Boot to run level S - Used after performing some system maintenance task such as backing up a file system. At this level, only local file systems are mounted and users cannot log into the system.	"SPARC: How to Boot a System to Run Level S (Single-User Level)" on page 182
	Boot interactively - Used after making temporary changes to a system file or the kernel for testing purposes.	"SPARC: How to Boot a System Interactively" on page 183
	Boot from the network - Used to boot a system from the network. This method is used for booting a diskless client.	"SPARC: How to Boot a System From the Network" on page 184
	Boot for recovery purposes - Used to boot the system when a damaged file or file system is preventing the system from booting. You might need to do one or both of the following to boot for recovery purposes:	
	First, stop the system to attempt recovery.	"SPARC: How to Stop the System for Recovery Purposes" on page 186
	Boot to repair an important system file that is preventing the system from booting successfully.	"SPARC: How to Boot a System for Recovery Purposes" on page 186
	Boot `kadb` - Used to troubleshoot system problems.	"SPARC: How to Stop the System for Recovery Purposes" on page 186

Task	Description	For Instructions
	Force a crash dump and reboot the system - Used to force a crash dump for troubleshooting purposes.	"SPARC: How to Force a Crash Dump and Reboot the System" on page 189

SPARC: Using the Boot PROM

System administrators typically use the PROM level to boot a system. Occasionally, however, you might need to change the way the system boots. For example, you might want to reset the device to boot from or run hardware diagnostics before you bring the system to a multiuser level.

You need to change the default boot device to do the following:

- Add a new drive to the system either permanently or temporarily
- Change the network boot strategy
- Temporarily boot a standalone system from the network

For a complete list of PROM commands, see `monitor`(1M) or `eeprom`(1M).

▼ SPARC: How to Find the PROM Revision for a System

Display a system's PROM revision level with the `banner` command.

```
ok banner
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.
```

Hardware configuration information, including the revision number of the PROM, is displayed. In this example, the PROM revision number is 3.15.

▼ SPARC: How to Identify Devices on the System

You might need to identify the devices on the system to figure out what are the appropriate devices to boot from.

Before you can safely use the `probe` commands to find out what devices are attached to the system, you need to do the following:

- Change the PROM `auto-boot?` parameter to false and

- Issue the `reset-all` command to clear system registers

If you run the `probe` commands without clearing the system registers, the following message is displayed:

```
ok probe-scsi
This command may hang the system if a Stop-A or halt command
has been executed.  Please type reset-all to reset the system
before executing this command.
Do you wish to continue? (y/n) n
```

1. **Change the PROM auto-boot? parameter to false.**

```
ok setenv auto-boot? false
```

2. **Clear the system's registers.**

```
ok reset-all
```

3. **Identify the devices on the system.**

```
ok probe-device
```

Example—Identifying the Default Boot Device

The following example shows how to identify the devices connected to an Ultra10 system.

```
ok setenv auto-boot? false
auto-boot? =          false
ok reset-all
Resetting ...

Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #10933339.
Ethernet address 8:0:20:a6:d4:5b, Host ID: 80a6d45b.

ok probe-ide
  Device 0  ( Primary Master )
          ATA Model: ST34321A

  Device 1  ( Primary Slave )
         Not Present

  Device 2  ( Secondary Master )
         Removable ATAPI Model: CRD-8322B

  Device 3  ( Secondary Slave )
         Not Present

ok
```

▼ SPARC: How to Change the Default Boot Device

1. **Become superuser.**
2. **Change to run level 0.**

   ```
   # init 0
   ```

   The `ok` PROM prompt is displayed.

   For more information, see `init`(1M).

3. **Change the value of the `boot-device` parameter.**

   ```
   ok setenv boot-device device[n]
   ```

`boot-device`	Identifies the parameter for setting the device from which to boot.
device[*n*]	Identifies the `boot-device` value such as a disk or the network. The *n* can be specified as the *disk number*.

   Use the `probe-scsi-all` command if you need help with identifying the disk number.

4. **Verify that the default boot device is changed.**

   ```
   ok printenv boot-device
   ```

5. **Save the new `boot-device` value.**

   ```
   ok reset
   ```

   The new `boot-device` value is written to the PROM.

SPARC: Examples—Changing the Default Boot Device

In this example, the default boot device is set to disk.

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device disk
boot-device =         disk
ok printenv boot-device
boot-device           disk                    disk
ok reset
```

```
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.

Boot device: disk  File and args:
SunOS Release 5.9 Version 64-bit
.
.
.
pluto console login:
```

In this example, the default boot device is set to the network.

```
# init 0
#
INIT: New run level: 0
.
.
.
The system is down.
syncing file systems... done
Program terminated
ok setenv boot-device net
boot-device =         net
ok printenv boot-device
boot-device           net                     disk
ok reset
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz), No Keyboard
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.



Boot device: net  File and args:
.
.
.
pluto console login:
```

▼ SPARC: How to Reset the System

Run the `reset` command from the `ok` prompt.

```
ok reset
```

This self-test program, which runs diagnostic tests on the hardware, is executed and the system is rebooted.

SPARC: Booting a System

If a system is turned off, turning it on starts the multiuser boot sequence. The following procedures show how to boot to different run levels from the `ok` PROM prompt. These procedures assume that the system has been cleanly shut down, unless stated otherwise.

Use the `who -r` command to verify that the system is brought to the specified run level. For a description of run levels, see Chapter 11.

▼ SPARC: How to Boot a System to Run Level 3 (Multiuser Level)

Use this procedure to boot a system that is currently at run level 0 to run level 3.

1. **Boot the system to run level 3.**

   ```
   ok boot
   ```

   The automatic boot procedure displays a series of startup messages, and brings the system to run level 3.

   For more information, see `boot`(1M).

2. **Verify that the system has booted to run level 3.**

   The login prompt is displayed when the boot process has finished successfully.

   hostname `console login:`

SPARC: Example—Booting a System to Run Level 3 (Multiuser Level)

The following example displays the messages from booting a system to run level 3.

```
ok boot
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
OpenBoot 3.15, 128 MB memory installed, Serial #number.
Ethernet address number, Host ID: number.

Rebooting with command: boot
Boot device: disk:a  File and args:
SunOS Release 5.9 Version Generic 64-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
configuring IPv4 interfaces: hme0.
Hostname: starbug
The system is coming up.  Please wait.
```

```
checking ufs filesystems
/dev/rdsk/c0t0d0s7: is clean.
/dev/rdsk/c0t0d0s4: is clean.
NIS domainname is Solar.COM
starting rpc services: rpcbind keyserv ypbind done.
Setting netmask of hme0 to 255.255.255.0
Setting default IPv4 interface for multicast: add net 224.0/4:
gateway starbug
syslog service starting.
Print services started.
volume management starting.
The system is ready.

starbug console login:
```

▼ SPARC: How to Boot a System to Run Level S (Single-User Level)

Use this procedure to boot a system that is currently at run level 0 to run level S.

1. **Boot the system to run level S.**

   ```
   ok boot -s
   ```

2. **Type the superuser password when the following message is displayed.**

   ```
   INIT: SINGLE USER MODE
   Type Ctrl-d to proceed with normal startup,

   (or give root password for system maintenance):  xxx
   ```

3. **Verify that the system is at run level S.**

   ```
   # who -r
   .       run-level S  Jun 10 15:27     3      0
   ```

4. **To bring the system up to multiuser state after you completed the system maintenance task, press Control-D.**

SPARC: Example—Booting a System to Run Level S (Single-User Level)

The following example displays the messages from booting a system to run level S.

```
ok boot -s
.
.
.
SunOS Release 5.9 Version Generic 64-bit
```

```
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
configuring IPv4 interfaces: le0.
Hostname: earth

INIT: SINGLE USER MODE

Type control-d to proceed with normal startup,
(or give root password for system maintenance): xxx
Sun Microsystems Inc.   SunOS 5.9  Generic May 2002
# who -r
   .       run-level S  Jul 14 11:37     3      0  ?
(Perform some maintenance task)
#  Press Control-D
```

▼ SPARC: How to Boot a System Interactively

Use this procedure to boot a system and you need to specify an alternate kernel or `/etc/system` file.

1. **Boot the system interactively.**

   ```
   ok boot -a
   ```

2. **Answer the system prompts as described in the following table.**

System Prompt	Action
`Enter filename [kernel/[sparcv9]/unix]:`	Provide the name of kernel to use for booting. Or, press Return to use the default kernel.
`Enter default directory for modules [/platform/‘uname -i‘/kernel /platform/‘uname -m/kernel /kernel /usr/kernel]:`	Provide an alternate path for the `modules` directory. Or, press Return to use the default kernel modules directory.
`Name of system file [etc/system]:`	Provide the name of an alternate system file and press Return. Type `/dev/null` if your `/etc/system` file has been damaged. Or, press Return to use the default `etc/system` file.
`root filesystem type [ufs]:`	Press Return to use the default root (`/`) file system. Type UFS for local disk booting, or NFS for network booting.
`Enter physical name of root device` [*physical_device_name*]:	Provide an alternate device name and press Return. Or, press Return to use the default physical name of the root device.

3. **If you are not prompted to answer the questions in the preceding table, verify that you typed the `boot -a` command correctly.**

SPARC: Example—Booting a System Interactively

In the following example, the default choices (shown in square brackets `[]`) are accepted.

```
ok boot -a
.
.
.
Rebooting with command: boot -a
Boot device: /pci@1f,0/pci@1,1/ide@3/disk@0,0:a  File and args: -a
Enter filename [kernel/sparcv9/unix]: Press Return
Enter default directory for modules [/platform/SUNW,Ultra-5_10/kernel
/platform/sun4u/kernel /kernel /usr/kernel]: Press Return
Name of system file [etc/system]: Press Return
SunOS Release 5.9 Version Generic 64-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
root filesystem type [ufs]: Press Return
Enter physical name of root device
[/pci@1f,0/pci@1,1/ide@3/disk@0,0:a]: Press Return
configuring IPv4 interfaces: hme0.
Hostname: starbug
The system is coming up.  Please wait.
checking ufs filesystems
.
.
.
The system is ready.
starbug console login:
```

▼ SPARC: How to Boot a System From the Network

Any system can boot from the network if there is a boot server available. You might want to boot a standalone system from the network temporarily if the system cannot boot from the local disk. For information on changing or resetting the default boot device, see "SPARC: How to Change the Default Boot Device" on page 179.

Two network configuration boot strategies are available on sun4u systems:

- RARP (Reverse Address Resolution Protocol and ONC+ RPC Bootparams Protocol)
- DHCP (Dynamic Host Configuration Protocol)

The default network boot strategy is set to RARP. You can use either strategy depending on whether a RARP boot server or a DHCP boot server is available on your network.

Note – Sun Ultra systems must have PROM version 3.25.*nn* or later to use the DHCP network boot strategy. For information on finding your PROM version, see "SPARC: How to Find the PROM Revision for a System" on page 177.

If both methods are available, you can specify which service to use in the `boot` command temporarily. Or, you can save the network boot strategy across system reboots at the PROM level, by setting up an NVRAM alias. The following example uses the `nvalias` command to set up a network device alias for booting DHCP by default on a Sun Ultra 10 system.

```
ok nvalias net     /pci@1f,4000/network@1,1:dhcp
```

This alias means that when you type `boot net`, the system boots by using the DHCP network book strategy.

Caution – You should not use the `nvalias` command to modify the `NVRAMRC` file unless you are very familiar with the syntax of this command and the `nvunalias` command. For information on using these commands, see the *OpenBoot 3.x Command Reference Manual*.

1. **If necessary, shut down the system.**
2. **Determine the method for booting from the network and select one of the following:**

   You must have already set up a RARP or DHCP boot server in your network to use either method to boot successfully.

   a. **Boot the system from the network by using the DHCP method.**

      ```
      ok boot net[:dhcp]
      ```

      If you have changed the PROM setting to boot DHCP by default, as in the preceding `nvalias` example, you only have to specify `boot net`.

   b. **Boot the system from the network by using the RARP method.**

      ```
      ok boot net[:rarp]
      ```

      Since RARP is the default network boot strategy, you only have to specify `boot net:rarp` if you have changed the PROM value to boot DHCP.

▼ SPARC: How to Stop the System for Recovery Purposes

1. **Type the stop key sequence for your system.**

   The monitor displays the `ok` PROM prompt.

   ```
   ok
   ```

   The specific stop key sequence depends on your keyboard type. For example, you can press Stop-A or L1-A. On terminals, press the Break key.

2. **Synchronize the file systems.**

   ```
   ok sync
   ```

3. **When you see the `syncing file systems...` message, press the stop key sequence for your system again.**

4. **Type the appropriate `boot` command to start the boot process.**

   For more information, see `boot`(1M).

5. **Verify that the system is booted to the specified run level.**

   ```
   # who -r
    .       run-level 3  May  2 07:39     3      0  S
   ```

SPARC: Example—Stopping the System for Recovery Purposes

```
Press Stop-A
ok sync
syncing file systems...
Press Stop-A
ok boot
```

▼ SPARC: How to Boot a System for Recovery Purposes

Use this procedure when an important file, such as `/etc/passwd`, has an invalid entry and causes the boot process to fail.

Substitute the device name of the file system to be repaired for the *devicename* variable in the following procedures. If you need help identifying a system's device names, refer to Chapter 30.

1. **Stop the system by using the system's stop key sequence.**

   Use the stop sequence for your system if you don't know the root password or if you can't log in to the system. For more information, see "SPARC: How to Stop the System

for Recovery Purposes" on page 186.

2. **Follow the instructions in the table, depending on whether you are booting from the Solaris installation CD or DVD or from the network.**

Boot Type	Action
Solaris installation CD or DVD	1. Insert the Solaris installation media into the drive. 2. Boot from the installation media in single-user mode: `ok boot cdrom -s`
The network if an installation server or remote CD or DVD drive are available	Use the following command: `ok boot net -s`

3. **Mount the file system that contains the file with an invalid entry.**

   # **mount /dev/dsk/***device-name* **/a**

4. **Change to the newly mounted file system.**

   # **cd /a/***file-system*

5. **Set the terminal type.**

   # **TERM=***sun*
   # **export TERM**

6. **Remove the invalid entry from the file by using an editor.**

   # **vi** *filename*

7. **Change to the root (/) directory.**

   # **cd /**

8. **Unmount the /a directory.**

   # **umount /a**

9. **Reboot the system.**

   # **init 6**

10. **Verify that the system booted to run level 3.**

    The login prompt is displayed when the boot process has finished successfully.

    hostname `console login:`

SPARC: Example—Booting a System for Recovery Purposes (Damaged Password File)

The following example shows how to repair an important system file (in this case, `/etc/passwd`) after booting from a local CD-ROM.

```
ok boot cdrom -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi passwd
(Remove invalid entry)
# cd /
# umount /a
# init 6
```

SPARC: Example—Booting a System if You Forgot Root Password

The following example shows how to recover when you forget the root password by booting from the network. This example assumes that the network boot server is already available. Be sure to apply a new root password after the system has rebooted.

```
ok boot net -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi shadow
(Remove root's encrypted password string)
# cd /
# umount /a
# init 6
```

▼ SPARC: How to Boot the System With the Kernel Debugger (`kadb`)

1. **If you need to stop the system, type the stop key sequence for your system.**

   The specific stop key sequence depends on your keyboard type. For example, you can press `Stop-A` or `L1-A`. On terminals, press the `Break` key.

   The PROM displays the `ok` prompt.

2. **Synchronize the file systems and write the crash dump.**

   ```
   > n
   ok sync
   ```

3. **When you see the `syncing file systems...` message, press the stop key sequence for your system again.**

4. **Boot the system with the kernel debugger.**

```
ok boot kadb
```

5. **Check `kadb` boot messages to verify that the system has booted with the kernel debugger.**

```
Rebooting with command: kadb
Boot device: /iommu/sbus/espdma@4,800000/esp@4,8800000/sd@3,0
.
.
.
```

SPARC: Example—Booting the System With the Kernel Debugger (`kadb`)

```
Press Stop-A
ok sync
syncing file systems...
Press Stop-A
ok boot kadb
```

SPARC: Forcing a Crash Dump and Rebooting the System

Forcing a crash dump and rebooting the system is sometimes necessary for troubleshooting purposes. The `savecore` feature is enabled by default.

For more information on system crash dumps, see "Managing System Crash Information (Tasks)" in *System Administration Guide: Advanced Administration*.

▼ SPARC: How to Force a Crash Dump and Reboot the System

Use this procedure to force a crash dump and reboot the system when the `savecore` feature is enabled.

1. **Type the stop key sequence for your system.**

The specific stop key sequence depends on your keyboard type. For example, you can press `Stop-a` or `L1-a`. On terminals, press the `Break` key.

The PROM displays the `ok` prompt.

2. **Synchronize the file systems and write the crash dump.**

```
> n
ok sync
```

After the crash dump is written to disk, the system will continue to reboot.

3. **Verify the system boots to run level 3.**

The login prompt is displayed when the boot process has finished successfully.

hostname `console login:`

SPARC: Example—Forcing a Crash Dump and Rebooting the System

Press Stop-A

```
ok sync
```

CHAPTER 14

x86: Booting a System (Tasks)

This chapter describes the procedures for booting an x86 based system.

For information on the procedures associated with booting an x86 system, see "x86: Booting a System (Task Map)" on page 191.

For overview information about the boot process, see Chapter 10.

For step-by-step instructions on booting a SPARC based system, see Chapter 13.

x86: Booting a System (Task Map)

Task	Description	For Instructions
Boot the Solaris Device Configuration Assistant	Used after changing the hardware configuration of the system. This utility enables you to boot the Solaris system from a different boot device, configure new or misconfigured hardware, or perform other device-related or boot-related tasks.	"x86: How to Boot the Solaris Device Configuration Assistant" on page 193
Boot the system	Select one of the following boot methods: Boot to run level 3 – Used after shutting down the system or performing some system hardware maintenance task.	"x86: How to Boot a System to Run Level 3 (Multiuser Level)" on page 193

Task	Description	For Instructions
	Boot to run level S - Used after performing some system maintenance task such as backing up a file system.	"x86: How to Boot a System to Run Level S (Single-User Level)" on page 194
	Boot interactively – Used after making temporary changes to a system file or the kernel for testing purposes.	"x86: How to Boot a System Interactively" on page 195
	Boot from the network - Used to boot a system from the network. This method is used for booting a diskless client.	"x86: How to Boot a System From the Network" on page 197
	Boot for recovery purposes - Used to boot the system when a damaged file is preventing the system from booting. You might need to do one or both of the following to boot for recovery purposes:	
	First, stop the system to attempt recovery.	"x86: How to Stop a System for Recovery Purposes" on page 198
	Boot to repair an important system file that is preventing the system from booting successfully.	"x86: How to Boot a System for Recovery Purposes" on page 198
	Boot `kadb` – Used to troubleshoot system problems.	"x86: How to Boot a System With the Kernel Debugger (`kadb`)" on page 203
	Force a crash dump and reboot the system - Used to force a crash dump for troubleshooting purposes.	"x86: Forcing a Crash Dump and Rebooting the System" on page 204

x86: Booting the Solaris Device Configuration Assistant

The Device Configuration Assistant (Solaris x86 Platform Edition) is a program that enables you to perform various hardware configuration and booting tasks. You can access the Solaris Device Configuration Assistant from either of the following:

- Solaris boot diskette

- Solaris Installation CD or DVD

In the procedures in this chapter, you might be requested to insert the Solaris Device Configuration Assistant boot diskette to boot the Configuration Assistant. If your system's BIOS supports booting from the CD or DVD, you can, instead, insert the Solaris installation CD or DVD to boot the Configuration Assistant.

▼ x86: How to Boot the Solaris Device Configuration Assistant

1. **Insert the Solaris Device Configuration Boot Diskette or the Solaris Installation CD or DVD in the appropriate drive.**
2. **If the system displays the `Type any key to continue` prompt, press any key to reboot the system.**

   You can also use the reset button at this prompt. If the system is shut down, turn the system on with the power switch.

   The first menu of the Configuration Assistant is displayed after a few minutes.

x86: Booting a System

The following procedures use the reset button to restart the system. If your system does not have a reset button, use the power switch to restart the system. You might be able to press the Ctrl-Alt-Del keys to interrupt system operation, depending upon the state of the system.

▼ x86: How to Boot a System to Run Level 3 (Multiuser Level)

Use this procedure to boot a system (that is currently at run level 0) to run level 3.

1. **If the system displays the `Type any key to continue` prompt, press any key to reboot the system.**

   You can also use the reset button at this prompt. If the system is shut down, turn the system on with the power switch.

   The Current Boot Parameters menu is displayed after a few minutes.

2. **Type `b` to boot the system to run level 3. Press Enter.**

If you do not make a selection within five seconds, the system is automatically booted to run level 3.

3. **Verify that the system has booted to run level 3.**

   The login prompt is displayed when the boot process has finished successfully.

   hostname `console login:`

Example—Booting a System to Run Level 3 (Multiuser Level)

```
Type any key to continue
           .
           .
           .
                     <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>      to boot with options
or      i <ENTER>                               to enter boot interpreter
or      <ENTER>                                 to boot with defaults

                  <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b
             .
             .
             .
venus console login:
```

▼ x86: How to Boot a System to Run Level S (Single-User Level)

Use this procedure to boot a system (that is currently at run level 0) to run level S.

1. **If the system displays the `Type any key to continue` prompt, press any key to reboot the system.**

   You can also use the reset button at this prompt. If the system is shut down, turn the system on with the power switch.

   The Current Boot Parameters menu is displayed after a few minutes.

2. **Type `b -s` to boot the system to run level S. Press Enter.**

   If you do not make a selection within five seconds, the system is automatically booted to run level 3.

3. **Type the superuser password, if prompted.**

4. **Verify that the system is at run level S by using the `who -r` command.**

```
# who -r
   .       run-level S  Jul 19 14:37     S      0  3
```

5. **Perform the maintenance task that required the run level change to S.**

6. **Press Control-D to bring the system back to run level 3.**

x86: Example—Booting a System to Run Level S (Single-User Level)

```
Type any key to continue
            .
            .
            .

                     <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>      to boot with options
or      i <ENTER>                               to enter boot interpreter
or      <ENTER>                                 to boot with defaults


                     <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b -s
              .
              .
              .
INIT: SINGLE USER MODE

Type Ctrl-d to proceed with normal startup,
(or give root password for system maintenance):  xxx
Entering System Maintenance Mode
              .
              .
              .
# who -r
   .       run-level S  Jul 19 14:37     S      0  3
(Perform some maintenance task)
# Press Control-D
```

▼ x86: How to Boot a System Interactively

Use this procedure to boot a system and you need to specify an alternate kernel or `/etc/system` file.

1. **If the system displays the `Type any key to continue` prompt, press any key to reboot the system.**

You can also use the reset button at this prompt. If the system is shut down, turn the system on with the power switch.

The Primary Boot Subsystem menu is displayed after a few minutes.

2. **Select the Solaris partition (if not marked as active) from the list and press Enter.**

   If you do not make a selection within five seconds, the active boot partition is selected automatically.

   The Current Boot Parameters menu is displayed after a few minutes.

3. **Type `b -a` to boot the system interactively. Press Enter.**

   If you do not make a selection within five seconds, the system is automatically booted to run level 3.

4. **Answer the system prompts as described in the following table.**

System Prompt	Action
`Enter default directory for modules: [/platform/i86pc/kernel /kernel /usr/kernel]:`	Provide an alternate path for the `modules` directory and press Enter. Or, press Enter to use the default `modules` directory path.
`Name of system file [etc/system]:`	Provide the name of an alternate system file and press Enter. Or, press Enter to use the default `/etc/system` file. Type `/dev/null` if your `/etc/system` file has been damaged.
`root filesystem type [ufs]:`	Press Enter to use the default root (`/`) file system. Type: UFS for local disk booting, or NFS for network booting.
`Enter physical name of root device` *`[physical_device_name]:`*	Provide an alternate device name and press Enter. Or, press Enter to use the default physical name of the root device `bootpath`.

x86: Example—Booting a System Interactively

In the following example, the default choices (shown in square brackets `[]`) are accepted.

```
Type any key to continue
            .
            .
            .

                    <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>      to boot with options
or      i  <ENTER>                              to enter boot interpreter
or      <ENTER>                                 to boot with defaults
```

```
                    <<< timeout in 5 seconds >>>>
Select (b)oot or (i)nterpreter: b -a
Enter default directory for modules [/platform/i86pc/kernel /kernel
/usr/kernel]: Press Enter
Name of system file [etc/system]:Press Enter
SunOS Release 5.9 Version Generic 32-bit
Copyright (c) 1983-2002 by Sun Microsystems, Inc.
root filesystem type [ufs]: Press Enter
Enter physical name of root device
[/pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a]: Press Enter
configuring IPv4 interfaces: dnet0.
Hostname: venus
(fsck messages)
The system is coming up.  Please wait
(More messages)
venus console login:
```

▼ x86: How to Boot a System From the Network

Any system can boot from the network if there is a boot server available. You might want to boot a standalone system from the network temporarily if the system cannot boot from the local disk.

If the system is capable of a PXE network boot, you might want to boot the system directly from the network without using either the Configuration Assistant boot diskette or the installation CD or DVD.

The menu, Set Network Configuration Strategy, on the Configuration Assistant's Boot Tasks Menu, enables you to select the appropriate boot strategy.

1. **Determine whether you want to boot from the network by using the RARP/bootparams method or the DHCP method.**

   There are two network configuration strategies to choose from, RARP (Reverse Address Resolution Protocol) or DHCP (Dynamic Host Configuration Protocol). The default network boot strategy is set to RARP. You can use either strategy depending on whether a RARP boot server or a DHCP boot server is available on your network.

   The PXE network boot is available only with DHCP.

2. **Insert the Configuration Assistant boot diskette or the installation CD or DVD that you wish to boot from.**

   Or, use the system or network adapter BIOS configuration program to enable the PXE network boot.

3. **If the system displays the `Type any key to continue` prompt, press any key to reboot the system.**

   You can also use the reset button at this prompt. If the system is shut down, turn the system on with the power switch.

The Solaris Device Configuration Assistant screen is displayed.

4. **Press the F2 key (F2_Continue) to scan for devices.**
   Device identification is performed and the Identified Devices screen is displayed.
5. **Press the F2 key (F2_Continue) to load drivers.**
   Bootable drivers are loaded.
   The Boot Solaris menu is displayed.
6. **Press the F4 key (F4_Boot Tasks).**
7. **Select `Set Network Configuration Strategy` and press the F2 key (F2_Continue).**
8. **Select either `RARP` or `DHCP` and press the F2 key (F2_Continue).**
   A screen that confirms your new network boot strategy appears.
   Your network boot strategy selection is saved as the default network boot method for the next time this diskette is used for booting.
9. **Press F3_Back to return to the Boot Solaris menu.**
10. **Select `NET` as the boot device. Then, press F2_Continue to boot the network device.**
    The Solaris boot option screen is displayed.

▼ x86: How to Stop a System for Recovery Purposes

If possible, stop the system by using one of the following commands:

- If the system is running, become superuser and type `init 0` to stop the system. After the `Type any key to continue` prompt appears, press any key to reboot the system.
- If the system is running, become superuser and type `init 6` to reboot the system.

If the system doesn't respond to any input from the mouse or keyboard, press the reset key, if it exists, to reboot the system. Or, you can use the power switch to reboot the system.

▼ x86: How to Boot a System for Recovery Purposes

Follow these steps to boot the system to repair a critical system resource. The example shows you how to boot from a Solaris Installation CD or from the network, mount the root (/) file system on the disk, and repair the `/etc/passwd` file.

Substitute the device name of the file system to be repaired for the *devicename* variable in the following procedure. If you need help with identifying a system's device names, refer to Chapter 30.

1. **Stop the system first by using the system stop key sequence.**

   Use the stop sequence for your system if you don't know the root password or if you can't log in to the system. For more information, see "x86: How to Stop a System for Recovery Purposes" on page 198.

2. **Boot from the Solaris installation CD or DVD (or from the network) to single-user mode.**

   a. **Insert the Configuration Assistant boot diskette or the installation CD or DVD that you wish to boot from.**

   b. **If the system displays the `Type any key to continue` prompt, press any key to reboot the system.**

      You can also use the reset button at this prompt. If the system is shut down, turn the system on with the power switch.

      The Solaris Device Configuration Assistant screen is displayed.

   c. **Press the F2 key (F2_Continue).**

      Device identification is performed and the Identified Devices screen is displayed.

   d. **Press the F2 key (F2_Continue).**

      Bootable drivers are loaded.

      The Boot Solaris menu is displayed.

   e. **Select the CD-ROM drive or network device. Then press the F2 key (F2_Continue).**

      The Current Boot Parameters menu is displayed.

   f. **Type `b -s` at the prompt. Press Enter.**

      After a few minutes, the single-user mode # prompt is displayed.

3. **Mount the root (`/`) file system that contains the invalid `passwd` file.**

   ```
   # mount /dev/dsk/devicename /a
   ```

4. **Change to the newly mounted `etc` directory.**

   ```
   # cd /a/etc
   ```

5. **Make the necessary change to the file by using an editor.**

   ```
   # vi filename
   ```

6. **Change to the root (`/`) directory.**

   ```
   # cd /
   ```

7. **Unmount the `/a` directory.**

   ```
   # umount /a
   ```

8. **Reboot the system.**

```
# init 6
```

9. **Verify that the system has booted to run level 3.**

   The login prompt is displayed when the boot process has finished successfully.

   hostname `console login:`

x86: Example—Booting a System for Recovery Purposes

The following example shows how to repair the `/etc/passwd` file after booting from a local CD-ROM.

```
Type any key to continue

SunOS Secondary Boot version 3.00


Solaris Intel Platform Edition Booting System

Running Configuration Assistant...
Autobooting from Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a

If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.
      .
      .
      .
Boot Solaris

Select one of the identified devices to boot the Solaris kernel and
choose Continue.

To perform optional features, such as modifying the autoboot and property
settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

[ ]  NET : DEC 21142/21143 Fast Ethernet
on Board PCI at Dev 3
[ ]  DISK: (*) Target 0, QUANTUM  FIREBALL1280A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  DISK: Target 1:ST5660A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  DISK: Target 0:Maxtor 9 0680D4
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[X]  CD  : Target 1:TOSHIBA  CD-ROM XM-5602B  1546
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
```

```
F2_Continue   F3_Back   F4_Boot Tasks   F6_Help
      .
      .
      .
               <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args: kernel/unix -r

Select the type of installation you want to perform:

1 Solaris Interactive
2 Custom JumpStart
3 Solaris Web Start

Enter the number of your choice followed by <ENTER> the key.

If you enter anything else, or if you wait for 30 seconds,
an interactive installation will be started.

Select type of installation:  b -s
      .
      .
      .
# mount /dev/dsk/c0t0d0s0 /a
      .
      .
      .
# cd /a/etc
# vi passwd
(Remove invalid entry)
# cd /
# umount /a
# init 6
```

x86: Example—Booting a System if You Forgot Root Password

The following example shows how to recover when you forget the root password by booting from the network. This example assumes that the boot server is already available. Be sure to apply a new root password after the system has rebooted.

```
Type any key to continue

SunOS Secondary Boot version 3.00


Solaris Intel Platform Edition Booting System

Running Configuration Assistant...
Autobooting from Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
```

```
If the system hardware has changed, or to boot from a different
device, interrupt the autoboot process by pressing ESC.

Press ESCape to interrupt autoboot in 5 seconds.
      .
      .
      .
Boot Solaris

Select one of the identified devices to boot the Solaris kernel and
choose Continue.

To perform optional features, such as modifying the autoboot and property
settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

[X]  NET : DEC 21142/21143 Fast Ethernet
on Board PCI at Dev 3
[ ]  DISK: (*) Target 0, QUANTUM  FIREBALL1280A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  DISK: Target 1:ST5660A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  DISK: Target 0:Maxtor 9 0680D4
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  CD  : Target 1:TOSHIBA  CD-ROM XM-5602B  1546
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1

F2_Continue   F3_Back   F4_Boot Tasks   F6_Help
      .
      .
      .
               <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args: kernel/unix -r

Select the type of installation you want to perform:

1 Solaris Interactive
2 Custom JumpStart
3 Solaris Web Start

Enter the number of your choice followed by <ENTER> the key.

If you enter anything else, or if you wait for 30 seconds,
an interactive installation will be started.

Select type of installation:  b -s
      .
      .
      .
# mount /dev/dsk/c0t0d0s0 /a
      .
```

```
      .
      .
# cd /a/etc
# vi shadow
(Remove root's encrypted password string)
# cd /
# umount /a
# init 6
```

▼ x86: How to Boot a System With the Kernel Debugger (kadb)

1. **If the system displays the `Type any key to continue` prompt, press any key to reboot the system.**

   You can also use the reset button at this prompt.

   If the system is shut down, turn the system on with the power switch.

2. **Type `b kadb` to boot the kernel debugger. Press Enter.**

   If you do not make a selection within five seconds, the system is automatically booted to run level 3.

3. **Verify that the system has booted to run level 3.**

   The login prompt is displayed when the boot process has finished successfully.

   hostname `console login:`

4. **Verify that you can access the kernel debugger by pressing F1-A.**

   The `kadb[0]:` prompt is displayed when you enter the kernel debugger.

Example—Booting a System With the Kernel Debugger (kadb)

```
Type any key to continue
        .
        .
        .
            <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>     to boot with options
or      i <ENTER>                              to enter boot interpreter
or      <ENTER>                                to boot with defaults

            <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter: b kadb
```

```
          .
          .
          .
naboo console login: (Enter login and password)
(Press F1-A to verify that you can access the kernel debugger)
```

x86: Forcing a Crash Dump and Rebooting the System

Forcing a crash dump and rebooting the system is sometimes necessary for troubleshooting purposes. The `savecore` feature is enabled by default.

For more information on system crash dumps, see "Managing System Crash Information (Tasks)" in *System Administration Guide: Advanced Administration*.

▼ x86: How to Force a Crash Dump and Reboot the System

The system must be booted with the kernel debugger option, `kadb`, to get to the `kadb[0]:` prompt and to enable you to force the crash dump.

Note – You must be in text mode to enter the kernel debugger (`kadb`). So, first exit any window system.

---

1. **Press F1–A.**

   ```
   kadb[0]:
   ```

   The `kadb[0]:` prompt is displayed.

2. **Type the following commands at the `kadb[0]:` prompt.**

   ```
   Press <F1-a>
   kadb[0]: vfs_syncall/W ffffffff
   kadb[0]: 0>eip
   kadb[0]: :c
   kadb[0]: :c
   kadb[0]: :c
   ```

   After you type the first `:c` , the system panics, so you need to type `:c` again. The system panics again, so type `:c` a third time to force the crash dump and reboot the system.

After the crash dump is written to disk, the system continues to reboot.

3. **Verify that the system has rebooted by logging in at the console login prompt.**

CHAPTER 15

The Boot Process (Reference)

This chapter describes the firmware used for booting SPARC based and x86 based systems. This chapter also provides an overview of the boot process on each platform.

This is a list of the reference information in this chapter.

- "SPARC: The Boot PROM" on page 207
- "SPARC: The Boot Process" on page 208
- "x86: The PC BIOS" on page 208
- "x86: Boot Subsystems" on page 209
- "x86: The Boot Process" on page 214

For step-by-step instructions on booting a system, see Chapter 13 or Chapter 14.

SPARC: The Boot PROM

Each SPARC based system has a PROM (programmable read-only memory) chip with a program called the *monitor*. The monitor controls the operation of the system before the Solaris kernel is available. When a system is turned on, the monitor runs a quick self-test procedure to checks the hardware and memory on the system. If no errors are found, the system begins the automatic boot process.

Note – Some older systems might require PROM upgrades before they will work with the Solaris system software. Contact your local service provider for more information.

SPARC: The Boot Process

The following table describes the boot process on SPARC based systems.

TABLE 15–1 SPARC: Description of the Boot Process

Boot Phase	Description
Boot PROM	1. The PROM displays system identification information and then runs self-test diagnostics to verify the system's hardware and memory.
	2. Then, the PROM loads the primary boot program, `bootblk`, whose purpose is to load the secondary boot program (that is located in the `ufs` file system) from the default boot device.
Boot Programs	3. The `bootblk` program finds and executes the secondary boot program, `ufsboot`, and loads it into memory.
	4. After the `ufsboot` program is loaded, the `ufsboot` program loads the kernel.
Kernel Initialization	5. The kernel initializes itself and begins loading modules by using `ufsboot` to read the files. When the kernel has loaded enough modules to mount the root (`/`) file system, the kernel unmaps the `ufsboot` program and continues, using its own resources.
	6. The kernel creates a user process and starts the `/sbin/init` process, which starts other processes by reading the `/etc/inittab` file.
`init`	7. The `/sbin/init` process starts the run control (`rc`) scripts, which execute a series of other scripts. These scripts (`/sbin/rc*`) check and mount file systems, start various processes, and perform system maintenance tasks.

x86: The PC BIOS

Before the kernel is started, the system is controlled by the read-only-memory (ROM) Basic Input/Output System (BIOS), which is the firmware interface on a PC.

Hardware adapters can have an on-board BIOS that displays the physical characteristics of the device and can be used to access the device.

During the startup sequence, the PC BIOS checks for the presence of any adapter BIOS, and if found, loads and executes each adapter BIOS. Each individual adapter's BIOS runs self-test diagnostics and displays device information.

x86: Boot Subsystems

At three points during the Solaris boot process, you can make the following choices about a booting system as follows:

- **Primary Boot Subsystem (Partition Boot Menu)** – This first menu appears if multiple operating environments exist on the disk. The menu enables you to boot any of the operating environments installed. By default, the operating environment that is designed as *active* is booted.

  Note that if you choose to boot a non-Solaris operating environment, you cannot reach the next two menus.
- **Interrupt the Autoboot Process** – If the autoboot process is interrupted, you can access the Solaris Device Configuration Assistant.

  The Solaris Device Configuration Assistant enables you to boot the Solaris system from a different boot device, configure new or misconfigured hardware, or perform other device-related or boot-related tasks.
- **Current Boot Parameters Menu** – Two forms of this menu exist, one for a normal Solaris boot and one menu for a Solaris installation boot:
  - The normal Current Boot Parameters menu enables you to boot the Solaris system with options, or enter the boot interpreter.
  - The install Current Boot Parameters menu enables you to select the type of installation to be performed, or customize the boot.

The following table summarizes the purpose of the primary x86 boot interfaces. See the sections that follow for a detailed description and example of each boot interface.

TABLE 15–2 x86: Boot Subsystems

Boot Subsystem	Purpose
Primary Boot Subsystem	This menu appears if the disk you are booting from contains multiple operating environments, including the Solaris operating environment.
Secondary Boot Subsystem	This menu appears each time you boot the Solaris release. The Solaris release is booted automatically unless you choose to run the Solaris Device Configuration Assistant by interrupting the autoboot process.

TABLE 15–2 x86: Boot Subsystems *(Continued)*

Boot Subsystem	Purpose
Solaris Device Configuration Assistant/Boot Diskette	There are two ways to access the Solaris Device Configuration Assistant menus: 1. Use the Solaris Device Configuration Assistant boot diskette or the Solaris installation CD (on systems that can boot from the CD-ROM drive) to boot the system. 2. Interrupt the autoboot process when you boot the Solaris software from an installed disk.
Current Boot Parameters Menu	This menu appears when you boot the Solaris release from the disk, CD-ROM, or the network. The menu presents a list of boot options.

Note – If you need to create the Solaris Device Configuration Assistant boot diskette, go to `http://soldc.sun.com/support/drivers/dca_diskettes`.

During the boot process, the boot subsystem menus allow you to customize boot choices. If the system receives no response during the time-out periods, it continues to boot automatically using the default selections. You can stop the boot process when each boot subsystem menu is displayed. Or, you can let the boot process continue automatically.

The following section provides examples of each boot subsystem screen.

x86: Booting the Solaris Release

During the device identification phase, the Solaris Device Configuration Assistant does the following:

- Scans for devices that are installed on the system
- Displays the identified devices
- Enables you to perform optional tasks such as selecting a keyboard type and editing devices and their resources

During the boot phase, the Solaris Device Configuration Assistant does the following:

- Displays a list of devices from which to boot. The device marked with an asterisk (*) is the default boot device.
- Enables you to perform optional tasks, such as editing autoboot settings and property settings, and choosing the network configuration strategy.

The following section provides examples of menus that appear during the device identification phase. The device output varies based on your system configuration.

x86: Screens Displayed During the Device Identification Phase

Several screens are displayed as the Solaris Device Configuration Assistant attempts to identify devices on the system.

x86: Configuration Assistant Screen

This screen appears each time you boot the Solaris Device Configuration Assistant. The Solaris Device Configuration Assistant runs every time the system is booted, although the autoboot process bypasses the menus.

```
Solaris Device Configuration Assistant

The Solaris(TM) (Intel Platform Edition) Device Configuration Assistant
scans to identify system hardware, lists identified devices, and can
boot the Solaris software from a specified device. This program must be
used to install the Solaris operating environment, add a driver,
or change the hardware on the system.

> To perform a full scan to identify all system hardware, choose Continue.

> To diagnose possible full scan failures, choose Specific Scan.

> To add new or updated device drivers, choose Add Driver.

About navigation...
    - The mouse cannot be used.
    - If the keyboard does not have function keys or they do not respond,
      press ESC. The legend at the bottom of the screen will change to
      show the ESC keys to use for navigation.
    - The F2 key performs the default action.

F2_Continue        F3_Specific Scan        F4_Add Driver        F6_Help
```

x86: Bus Enumeration Screen

The Bus Enumeration screen appears briefly while the Solaris Device Configuration Assistant gathers hardware configuration data for devices that can be detected automatically.

```
Bus Enumeration

Determining bus types and gathering hardware configuration data ...

Please wait ...
```

x86: Scanning Devices Screen

The Scanning Devices screen appears while the Solaris Device Configuration Assistant manually scans for devices that can only be detected with special drivers.

```
Scanning Devices

The system is being scanned to identify system hardware.

If the scanning stalls, press the system's reset button. When the
system reboots, choose Specific Scan or Help.


Scanning: Floppy disk controller

#######################
|      |        |         |         |         |
0              20       40        60           80       100

Please wait ...
```

x86: Identified Devices Screen

The Identified Devices screen displays which devices have been identified on the system. From here, you can continue to the Boot Solaris menu or perform optional device tasks, such as setting a keyboard configuration, viewing and editing devices, setting up a serial console, and saving and deleting configurations.

```
Identified Devices

The following devices have been identified on this system. To identify
devices not on this list or to modify device characteristics, such as
keyboard configuration, choose Device Tasks. Platform types may be
included in this list.

ISA: Floppy disk controller
    ISA: Motherboard
    ISA: PnP bios: 16550-compatible serial controller
    ISA: PnP bios: 16550-compatible serial controller
    ISA: PnP bios: Mouse controller
    ISA: PnP bios: Parallel port
    ISA: System keyboard (US-English)
    PCI: Bus Mastering IDE controller
    PCI: Universal Serial Bus
    PCI: VGA compatible display adapter

F2_Continue   F3_Back   F4_Device Tasks   F6_Help
```

x86: Menus Displayed During the Boot Phase

During this phase, you can determine the way in which the system is booted.

x86: Boot Solaris Menu

The Boot Solaris menu allows you to select the device from which to boot the Solaris release. You can also perform optional tasks, such as viewing and editing autoboot and property settings. Once you select a boot device and you choose Continue, the Solaris kernel begins to boot.

```
Boot Solaris
Select one of the identified devices to boot the Solaris kernel and
choose Continue.

To perform optional features, such as modifying the autoboot and property
settings, choose Boot Tasks.

An asterisk (*) indicates the current default boot device.

> To make a selection use the arrow keys, and press Enter to mark it [X].

[X]  DISK: (*) Target 0:QUANTUM  FIREBALL1280A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  DISK: Target 1:ST5660A
on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  DISK: Target 0:Maxtor 9 0680D4
    on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1
[ ]  CD  : Target 1:TOSHIBA  CD-ROM XM-5602B  1546
    on Bus Mastering IDE controller on Board PCI at Dev 7, Func 1

F2_Continue   F3_Back   F4_Boot Tasks   F6_Help
```

x86: Current Boot Parameters Menu

This menu appears each time you boot the Solaris release from the local disk. Let the five-second timeout elapse if you want to boot the default Solaris kernel. If you want to boot with different options, select an appropriate option before the time-out period elapses.

```
                    <<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0:a
Boot args:
Type    b [file-name] [boot-flags] <ENTER>       to boot with options
or      i <ENTER>                                to enter boot interpreter
or      <ENTER>                                  to boot with defaults


                    <<< timeout in 5 seconds >>>

Select (b)oot or (i)nterpreter:
```

x86: The Boot Process

The following table describes the boot process on x86 based systems.

TABLE 15–3 x86: Description of the Boot Process

Boot Phase	Description
BIOS	1. When the system is turned on, the BIOS runs self-test diagnostics to verify the system's hardware and memory. The system begins to boot automatically if no errors are found. If errors are found, error messages are displayed that describe recovery options. The BIOS of additional hardware devices are run at this time. 2. The BIOS boot program tries to read the first physical sector from the boot device. This first disk sector on the boot device contains the master boot record `mboot`, which is loaded and executed. If no `mboot` file is found, an error message is displayed.
Boot Programs	3. The master boot record, `mboot`, which contains disk information needed to find the active partition and the location of the Solaris boot program, `pboot`, loads and executes `pboot`. 4. The Solaris boot program, `pboot` loads `bootblk`, the primary boot program, whose purpose is to load the secondary boot program that is located in the `ufs` file system. 5. If there is more than one bootable partition, `bootblk` reads the `fdisk` table to locate the default boot partition, and builds and displays a menu of available partitions. You have a 30-second interval to select an alternate partition from which to boot. This step only occurs if there is more than one bootable partition present on the system. 6. `bootblk` finds and executes the secondary boot program, `boot.bin` or `ufsboot`, in the root (`/`) file system. You have a 5–second interval to interrupt the autoboot to start the Solaris Device Configuration Assistant. 7. The secondary boot program, `boot.bin` or `ufsboot`, starts a command interpreter that executes the `/etc/bootrc` script, which provides a menu of choices for booting the system. The default action is to load and execute the kernel. You have a 5–second interval to specify a boot option or to start the boot interpreter.
Kernel initialization	8. The kernel initializes itself and begins loading modules by using the secondary boot program (`boot.bin` or `ufsboot`) to read the files. When the kernel has loaded enough modules to mount the root (`/`) file system, the kernel unmaps the secondary boot program and continues, using its own resources.

TABLE 15–3 x86: Description of the Boot Process *(Continued)*

Boot Phase	Description
	9. The kernel creates a user process and starts the `/sbin/init` process, which starts other processes by reading the `/etc/inittab` file.
`init`	10. The `/sbin/init` process starts the run control (`rc`) scripts, which execute a series of other scripts. These scripts (`/sbin/rc*`) check and mount file systems, start various processes, and perform system maintenance tasks.

CHAPTER 16

Managing Removable Media Topics

This topic map lists the chapters that provide information on managing removable media.

Chapter 17	Provides overview information about managing removable media from the command line.
Chapter 18	Provides step-by-step instructions for accessing removable media from the command line.
Chapter 19	Provides step-by-step instructions for formatting removable media from the command line.
Chapter 20	Provides step-by-step instructions for writing data and audio CDs.

CHAPTER 17

Managing Removable Media (Overview)

This chapter provides general guidelines for managing removable media in the Solaris environment.

This is a list of the overview information in this chapter.

- "What's New in Managing Removable Media?" on page 219
- "Where to Find Managing Removable Media Tasks" on page 220
- "Removable Media Features and Benefits" on page 220
- "Comparison of Automatic and Manual Mounting" on page 221
- "What You Can Do With Volume Management" on page 222

What's New in Managing Removable Media?

Volume management features have been improved to fully support removable media. This improvement means that DVD-ROMs, Iomega and Universal Serial Bus (USB) Zip drives and Jaz drives, CD-ROMs, and diskettes are mounted and available for reading when they are inserted.

You can use both the Common Desktop Environment (CDE) volume management and the Solaris command line to fully manage removable media.

With the volume management improvements, you can:

- Format, label, and set read or write software protection on removable media with the new `rmformat` command. This command replaces the `fdformat` command for formatting removable media.
- Create and verify a PCFS file system on removable media with the `mkfs_pcfs` and `fsck_pcfs` commands.

- Create an `fdisk` partition and a PCFS file system on removable media on a SPARC system to facilitate data transfers to x86 systems.

Guidelines for using removable media are:

- Use UDFS and PCFS to transfer data between DVD media.
- Use the `tar` or `cpio` commands to transfer files between rewritable media such as a PCMCIA memory card or diskette with a UFS file system. A UFS file system that is created on a SPARC system is not identical to a UFS file system on PCMCIA or to a diskette that is created on an x86 system.
- Set write protection to protect important files on Jaz or Zip drives or diskettes. Apply a password to Iomega media.

Where to Find Managing Removable Media Tasks

Use these references to find step-by-step instructions for managing removable media.

Removable Media Management Task	For More Information
Access removable media	Chapter 18
Format removable media	Chapter 19
Write data and music CDs	Chapter 20

For information on using removable media with File Manager in the Common Desktop Environment, see *Solaris Common Desktop Environment: User's Guide*.

Removable Media Features and Benefits

The Solaris environment gives users and software developers a standard interface for dealing with removable media. Referred to as volume management, this interface provides three major benefits:

- By automatically mounting removable media, it simplifies their use. (For a comparison between manual and automatic mounting, see the following section.)
- It enables you to access removable media without having to become superuser.

- It allows you to give other systems on the network automatic access to any removable media on your local system. For more information, see Chapter 18.

Comparison of Automatic and Manual Mounting

The following table compares the steps involved in manual mounting (without volume management) and automatic mounting (with volume management) of removable media.

TABLE 17–1 Comparison of Manual and Automatic Mounting

Steps	Manual Mounting	Automatic Mounting
1	Insert media.	Insert media.
2	Become superuser.	For diskettes, use the `volcheck` command.
3	Determine the location of the media device.	Volume manager (`vold`) automatically performs many of the tasks previously required to manually mount and work with removable media.
4	Create a mount point.	
5	Make sure you are not in the mount point directory.	
6	Mount the device using the proper `mount` options.	
7	Exit the superuser account.	
8	Work with files on media.	Work with files on media.
9	Become superuser.	
10	Unmount the media device.	
11	Eject media.	Eject media.
12	Exit the superuser account.	

What You Can Do With Volume Management

Essentially, volume management enables you to access removable media just as manual mounting does, but more easily and without the need for superuser access. To make removable media easier to work with, you can mount removable media in easy-to-remember locations.

TABLE 17–2 How to Access Data on Removable Media Managed by Volume Manager

Access	Insert	Find the Files Here
Files on the first diskette	The diskette and enter `volcheck`	`/floppy`
Files on the first removable hard disk	The removable hard disk and enter `volcheck`	`/rmdisk/jaz0` or `/rmdisk/zip0`
Files on the first CD	The CD and wait for a few seconds	`/cdrom/`*volume-name*
Files on the first DVD	The DVD and wait for a few seconds	`/dvd/`*volume-name*
Files on the first PCMCIA	The PCMCIA and wait for a few seconds	`/pcmem/pcmem0`

If your system has more than one type of removable device, see the following table for their access points.

TABLE 17–3 Where to Access Removable Media

Media Device	Access File Systems With This Path	Access Raw Data With This Path
First diskette drive	`/floppy/floppy0`	`/vol/dev/aliases/floppy0`
Second diskette drive	`/floppy/floppy1`	`/vol/dev/aliases/floppy1`
First CD-ROM drive	`/cdrom/cdrom0`	`/vol/dev/aliases/cdrom0`
Second CD-ROM drive	`/cdrom/cdrom1`	`/vol/dev/aliases/cdrom1`
First removable hard disk	`/rmdisk/jaz0`	`/vol/dev/aliases/jaz0`
	`/rmdisk/zip0`	`/vol/dev/aliases/zip0`
First PCMCIA drive	`/pcmem/pcmem0`	`/vol/dev/aliases/pcmem0`

CHAPTER 18

Accessing Removable Media (Tasks)

This chapter describes how to access removable media from the command line in the Solaris environment.

For information on the procedures associated with accessing removable media, see the following:

- "Accessing Removable Media (Task Map)" on page 223
- "Accessing Removable Media on a Remote System (Task Map)" on page 232

For background information on removable media, see Chapter 17.

Accessing Removable Media (Task Map)

Task	Description	For Instructions
1. (Optional) Add the removable media drive	Add the removable media drive to your system, if necessary.	"How to Add a New Removable Media Drive" on page 226
2. (Optional) Decide whether you want to use removable media with or without volume management (`vold`)	Volume management (`vold`) runs by default. Decide whether you want to use removable media with or without volume management.	"How to Stop and Start Volume Management (`vold`)" on page 227
3. Access removable media	Access different kinds of removable media with or without volume management running.	"How to Access Information on Removable Media" on page 228

Task	Description	For Instructions
4. (Optional) Copy files or directories	Copy files or directories from the media as you would from any other location in the file system.	"How to Copy Information From Removable Media" on page 229
5. (Optional) Configure a system to play musical CDs or DVDs	You can configure a system to play musical CDs or DVDs, but you will need third-party software to play the media.	"How to Play a Musical CD or DVD" on page 229
6. Find out if the media still in use	Before ejecting the media, find out if it is still in use.	"How to Find Out If Removable Media Is Still in Use" on page 230
7. Eject the Media	When you finish, eject the media from the drive.	"How to Eject Removable Media" on page 231

Accessing Removable Media (Overview)

You can access information on removable media with or without using volume manager. For information on accessing information on removable media with CDE's File Manager, see "Using Removable Media with File Manager" in *Solaris Common Desktop Environment: User's Guide*.

Starting in the Solaris 8 6/00 release, volume manager (`vold`) actively manages all removable media devices. This means any attempt to access removable media with device names such as `/dev/rdsk/c`*`n`*`t`*`n`*`d`*`n`*`s`*`n` or `/dev/dsk/c`*`n`*`t`*`n`*`d`*`n`*`s`*`n` will be unsuccessful.

Using Removable Media Names

You can access all removable media with different names. The following table describes the different media names that can be accessed with or without volume management.

TABLE 18–1 Removable Media Names

Media	Volume Management Device Name	Volume Management Device Alias Name	Device Name

First diskette drive	`/floppy`	`/vol/dev/aliases/floppy0`	`/dev/rdiskette`
			`/vol/dev/rdiskette0/`
			volume-name

First, second, third CD-ROM or DVD-ROM drives	`/cdrom0`
	`/cdrom1`
	`/cdrom2`
		`/vol/dev/aliases/cdrom0`
		`/vol/dev/aliases/cdrom1`
		`/vol/dev/aliases/cdrom2`
			`/vol/dev/rdsk/c`*n*`t`*n* `[d`*n*`]/`
			volume-name

First, second, third Jaz drive	`/rmdisk/jaz0`
	`/rmdisk/jaz1`
	`/rmdisk/jaz2`
		`/vol/dev/aliases/jaz0`
		`/vol/dev/aliases/jaz1`
		`/vol/dev/aliases/jaz2`
			`/vol/dev/rdsk/c`*n*`t`*n*`d`*n*`/`
			volume-name

First, second, third Zip drive	`/rmdisk/zip0`
	`/rmdisk/zip1`
	`/rmdisk/zip2`
		`/vol/dev/aliases/zip0`
		`/vol/dev/aliases/zip1`
		`/vol/dev/aliases/zip2`
			`/vol/dev/rdsk/c`*n*`t`*n*`d`*n*`/`
			volume-name

First, second, third, PCMCIA drive	`/pcmem/pcmem0`
	`/pcmem/pcmem1`
	`/pcmem/pcmem2`
		`/vol/dev/aliases/pcmem0`
		`/vol/dev/aliases/pcmem1`
		`/vol/dev/aliases/pcmem2`
			`/vol/dev/rdsk/c`*n*`t`*n*`d`*n*`/`
			volume-name

Use this table to identify which removable media name to use with specific Solaris commands.

Solaris Command	Device Name	Usage Examples

`ls`, `more`, `vi`	`/floppy`
	`/cdrom`
	`/rmdisk/zip0`
	`/rmdisk/jaz0`
	`/pcmem/pcmem0`
		`ls /floppy/myfiles/`
		`more /cdrom/myfiles/filea`

`fsck`, `newfs`, `mkfs`	`/vol/dev/aliases/floppy0` `/vol/dev/rdsk/c`*n*`t`*n*`d`*n*	`newfs /vol/dev/aliases/floppy0` `mkfs -F udfs /vol/dev/rdsk/c`*n*`t`*n*`d`*n*

Guidelines for Accessing Removable Media Data

Most CDs and DVDs are formatted to the ISO 9660 standard, which is portable, so most CDs and DVDs can be mounted by volume management. However, CDs or DVDs with UFS file systems are not portable between architectures, so they must be used on the architecture for which they were designed.

For example, a CD or DVD with a UFS file system for a SPARC platform cannot be recognized by an x86 platform. Likewise, an x86 UFS CD cannot be mounted by volume management on a SPARC platform. The same limitation applies to diskettes. (Actually, some architectures share the same bit structure, so occasionally a UFS format specific to one architecture will be recognized by another architecture, but the UFS file system structure was not designed to guarantee this compatibility).

To accommodate the different formats, the CD or DVD is split into slices, which are similar in effect to partitions on hard disks. The 9660 portion is portable, but the UFS portion is architecture-specific. If you are having trouble mounting a CD or DVD, particularly if it is an installation CD or DVD, make sure its UFS file system is appropriate for your system's architecture (check the label on the CD or DVD).

Accessing Jaz Drives or Zip Drives

You can determine whether accessing your Jaz or Zip drives changes from previous Solaris releases, depending on the following:

- If you are upgrading from the Solaris 8 6/00 release to the Solaris 9 release, you can continue to access your Jaz drives and Zip drives in the same way as in previous releases.
- If you are freshly installing the Solaris 9 release, you cannot access your Jaz drives and Zip drives in the same way as in previous Solaris releases.

  Follow these steps if you want to access your Jaz and Zip drives in the same way as in previous Solaris releases:

  1. Comment the following line in the `/etc/vold.conf` file by inserting a pound (#) sign at the beginning of the text, like this:

     ```
     # use rmdisk drive /dev/rdsk/c*s2 dev_rmdisk.so rmdisk%d
     ```

  2. Reboot the system.

▼ How to Add a New Removable Media Drive

Adding a new removable media drive involves creating the `/reconfigure` file and rebooting the system so that volume management recognizes the new media drive.

1. **Become superuser.**
2. **Create the `/reconfigure` file.**

```
# touch /reconfigure
```

3. **Bring the system to run level 0.**

```
# init 0
```

4. **Turn off power to the system.**
5. **Connect the new media drive.**

   See your hardware handbook for specific instructions.

6. **Turn on power to the system.**

   The system comes up to multiuser mode automatically.

▼ How to Stop and Start Volume Management (`vold`)

Occasionally, you might want to manage media without the help of volume management. This section describes how to stop and restart volume management.

▼ How to Stop Volume Management (`vold`)

1. **Make sure media is not being used.**

   If you are not sure whether you have found all users of the media, use the `fuser` command, as described in "How to Find Out If Removable Media Is Still in Use" on page 230.

2. **Become superuser.**
3. **Enter the `volmgt stop` command.**

```
# /etc/init.d/volmgt stop
#
```

▼ How to Restart Volume Management (`vold`)

1. **Become superuser.**
2. **Enter the `volmgt start` command.**

```
# /etc/init.d/volmgt start
volume management starting.
```

▼ How to Access Information on Removable Media

1. **Insert the media.**

   The media is mounted after a few seconds.

2. **Check for media in the drive.**

   ```
   % volcheck
   ```

   Use the appropriate device name to access information by using the command-line interface. See Table 18–1 for an explanation of device names.

3. **List the contents of the media.**

   ```
   % ls /media
   ```

Examples—Accessing Information on Removable Media

Access information on a diskette as follows:

```
$ volcheck
$ ls /floppy
myfile
```

Access information on a Jaz drive as follows:

```
$ volcheck
$ ls /rmdisk
jaz0/        jaz1/
```

Access information on a CD-ROM as follows:

```
$ volcheck
$ ls /cdrom
solaris_9_sparc/
```

View the symbolic links on a CD-ROM as follows:

```
$ ls -lL /cdrom/cdrom0
total 166
drwxr-xr-x   4 root     root        2048 Jul 21 05:18 MU
drwxr-xr-x   4 root     root        2048 Jul 21 05:18 Solaris_7_MU3
-rwxr-xr-x   1 root     root       30952 Jul 21 05:18 backout_mu
-rwxr-xr-x   1 root     root       49604 Jul 21 05:18 install_mu
```

Access information on a PCMCIA memory card as follows

```
$ ls /pcmem/pcmem0
pcmem0 myfiles
```

▼ How to Copy Information From Removable Media

You can access files and directories on removable media just like any other file system. The only significant restrictions are ownership and permissions.

For instance, if you copy a file from a CD into your file system, you'll be the owner, but you won't have write permissions (because the file never had them on the CD). You'll have to change the permissions yourself.

1. **Make sure the media is mounted.**

   $ **ls** */media*

   The `ls` command displays the contents of a mounted media. If no contents are displayed, see "How to Access Information on Removable Media" on page 228.

2. **(Optional) Copy the files or directories.**

   For example, for a CD, you would do the following:

```
$ cp /cdrom/sol_8_u3_sparc_2/Solaris_8/EA/products/Live*/README*
$ ls -l
-r--r--r--   1 pmorph   users       3002 May  9 08:09 README_Live_Upgrade
```

   For example, for a PCMCIA memory card, you would do the following:

```
$ cp /pcmem/pcmem0/readme2.doc .
$ cp -r /pcmem/pcmem0/morefiles .
```

▼ How to Play a Musical CD or DVD

To play musical media from a media drive attached to a system running the Solaris release, you'll need to access public domain software, such as `xmcd`, that is available from the following locations:

- http://www.ibiblio.org/tkan/xmcd

  This site includes frequent updates to the `xmcd` software, which includes the version of `xmcd` that plays on newer Sun hardware, such as the Sun Blade™ systems.
- http://www.sun.com/software/solaris/freeware/pkgs_download.html

Keep the following in mind when using the `xmcd` software with CDDA (CD Digital Audio) support to play musical media:

- Use `xmcd`, version 3.1 (or later) on Sun Blade systems because this version has CDDA support, which must be enabled in order to listen to CDs on these systems.
- Enable CDDA by launching `xmcd`, clicking on the options button (it has a hammer and screwdriver on the button), and then by clicking on "CDDA playback".
- When CDDA is enabled, audio is directed to the audio device, so headphones and external speakers should be connected to the audio device and not to the media drive itself.

- CDDA can be enabled on other machines too. Enabling CDDA is required for playing media on the Sun Blade systems.

Consider the following issues as well:

- If you are using `xmcd` with standard playback on a system that *does not* have an internal connection from the CD-ROM to the audio device, you must insert headphones into the CD-ROM drive's headphone port.
- If you are using `xmcd` with standard playback on a system that *does* have an internal connection from the CD-ROM to the audio device, you can do either of the following:

  1. Insert headphones into the headphone port of the CD-ROM drive.
  2. Insert headphones into the headphone port on the audio device.

  If you choose #2, you must do the following:

  - Select the internal CD as the input device.
  - Make sure that Monitor Volume is non-zero.

  You can do both of these from `sdtaudiocontrol`'s record panel.

Once you install the `xmcd` software, you can play a musical CD simply by inserting it into the CD-ROM drive and starting the `xmcd` control panel.

1. **Install the `xmcd` software.**
2. **Insert the media into the media drive.**
3. **Invoke the `xmcd` command.**

   ```
   % ./xmcd &
   ```

▼ How to Find Out If Removable Media Is Still in Use

1. **Become superuser.**
2. **Identify the processes accessing the media.**

   ```
   # fuser -u [-k] /media
   ```

`-u`	Displays the user of the media.
`-k`	Kills the process accessing the media.

   For more information on using the `fuser` command, see `fuser`(1M).

3. **(Optional) Kill the process accessing the media.**

```
# fuser -u -k /media
```

Caution – Killing the process accessing the media should only be used in emergency situations.

4. **Verify the process is gone.**

```
# pgrep process-ID
```

Example—Finding Out If the Media Is Still in Use

The following example shows that the process `26230c`, owner `ripley`, is accessing the `/cdrom/cdrom0/Solaris_8/EA/products/Live_Upgrade_1.0` directory.

```
# fuser -u /cdrom/cdrom0/Solaris_8/EA/products/Live_Upgrade_1.0
/cdrom/cdrom0/Solaris_8/EA/products/Live_Upgrade_1.0:    26230c(ripley)
```

▼ How to Eject Removable Media

1. **Make sure the media is not being used.**

   Remember, media is "being used" if a shell or an application is accessing any of its files or directories. If you are not sure whether you have found all users of a CD (a shell hidden behind a desktop tool might be accessing it), use the `fuser` command, as described in "How to Find Out If Removable Media Is Still in Use" on page 230.

2. **Eject the media.**

```
# eject media
```

   For example, for a CD, you would do the following

```
# eject cdrom
```

   For example, for a PCMCIA memory card, you would do the following:

```
# eject pcmem0
```

Accessing Removable Media on a Remote System (Task Map)

The following table describes the tasks need to access removable media on a remote system.

Task	Description	For Instructions
1. Make local media available to remote systems	Add the removable media drive to your system, if necessary.	"How to Make Local Media Available to Other Systems" on page 232
2. Access removable media on remote systems	Insert the media into the drive.	"How to Access Information on Removable Media" on page 228

▼ How to Make Local Media Available to Other Systems

You can configure your system to share its media drives to make any media in those drives available to other systems. (This does not apply to musical CDs.) Once your media drives are shared, other systems can access the media they contain simply by mounting them, as described in "How to Access Removable Media on Remote Systems" on page 235.

1. **Become superuser.**
2. **Find out whether the NFS daemon (`nfsd`) is running.**

```
# ps -ef | grep nfsd
root 14533    1 17 10:46:55 ?     0:00 /usr/lib/nfs/nfsd -a 16
root 14656  289  7 14:06:02 pts/3 0:00 grep nfsd
```

   If the daemon is running, a line for `/usr/lib/nfs/nfsd` will appear, as shown above. If the daemon is not running, only the `grep nfsd` line will appear.

3. **Identify the `nfsd` status and select one of the following:**

   a. **If `nfsd` is running, go to Step 8.**

   b. **If `nfsd` is *not* running, continue with Step 4.**

4. **Create a dummy directory for `nfsd` to share.**

```
# mkdir / dummy-dir
```

dummy-dir	Can be any directory name; for example, *dummy*. This directory will not contain any files. Its only purpose is to "wake up" the NFS daemon so that it notices your shared media drive.

5. **Add the following entry into the `/etc/dfs/dfstab` file.**

   ```
   share -F nfs -o ro [-d comment] /dummy-dir
   ```

   When you start the NFS daemon, it will see this entry, "wake up," and notice the shared media drive. Note that the comment (preceded by `-d`) is optional.

6. **Start the NFS daemon.**

   ```
   # /etc/init.d/nfs.server start
   ```

7. **Verify that the NFS daemon is indeed running.**

   ```
   # ps -ef | grep nfsd
   root 14533    1 17 10:46:55 ?     0:00 /usr/lib/nfs/nfsd -a 16
   root 14656  289  7 14:06:02 pts/3 0:00 /grep nfsd
   ```

8. **Eject any media currently in the drive.**

   ```
   # eject media
   ```

9. **Assign root write permissions to the `/etc/rmmount.conf` file.**

   ```
   # chmod 644 /etc/rmmount.conf
   ```

10. **Add the following lines to the `/etc/rmmount.conf` file.**

    ```
    # File System Sharing
    share media*
    ```

    These lines share any media loaded into your system's CD-ROM drive. You can, however, limit sharing to a particular CD or series of CDs, as described in `share`(1M).

11. **Remove write permissions from the `/etc/rmmount.conf` file.**

    ```
    # chmod 444 /etc/rmmount.conf
    ```

    This step returns the file to its default permissions.

12. **Load the media.**

    The media you now load, and all subsequent media, will be available to other systems. Remember to wait until the light on the drive stops blinking before you verify this task.

    To access the media, the remote user must mount it by name, according to the instructions in "How to Access Removable Media on Remote Systems" on page 235.

13. **Verify that the media is indeed available to other systems by using the `share` command.**

If the media is available, its share configuration will be displayed. (The shared `dummy` directory will also be displayed.)

```
# share
-    /dummy  ro "dummy dir to wake up NFS daemon"
-    /cdrom/sol_9_sparc  ro  ""
```

Example—Making Local CDs Available to Other Systems

The following example shows how to make any local CD available to other systems on the network.

```
# ps -ef | grep nfsd
    root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
    root 10118     1  0 08:24:39 ?        0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(Add the following line:)
share -F nfs -o ro  /dummy
# eject cdrom0
# chmod 644 /etc/rmmount.conf
# vi /etc/rmmount
(Add the following line to the File System Sharing section:)
share cdrom*
# chmod 444 /etc/rmmount.conf
(Load a CD.)
# share
-               /dummy   ro   ""
-               /cdrom/sol_9_sparc/s5   ro   ""
-               /cdrom/sol_9_sparc/s4   ro   ""
-               /cdrom/sol_9_sparc/s3   ro   ""
-               /cdrom/sol_9_sparc/s2   ro   ""
-               /cdrom/sol_9_sparc/s1   ro   ""
-               /cdrom/sol_9_sparc/s0   ro   ""
#
```

Example—Making Local Diskettes Available to Other Systems

The following example shows how to make any local diskette available to other systems on the network.

```
# ps -ef | grep nfsd
    root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
    root 10118     1  0 08:24:39 ?        0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(Add the following line:)
share -F nfs -o ro  /dummy
# eject floppy0
# chmod 644 /etc/rmmount.conf
```

```
# vi /etc/rmmount
(Add the following line to the File System Sharing section.)
share floppy*
# chmod 444 /etc/rmmount.conf
(Load a diskette.)
# volcheck -v
media was found
# share
-               /dummy   ro   ""
-               /floppy/myfiles   rw   ""
```

Example—Making Local PCMCIA Memory Cards Available to Other Systems

The following example shows how to make any local PCMCIA memory card available to other systems on the network.

```
# ps -ef | grep nfsd
    root 10127  9986  0 08:25:01 pts/2    0:00 grep nfsd
    root 10118     1  0 08:24:39 ?        0:00 /usr/lib/nfs/nfsd -a
# mkdir /dummy
# vi /etc/dfs/dfstab
(Add the following line:)
share -F nfs -o ro  /dummy
# eject pcmem0
# chmod 644 /etc/rmmount.conf
# vi /etc/rmmount
(Add the following line to the File System Sharing section:)
share floppy*
# chmod 444 /etc/rmmount.conf
(Load a PCMCIA memory card.)
# volcheck -v
media was found
# share
-               /dummy   ro   ""
-               /pcmem/myfiles   rw   ""
```

▼ How to Access Removable Media on Remote Systems

You can access media on a remote system by mounting it manually into your file system, provided the other system has shared its media according to the instructions in "How to Make Local Media Available to Other Systems" on page 232.

1. **Select an existing directory to serve as the mount point or create one.**

   `$ mkdir` *directory*

   directory is the name of the directory that you create to serve as a mount point for the other system's CD.

2. **Find the name of the media you want to mount.**

```
$ showmount -e system-name
export list for system-name:
/cdrom/sol_9_sparc (everyone)
```

3. **As superuser, mount the media.**

```
# mount -F nfs -o ro system-name:/media/media-name  local-mount-point
```

system-name	The name of the system whose media you will mount.
media-name	The name of the media you want to mount.
local-mount-point	The local directory onto which you will mount the remote media.

4. **Log out as superuser.**

5. **Verify that the media is mounted.**

```
$ ls /media
```

Example—Accessing CDs on Other Systems

The following example shows how to mount the CD named `sol_9_sparc` from the remote system `starbug` onto the `/cdrom` directory of the local system.

```
$ showmount -e starbug
export list for starbug:
/cdrom/sol_9_sparc (everyone)
$ su
Password: password
# mount -F nfs -o ro starbug:/cdrom/sol_9_sparc /cdrom
# exit
$ ls /cdrom
cdrom0     sol_9_sparc
```

Example—Accessing Diskettes on Other Systems

The following example shows how to mount the diskette named `myfiles` from the remote system `mars` onto the `/floppy` directory of the local system.

```
$ cd /net/mars
$ ls /floppy
floppy0     myfiles
$ su
Password: password
# mount -F nfs mars:/floppy/myfiles /floppy
# exit
```

```
$ ls /floppy
myfiles
```

Example—Accessing PCMCIA Memory Cards on Other Systems

The following example shows how to mount the PCMCIA memory card named `myfiles` from the remote system `mars` onto the `/pcmem` directory of the local system.

```
$ cd /net/mars
$ ls /pcmem
pcmem0     myfiles
$ su
Password: password
# mount -F nfs mars:/pcmem/myfiles /pcmem
# exit
$ ls /pcmem
myfiles
```

CHAPTER 19

Formatting Removable Media (Tasks)

This chapter describes how to format removable media from the command line in the Solaris environment.

For information on the procedures associated with formatting removable media, see "Formatting Removable Media (Task Map)" on page 239.

For background information on removable media, see Chapter 17.

Formatting Removable Media (Task Map)

Task	Description	For Instructions
1. Load unformatted media	Insert the media into the drive and enter the `volcheck` command.	"How to Load a Removable Media" on page 242
2. Format the media	Format removable media.	"How to Format Removable Media (`rmformat`)" on page 244
3. (Optional.) Add a UFS file system	Add a UFS file system to use the diskette for transferring files.	"How to Format Removable Media for Adding a File System" on page 244

Task	Description	For Instructions
4. (Optional) Check the media	Verify the integrity of the file system on the media.	"How to Check a File System on Removable Media" on page 246
5. (Optional) Repair bad blocks on the media	Repair any bad blocks on the media, if necessary.	"How to Repair Bad Blocks on Removable Media" on page 247
6. (Optional) Apply Read or Write and Password Protection	Apply read or write protection or password protection on the media, if necessary.	"How to Enable or Disable Write Protection on Removable Media" on page 247

Formatting Removable Media Overview

The `rmformat` command is a non-superuser utility that you can use to format and protect rewritable removable media. The `rmformat` command has three formatting options:

- `quick` – This option formats removable media without certification or with limited certification of certain tracks on the media.
- `long` – This option formats removable media completely. For some devices, the use of this option might include the certification of the whole media by the drive itself.
- `force` – This option formats completely without user confirmation. For media with a password-protection mechanism, this option clears the password before formatting. This feature is useful when a password is forgotten. On media without password protection, this option forces a long format.

Formatting Removable Media Guidelines

Keep the following in mind when formatting removable media:

- Close and quit the file manager window.

  File Manager automatically displays a formatting window when you insert an unformatted media. To avoid the window, quit from File Manager. If you prefer to keep File Manager open, quit the formatting window when it appears.
- Volume manager (`vold`) mounts file systems automatically so you might have to unmount media before you can format it, if it contains an existing file system.

Removable Media Hardware Considerations

This section describes removable media hardware considerations.

Diskette Hardware Considerations

Keep the following in mind when formatting diskettes:

- For information on diskette names, see Table 18–1.
- Diskettes that are not named (that is, they have no "label") are assigned the default name of `noname`.

A Solaris system can format diskettes for use on both Solaris and DOS systems. However, the hardware platform imposes some limitations. These limitations are summarized in the following table.

Platform Type	Diskettes Format Type
SPARC based systems	UFS
	MS-DOS or NEC-DOS (PCFS)
	UDFS
x86 based systems	UFS
	MS-DOS or NEC-DOS (PCFS)
	UDFS

Diskettes formatted for UFS are restricted to the hardware platform on which they were formatted. In other words, a UFS diskette formatted on a SPARC based platform cannot be used for UFS on an x86 platform, nor can a diskette formatted on an x86 platform be used on a SPARC based platform. This is because the SPARC and x86 UFS formats are different. SPARC uses little-endian bit coding, x86 uses big-endian.

A complete format for SunOS file systems consists of the basic "bit" formatting plus the structure to support a SunOS file system. A complete format for a DOS file system consists of the basic "bit" formatting plus the structure to support either an MS-DOS or an NEC-DOS file system. The procedures required to prepare a diskette for each type of file system are different. Therefore, before you format a diskette, consider which procedure to follow. For more information, see "Formatting Removable Media (Task Map)" on page 239.

On a Solaris system (either SPARC or x86), you can format diskettes with the following densities.

Diskette Size	Diskette Density	Capacity
3.5″	High Density (HD)	1.44 Mbytes
3.5″	Double Density (DD)	720 Kbytes

By default, the diskette drive formats a diskette to a like density. This default means that a 1.44 Mbyte drive attempts to format a diskette for 1.44 Mbytes, whether the diskette is in fact a 1.44 Mbyte diskette or not, unless you instruct it otherwise. In other words, a diskette can be formatted to its capacity or lower, and a drive can format to its capacity or lower.

PCMCIA Memory Card Hardware Considerations

A Solaris platform can format PCMCIA memory cards for use on both Solaris and DOS platforms. However, the hardware platform imposes some limitations. These limitations are summarized in the following table.

Platform Type	PCMCIA Memory Cards Format Type
SPARC based systems	UFS
	MS-DOS or NEC-DOS (PCFS)
x86 based systems	UFS
	MS-DOS or NEC-DOS (PCFS)

PCMCIA memory cards formatted for UFS are restricted to the hardware platform on which they were formatted. In other words, a UFS PCMCIA memory card formatted on a SPARC platform cannot be used for UFS on an x86 platform. Likewise, PCMCIA memory cards formatted on an x86 platform cannot be used on a SPARC platform. This is because the SPARC and x86 UFS formats are different.

A complete format for UFS file systems consists of the basic “bit” formatting plus the structure to support a UFS file system. A complete format for a DOS file system consists of the basic “bit” formatting plus the structure to support either an MS-DOS or an NEC-DOS file system. The procedures required to prepare a PCMCIA memory card for each type of file system are different. Therefore, before you format a PCMCIA memory card, consider which file system you are using.

▼ How to Load a Removable Media

1. **Insert the media.**
2. **Make sure the media is formatted.**

If you aren't sure, insert it and check the status messages in the console, as described in Step 3. If you need to format the diskette, go to "How to Format Removable Media (rmformat)" on page 244.

3. **Notify volume management.**

```
$ volcheck -v
media was found
```

Two status messages are possible:

`media was found`	Volume management detected the media and will attempt to mount it in the directory described in Table 18–1.
	
	If the media is formatted properly, no error messages appear in the console.
	
	If the media is not formatted, the "`media was found`" message is still displayed, but the error messages similar to the following appear in the Console:
	
	`fd0: unformatted diskette or no diskette in the drive`
	
	`fd0: read failed (40 1 0)`
	
	`fd0: bad format`
	
	You must format the media before volume management can mount it. For more information, see Chapter 19.

`no media was found`	Volume management did not detect the media. Make sure the media is inserted properly and run volcheck again. If unsuccessful, check the media, it could be damaged. You can also try to mount the media manually.

4. **Verify that the media was mounted by listing its contents.**

For example, do the following for a diskette:

```
$ ls /floppy
floppy0 myfiles
```

As described earlier, `floppy0` is a symbolic link to the actual name of the diskette, In this case, `myfiles`. If the diskette has no name but is formatted correctly, the system will refer to it as `unnamed_floppy`.

If nothing appears under the `/floppy` directory, the diskette was either not mounted or is not formatted properly. To find out, run the `mount` command and look for the line that begins with `/floppy` (usually at the end of the listing):

`/floppy/`*name* on `/vol/dev/diskette0/`*name*

If the line does not appear, the diskette was not mounted. Check the console window for error messages.

▼ How to Format Removable Media (rmformat)

You can use the rmformat command to format the media. By default, this command creates two partitions on the media: partition 0 and partition 2 (the whole media).

1. **Verify that the volume manager is running, which means you can use the shorter nickname for the device name.**

```
$ ps -ef | grep vold
root   212     1  0   Nov 03 ?        0:01 /usr/sbin/vold
```

For information on starting vold, see "How to Restart Volume Management (vold)" on page 227. For information on identifying media device names, see "Using Removable Media Names" on page 224.

2. **Format the removable media.**

```
$ rmformat -F [ quick | long | force ] device-name
```

See the previous section for more information on rmformat formatting options.

If the rmformat output indicates bad blocks, see "How to Repair Bad Blocks on Removable Media" on page 247 for information on repairing bad blocks.

3. **(Optional) Label the removable media with an 8-character label to be used in the Solaris environment.**

```
$ rmformat -b label device-name
```

For information on creating a DOS label, see mkfs_pcfs(1M).

Examples—Formatting Removable Media

This example shows how to format a diskette.

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
.....................................................................
```

This example shows how to format a Zip drive.

```
$ rmformat -F quick /vol/dev/aliases/zip0
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
.....................................................................
```

▼ How to Format Removable Media for Adding a File System

1. **Format the media.**

```
$ rmformat -F quick device-name
```

2. **(Optional) Create an alternate Solaris partition table.**

   $ **rmformat -s** *slice-file device-name*

   A sample slice file looks like the following:

```
slices: 0 = 0, 30MB, "wm", "home" :
              1 = 30MB, 51MB :
              2 = 0, 94MB, "wm", "backup" :
              6 = 81MB, 13MB
```

3. **Become superuser.**

4. **Determine the appropriate file system type and select one of the following:**

   a. **Create a UFS file system.**

      # **newfs** *device-name*

   b. **Create a UDFS file system.**

      # **mkfs -F udfs** *device-name*

Example—Formatting a Diskette for a UFS File System

The following example shows how to format a diskette and create a UFS file system on the diskette.

```
$ rmformat -F quick /vol/dev/aliases/floppy0
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
$ su
# /usr/sbin/newfs /vol/dev/aliases/floppy0
newfs: construct a new file system /dev/rdiskette: (y/n)? y
/dev/rdiskette: 2880 sectors in 80 cylinders of 2 tracks, 18 sectors
        1.4MB in 5 cyl groups (16 c/g, 0.28MB/g, 128 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 640, 1184, 1792, 2336,
#
```

Example—Formatting a PCMCIA Memory Card for a UFS File System

The following example shows how to format a PCMCIA memory card and create a UFS file system on the card.

```
$ rmformat -F quick /vol/dev/aliases/pcmem0
$ su
# /usr/sbin/newfs -v /vol/dev/aliases/pcmem0
newfs: construct a new file system /vol/dev/aliases/pcmem0:(y/n)? y
.
.
```

```
.
#
```

Examples—Formatting Removable Media for a PCFS File System

This example shows how to create an alternate `fdisk` partition.

```
$ rmformat -F quick /dev/rdsk/c0t4d0s2:c
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
$ su
# fdisk /dev/rdsk/c0t4d0s2:c
# mkfs -F pcfs /dev/rdsk/c0t4d0s2:c
Construct a new FAT file system on /dev/rdsk/c0t4d0s2:c: (y/n)? y
#
```

This example shows how to create a PCFS file system without an `fdisk` partition.

```
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
$ su
# mkfs -F pcfs -o nofdisk,size=2 /dev/rdiskette
Construct a new FAT file system on /dev/rdiskette: (y/n)? y
#
```

▼ How to Check a File System on Removable Media

1. **Become superuser.**
2. **Identify the name service and select one of the following:**
   a. **Check a UFS file system.**

      # **fsck -F ufs** *device-name*

   b. **Check a UDFS file system.**

      # **fsck -F udfs** *device-name*

   c. **Check a PCFS file system.**

      # **fsck -F pcfs** *device-name*

Example—Checking a PCFS File System on Removable Media

The following example shows how check the consistency of a PCFS file system on media.

```
# fsck -F pcfs /dev/rdsk/c0t4d0s2
** /dev/rdsk/c0t4d0s2
** Scanning file system meta-data
** Correcting any meta-data discrepancies
1457664 bytes.
0 bytes in bad sectors.
0 bytes in 0 directories.
0 bytes in 0 files.
1457664 bytes free.
512 bytes per allocation unit.
2847 total allocation units.
2847 available allocation units.
#
```

▼ How to Repair Bad Blocks on Removable Media

You can only use the `rmformat` command to verify, analyze, and repair bad sectors that are found during verification if the drive supports bad block management. Most diskettes and PCMCIA memory cards do not support bad block management.

If the drive supports bad block management, a best effort is made to rectify the bad block. If the bad block cannot be rectified despite the best effort mechanism, a message indicates a failure to repair.

1. **Repair bad blocks on removable media.**

   `$` **`rmformat -c`** *block-numbers device-name*

   Supply the block number in decimal, octal, or hexadecimal format from a previous `rmformat` session.

2. **Verify the media.**

   `$` **`rmformat -V read`** *device-name*

Applying Read or Write and Password Protection to Removable Media

You can apply read protection or write protection and set a password on Iomega media such as Zip drives and Jaz drives.

▼ How to Enable or Disable Write Protection on Removable Media

1. **Determine whether you want to enable or disable write protection and select one of the following:**

   a. **Enable write protection.**

```
$ rmformat -w enable device-name
```

b. Disable write protection.

```
$ rmformat -w disable device-name
```

2. Verify whether the media's write protection is enabled or disabled.

```
$ rmformat -p device-name
```

▼ How to Enable or Disable Read or Write Protection and a Password on Iomega Media

You can apply a password with a maximum of 32 characters for Iomega media that support this feature. You cannot set read protection or write protection without a password on Iomega media. In this situation, you are prompted to provide a password.

You receive a warning message if you attempt to apply a password on media that does not support this feature.

1. Determine whether you want to enable or disable read protection or write protection and a password.

a. Enable read protection or write protection.

```
$ rmformat -W enable device-name
Please enter password (32 chars maximum): xxx
Please reenter password:
```

```
$ rmformat -R enable device-name
Please enter password (32 chars maximum): xxx
Please reenter password:
```

b. Disable read protection or write protection and remove the password.

```
$ rmformat -W disable device-name
Please enter password (32 chars maximum): xxx
```

```
$ rmformat -R disable device-name
Please enter password (32 chars maximum): xxx
```

2. Verify whether the media's read protection or write protection is enabled or disabled.

```
$ rmformat -p device-name
```

Examples—Enabling or Disabling Read or Write Protection

This example shows how to enable write protection and set a password on a Zip drive.

```
$ rmformat -W enable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
Please reenter password: xxx
```

This example shows how to disable write protection and remove the password on a Zip drive.

```
$ rmformat -W disable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
```

This example shows how to enable read protection and set a password on a Zip drive.

```
$ rmformat -R enable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
Please reenter password: xxx
```

This example shows to disable read protection and remove the password on a Zip drive.

```
$ rmformat -R disable /vol/dev/aliases/zip0
Please enter password (32 chars maximum): xxx
```

CHAPTER 20

Writing CDs (Tasks)

This chapter provides step-by-step instructions for writing and copying data and audio CDs with the `cdrw` command.

- "How to Restrict User Access to Removable Media with RBAC" on page 254
- "How to Identify a CD Writer" on page 254
- "How to Check the CD Media" on page 255
- "How to Create an ISO 9660 File System for a Data CD" on page 256
- "How to Create a Multi-Session Data CD" on page 257
- "How to Create an Audio CD" on page 259
- "How to Extract an Audio Track on a CD" on page 260
- "How to Copy a CD" on page 261
- "How to Erase CD-RW Media" on page 261

Working with Audio and Data CDs

This Solaris release provides the `cdrw` command, which enables you to write CD file systems in ISO 9660 format with Rock Ridge or Joliet extensions on CD-R or CD-RW media devices.

You can use the `cdrw` command to:

- Create data CDs
- Create audio CDs
- Extract audio data from an audio CD
- Copy CDs
- Erase CD-RW media

The `cdrw` command is available on the Software Supplement for the Solaris 8 Operating Environment 1/01 CD and is also part of the Solaris 9 release.

For information on recommended CD-R or CD-RW devices, go to `http://www.sun.com/io_technologies/pci/removable.html`.

For information on copying a Solaris CD, see the *Building a Bootable Jumpstart™ Installation CD-ROM* article from `http://www.sun.com/blueprints/browsesubject.html`.

CD Media Commonly Used Terms

Commonly used terms when referring to CD media are:

Term	Description
CD-R	CD read media that can be written once and after that, can only be read from.
CD-RW	CD rewritable media that can be written to and erased. CD-RW media can only be read by CD-RW devices.
ISO 9660	ISO, an acronym for Industry Standards Organization, is an organization that sets standards computer storage formats. An ISO 9660 file system is a standard CD-ROM file system that enables you to read the same CD-ROM on any major computer platform. The standard, issued in 1988, was written by an industry group named High Sierra, named after the High Sierra Hotel in Nevada. Almost all computers with CD-ROM drives can read files from an ISO 9660 file system.
Joliet extensions	Adds Windows™ file system information.
Rock Ridge extensions	Adds UNIX™ file system information. (Rock Ridge is named after the town in Blazing Saddles.) **Note –** These extensions are not exclusive. You can specify both `mkisofs -R` and `-j` options for compatibility with both systems. (See mkisofs(1M) for details).
MMC-compliant record	Acronym for Multi Media Command, which means these recorder comply with a common command set. Programs that can write to one MMC-compliant recorder should be able to write to all others.
Red Book CDDA	Acronym for Compact Disc Digital Audio, which is an industry standard method for storing digital audio on compact discs. It is also known by the term "Red Book" format. The official industry specification calls for one or more audio files sampled in 16-bit stereo sound at a sampling rate of 44.1 kilohertz (kHz).

Commonly used terms when working with the CD media are:

Term	Description
blanking	The process of erasing data from the CD-RW media.
`mkisofs`	Command for making a ISO file system to write onto a CD.
session	A complete track with lead-in and lead-out information.
track	A complete data or audio unit.

Writing Data and Audio CDs

The process of writing to a CD cannot be interrupted and needs a constant stream of data. Consider using the `cdrw -S` option to simulate writing to the media to verify if the system can provide data at a rate good enough for writing to the CD.

Write errors can be caused by one of the following:

- The media cannot handle the drive speed. For example, some media are only certified for 2x or 4x speeds.
- The system is running too many heavy processes that can starve the writing process.
- Network congestion can cause delays in reading the image if the image is on a remote system.
- The source drive might be slower than the destination drive when copying from CD-to-CD.

If any of these problems occur, you can lower the writing speed of the device with the `cdrw -p` option.

For example, simulate writing at 4x speed.

```
$ cdrw -iS -p 4 image.iso
```

You can also use the `cdrw -C` option to use the stated media capacity for copying an 80–minute CD. Otherwise, the `cdrw` command uses a default value of 74 minutes for copying an audio CD.

For more information, see `cdrw`(1) .

Restricting User Access to Removable Media with RBAC

By default, all users can access removable media in the Solaris 9 release. However, you can restrict user access to removable media by setting up a role through role based access control (RBAC). Access to removable media is restricted by assigning the role to a limited set of users.

For a discussion of using roles, see "RBAC Roles" in *System Administration Guide: Security Services*.

▼ How to Restrict User Access to Removable Media with RBAC

1. **Become superuser or assume an equivalent role.**
2. **Start the Solaris Management Console.**

   ```
   $ /usr/sadm/bin/smc &
   ```

   For more information on starting the console, see "How to Start the Solaris Management Console in a Name Service Environment" on page 61.
3. **Set up a role that includes the Device Management rights.**

   For more information, see "How to Create a Role by Using the Administrative Roles Tool" in *System Administration Guide: Security Services*.
4. **Add users who need to use the `cdrw` command to the newly created role.**
5. **Comment the following line in the `/etc/security/policy.conf` file.**

   ```
   AUTHS_GRANTED=solaris.device.cdrw
   ```

   If you do not do this step, all users still have access to the `cdrw` command, not just the members of the device management role.

   After this file is modified, the device management role members are the only users who can use the `cdrw` command. Everyone else is denied access with the following message:

   ```
   Authorization failed, Cannot access disks.
   ```

▼ How to Identify a CD Writer

Use the `cdrw -l` command to identify the CD writers on the system.

```
% cdrw -l
Looking for CD devices...
    Node              |    Connected Device            |  Device type
```

```
---------------------+--------------------------------+-----------------
 cdrom0              | YAMAHA   CRW8424S         1.0d | CD Reader/Writer
```

If you want to use a specific CD writer, use the `-d` option. For example:

```
% cdrw -a filename.wav -d cdrom2
```

Use the `cdrw -M` command to to identify whether the media is blank or whether there is an existing table of contents.

```
% cdrw -M

Device : YAMAHA   CRW8424S
Firmware : Rev. 1.0d (06/10/99)
Media is blank
%
```

▼ How to Check the CD Media

The `cdrw` command works with or without `vold` running. However, you must have superuser or role access to stop and start the `vold` daemon.

1. **Insert a CD into the CD-RW device.**

   The CD can be any CD that the device can read.

2. **Check that the CD-RW drive is connected properly by listing the device.**

```
$ cdrw -l
 Looking for CD devices...
    Node                  Connected Device                Device type
---------------------+--------------------------------+-----------------
 cdrom1              | YAMAHA   CRW8424S         1.0d | CD Reader/Writer
```

3. **(Optional) If you do not see the drive in the list, you might have to do a reconfiguration boot so that the system recognizes the device.**

```
# touch /reconfigure
# init 6
```

Or, use the following commands to add the CD-RW device without rebooting the system.

```
# drvconfig
# disks
```

Then restart `vold`.

```
# /etc/init.d/vold stop
# /etc/init.d/vold start
```

Creating a Data CD

Prepare the data first by using the `mkisofs` command to convert the file and file information into the High Sierra format used on CDs.

▼ How to Create an ISO 9660 File System for a Data CD

1. **Insert a blank CD into the CD-RW device.**
2. **Create the ISO 9660 file system on the new CD.**

   `% mkisofs -r` */pathname* `>` *cd-file-system*

`-r`	Creates Rock Ridge information and resets file ownerships to zero.
/pathname	Identifies the pathname used to create the ISO 9660 file system.
`>` *cd-file-system*	Identifies the name of the file system to be put on the CD.

3. **Copy the CD file system onto the CD.**

   `% cdrw -i` *cd-file-system*

`-i` *cd-file-system*	Specifies the image file for creating a data CD.

Example—Creating an ISO 9660 File System for a Data CD

The following example shows how to create a ISO 9660 file system for a data CD.

```
% mkisofs -r /home/dubs/ufs_dir > ufs_cd
Total extents actually written = 56
Total translation table size: 0
Total rockridge attributes bytes: 329
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 8000
56 extents written (0 Mb)
```

Then copy the CD file system onto the CD. For example:

```
% cdrw -i ufs_cd
Initializing device...done.
Writing track 1...done.
```

```
Finalizing (Can take several minutes)...done.
```

▼ How to Create a Multi-Session Data CD

This procedure describes how to put more than one session on the CD. This procedure includes an example of copying the `infoA` and `infoB` directories onto the CD.

1. **Create the file system for the first CD session.**

```
% mkisofs -o infoA -r -V my_infoA /data/infoA
Total translation table size: 0
Total rockridge attributes bytes: 24507
Total directory bytes: 34816
Path table size(bytes): 98
Max brk space used 2e000
8929 extents written (17 Mb)
```

`-o infoA`	Identifies the name of the ISO file system.
`-r`	Creates Rock Ridge information and resets file ownerships to zero.
`-V my_infoA`	Identifies a volume label to be used as the mount point by `vold`.
`/data/infoA`	Identifies the ISO image directory to create.

2. **Copy the ISO file system for the first session onto the CD.**

```
$ cdrw -iO infoA
Initializing device...done.
Writing track 1...done.
done.
Finalizing (Can take several minutes)...done.
```

`-i` *infoA*	Identifies the name of the image file to write to the CD.
`-O`	Keeps the CD open for writing.

3. **Re-insert the CD after it is ejected.**

4. **Identify the pathname of the CD media to include in the next write session.**

```
% eject -n
.
.
.
cdrom0 -> /vol/dev/rdsk/c2t4d0/my_infoA
```

Note the /vol/dev/... pathname.

5. **Identify the next writeable address on the CD to write the next session.**

```
cdrw -M /cdrom
Device : YAMAHA   CRW8424S
Firmware : Rev. 1.0d (06/10/99)

Track No. |Type    |Start address
----------+--------+-------------
 1        |Audio   |0
 2        |Audio   |33057
 3        |Data    |60887
 4        |Data    |68087
 5        |Data    |75287
Leadout   |Data    |84218

Last session start address: 75287
Next writable address: 91118
```

Note the address in the `Next writable address:` output so you can provide this when you write the next session.

6. **Create the next ISO file system for the next CD session and write it onto the CD.**

```
mkisofs -o infoB -r -C 0,91118 -M /vol/dev/rdsk/c2t4d0/my_infoA /data/infoB
Total translation table size: 0
Total rockridge attributes bytes: 16602
Total directory bytes: 22528
Path table size(bytes): 86
Max brk space used 20000
97196 extents written (189 Mb)
```

`-o` *infoB*	Identifies the name of the ISO file system.
`-r`	Creates Rock Ridge information and resets file ownerships to zero.
`-C` *0,91118*	Identifies the starting address of the first session and the next writable address.
`-M` `/vol/dev/rdsk/c2t4d0/my_infoA`	Specifies the path of the existing ISO image to be merged.
`/data/infoB`	Identifies the ISO image directory to create.

Creating an Audio CD

You can use the `cdrw` command to create audio CDs from individual audio tracks or from `.au` and `.wav` files.

The supported audio formats are:

Format	Description
sun	Sun .au files with data in Red Book CDDA format
wav	RIFF (.wav) files with data in Red Book CDDA format
cda	.cda files with raw CD audio data, which is 16–bit PCM stereo at 44.1 kHz sample rate in little-endian byte order)
aur	.aur files with raw CD data in big-endian byte order

If no audio format is specified, the `cdrw` command tries to determine the audio file format based on the file extension. The case of the characters in the extension is ignored.

▼ How to Create an Audio CD

This procedure describes how to copy audio files onto a CD.

1. **Insert a blank CD into the CD-RW device.**
2. **Change to the directory that contains the audio files.**

   `% cd` */myaudiodir*
3. **Copy the audio files onto the CD.**

   `% cdrw -a` *track1.wav track2.wav track3.wav*

   The `-a` option creates an audio CD.

Examples—Creating an Audio CD

The following example shows how to create an audio CD.

```
% cdrw -a bark.wav chirp.au meow.wav
Initializing device...done.
Writing track 1...done.
done.
Writing track 2...done.
Writing track 3...done.
done.
Finalizing (Can take several minutes)...done.
```

The following example shows how to create a multisession audio CD. The CD is ejected after the first session is written. Re-insert the CD before the next writing session.

```
$ cdrw -a0 groucho.wav chico.au harpo.wav
Initializing device...done.
Writing track 1...done.
```

```
done.
Writing track 2...done.
Writing track 3...done.
done.
Finalizing (Can take several minutes)...done.
<Re-insert CD>
$ cdrw -a zeppo.au
Initializing device...done.
Writing track 1...done.
done.
Finalizing (Can take several minutes)...done.
```

▼ How to Extract an Audio Track on a CD

Use the following procedure to extract an audio track from a CD and copy it to a new CD.

If you don't use the `cdrw -T` option to specify the audio file type, `cdrw` uses the filename extension to determine the audio file type. For example, the `cdrw` command detects that this file is a `.wav` file.

```
$ cdrw -x 1 testme.wav
```

1. **Insert a audio CD into the CD-RW device.**
2. **Extract an audio track.**

   `% cdrw -x -T` *audio-type* `1` *audio-file*

`-x`	Extracts audio data from an audio CD.
T *audio-type*	Identifies the type of audio file to be extracted. Supported audio types are `sun`, `wav`, `cda`, or `aur`.

3. **Copy the track to a new CD.**

   `$ cdrw -a` *audio-file*

Examples—Extracting and Creating Audio CDs

The following example shows how to extract the first track from an audio CD and names the file `song1.wav`.

```
% cdrw -x -T wav 1 song1.wav
Extracting audio from track 1...done.
```

This example describes how to copy a track to an audio CD.

```
% cdrw -a song1.wav
Initializing device...done.
Writing track 1...done.
Finalizing (Can take several minutes)...done.
```

▼ How to Copy a CD

This procedure describes how to extract all the tracks from an audio CD into a directory and then copy all them onto a blank CD.

Note – By default, the `cdrw` command copies the CD into the `/tmp` directory. The copying might require up to 700 Mbytes of free space. If there is insufficient space in the `/tmp` directory for copying the CD, use the `-m` option to specify an alternate directory.

---

1. **Insert an audio CD into a CD-RW device.**
2. **Extract the tracks from the audio CD.**

   ```
   % mkdir music_dir
   % cdrw -c -m music_dir
   ```

   An `Extracting audio ...` message is display for each track.

   The CD is ejected when all the tracks are extracted.
3. **Insert a blank CD and press Return.**

   After the tracks are extracted, the audio CD is ejected, and you are prompted to insert a blank CD.

Example—Copying a CD

This example describes how to copy one CD to another CD. You must have two CD-RW devices to do this task.

```
$ cdrw -c -s cdrom0 -d cdrom1
```

▼ How to Erase CD-RW Media

You have to erase existing CD-RW data before the CD can be rewritten.

1. **Erase the entire media or just the last session on the CD by selecting one of the following:**

   a. **Erase the last session only.**

```
% cdrw -d cdrom0 -b session
```

Erasing just the last session with the `-b session` option is faster than erasing the entire media with the `-b all` option. You can use the `-b session` option even if you used the `cdrw` command to create a data or audio CD in just one session.

b. Erase the entire media.

```
% cdrw -d cdrom0 -b all
```

CHAPTER 21

Managing Software Topics

This topic map lists the chapters that provide information on managing software in the Solaris environment.

Chapter 22	Provides overview information about adding and removing software products.
Chapter 23	Provides step-by-step instructions for adding and removing software packages.
Chapter 24	Provides overview information about managing Solaris patches.
Chapter 25	Provides step-by-step instructions for managing Solaris patches.

CHAPTER 22

Managing Software (Overview)

The management of software involves adding and removing software from standalone systems, servers, and their clients. This chapter describes background and other information about the various tools available for installing and managing software.

This is a list of the overview information in this chapter.

- "What's New in Software Management?" on page 265
- "Where to Find Software Management Tasks" on page 267
- "Overview of Software Packages" on page 267
- "Tools for Managing Software Packages" on page 268
- "Adding or Removing a Software Package" on page 268
- "Key Points for Adding or Removing Software Packages" on page 269
- "Guidelines for Removing Packages" on page 269
- "Avoiding User Interaction When Adding Packages" on page 270

This chapter does not describe installing the Solaris software on a new system, nor does it describe installing or upgrading a new version of the Solaris software. For information on installing or upgrading Solaris software, see *Solaris 9 Installation Guide*.

What's New in Software Management?

This section describes new software management features in the Solaris 9 release.

Signed Patches

All patches that are available for the Solaris 2.6, 7, 8, and 9 releases include a digital signature. A valid digital signature ensures that the patch has not been modified since the signature was applied.

Using signed patches is a secure method of downloading or applying patches because the patches include a digital signature that can be verified before the patch is applied to your system.

Signed patches are stored in Java archive format (JAR) files and are available from the SunSolve Online[SM] web site.

For overview information about signed patches, see Chapter 24. For troubleshooting signed patch problems, go to `http:sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/spfaq`.

Solaris Product Registry 3.0

The Solaris Product Registry 3.0 is a GUI tool that enables you to install and uninstall software packages.

For information on using this product to manage software packages, see "Adding and Removing Software With the Solaris Product Registry (GUI Tasks)" on page 276.

`prodreg` Command Enhancements

You can now use several options to the `prodreg` command to access and manage the Solaris Product Registry from the command line.

For information on using the `prodreg` command to administer software packages, see "Verifying and Removing Software With the Solaris Product Registry Command-Line Interface (`prodreg`)" on page 279.

Patch Analyzer

When you use the Solaris™ Web Start program to upgrade to a Solaris 9 Update Release, the patch analyzer performs an analysis on your system to determine which (if any) patches will be removed or downgraded by upgrading to the Solaris Update Release. You do not need to use the Patch Analyzer when you upgrade to the Solaris 9 release.

For information on using this tool when you are upgrading to a Solaris 9 update release, see "Upgrading to a Solaris Update Release (Tasks)" in *Solaris 9 Installation Guide*.

Solaris Management Console Patch Manager

The Solaris Management Console provides a new Patches Tool for managing patches. You can only use the Patches Tool to add patches to a system running the Solaris 9 release.

For information on starting the Solaris Management Console, see "How to Start the Console as Superuser or as a Role" on page 54.

Where to Find Software Management Tasks

Use this table to find step-by-step instructions for managing software.

Software Management Topics	For More Information
Installing Solaris software	*Solaris 9 Installation Guide*
Adding or removing Solaris software packages after installation	Chapter 23
Adding or removing Solaris patches after installation	Chapter 25
Troubleshooting software package problems	"Troubleshooting Software Package Problems (Tasks)" in *System Administration Guide: Advanced Administration*

Overview of Software Packages

Software management involves installing or removing software products. Sun and its third-party vendors deliver products in a form called a software *package*. (The term *packaging* generically refers to the method for distributing and installing software products to systems where the products will be used.) A package is a collection of files and directories in a defined format. This format conforms to the Application Binary Interface (ABI), which is a supplement to the System V Interface Definition. The Solaris operating environment provides a set of utilities that interpret this format and provide the means to install or remove a package or to verify its installation.

Tools for Managing Software Packages

The tools for adding and removing software packages from a system after the Solaris release is installed on a system are the following:

TABLE 22–1 Software Package Tools

Add, Remove, and Display Software Package Information With This Tool	Additional Features
The Solaris Web Start program	Launch an installer to add products included in the Solaris 9 media pack. You cannot add individual software packages.
Solaris Product Registry (GUI)	Launch an installer to add, remove, or display software product information. Use Product Registry to remove or display information about software products that were originally installed by using the Solaris Web Start program or the Solaris `pkgadd` command.
Solaris Product Registry `prodreg` Viewer (command line interface)	Use the `prodreg` command to remove or display information about software products that were originally installed by using the Solaris Web Start program or the Solaris `pkgadd` command.
Package commands (`pkgadd`, `pkgrm`, `pkginfo`)	Incorporate these commands into scripts, set up optional files to avoid user interaction or perform special checks, and copy software packages to spool directories.

Adding or Removing a Software Package

All the software management tools that are listed in Table 22–1are used to add, remove, or query information about installed software. Admintool, the Solaris Product Registry `prodreg` viewer, and the Web Start program all access install data that is stored in the Solaris Product Registry. The package tools, such as the `pkgadd` and `pkgrm` commands, also access or modify install data.

When you add a package, the `pkgadd` command uncompresses and copies files from the installation media to a local system's disk. When you remove a package, the `pkgrm` command deletes all files associated with that package, unless those files are also shared with other packages.

Package files are delivered in package format and are unusable as they are delivered. The `pkgadd` command interprets the software package's control files, and then uncompresses and installs the product files onto the system's local disk.

Although the `pkgadd` and `pkgrm` commands do not log their output to a standard location, they do keep track of the product that is installed or removed. The `pkgadd` and `pkgrm` commands store information about a package that has been installed or removed in a software product database.

By updating this database, the `pkgadd` and `pkgrm` commands keep a record of all software products installed on the system.

Key Points for Adding or Removing Software Packages

Keep the following key points in mind before you install or remove packages on your system:

- Package naming conventions – Sun packages always begin with the prefix `SUNW`, as in `SUNWaccr`, `SUNWadmap`, and `SUNWcsu`. Third-party packages usually begin with a prefix that corresponds to the company's stock symbol.
- What software is already installed – You can use the Web Start program, Solaris Product Registry `prodreg` viewer (either GUI or CLI), Admintool, or the `pkginfo` command to determine the software that is already installed on a system.
- How servers and clients share software – Clients might have software that resides partially on a server and partially on the client. In such cases, adding software for the client requires that you add packages to both the server and the client.

Guidelines for Removing Packages

You should use one of these tools to remove a package, even though you might be tempted to use the `rm` command instead. For example, you could use the `rm` command to remove a binary executable file, but that is not the same as using the `pkgrm` command to remove the software package that includes that binary executable.

Using the `rm` command to remove a package's files will corrupt the software products database. If you really only want to remove one file, you can use the `removef` command, which will update the software product database correctly so that the file is no longer a part of the package. For more information, see `removef`(1M).

If you intend to keep multiple versions of a package (for example, multiple versions of a document processing application), install new versions into a different directory than the already installed package with the `pkgadd` command. The directory where a package is installed is referred to as the base directory. You can manipulate the base directory by setting the `basedir` keyword in a special file called an administration file. For more information on using an administration file and on setting the base directory, see "Avoiding User Interaction When Adding Packages" on page 270 and `admin`(4).

Note – If you use the upgrade option when installing the Solaris software, the Solaris installation software consults the software product database to determine the products that are already installed on the system.

Avoiding User Interaction When Adding Packages

Using an Administration File

When you use the `pkgadd -a` command, the command consults a special administration file for information about how the installation should proceed. Normally, the `pkgadd` command performs several checks and prompts the user for confirmation before it actually adds the specified package. You can, however, create an administration file that indicates to the `pkgadd` command that it should bypass these checks and install the package without user confirmation.

The `pkgadd` command, by default, checks the current working directory for an administration file. If the `pkgadd` command doesn't find an administration file in the current working directory, it checks the `/var/sadm/install/admin` directory for the specified administration file. The `pkgadd` command also accepts an absolute path to the administration file.

Caution – Use administration files judiciously. You should know where a package's files are installed and how a package's installation scripts run before using an administration file to avoid the checks and prompts that the `pkgadd` command normally provides.

The following example shows an administration file that will prevent the `pkgadd` command from prompting the user for confirmation before installing the package.

```
mail=
instance=overwrite
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
```

Besides using administration files to avoid user interaction when you add packages, you can use them in several other ways. For example, you can use an administration file to quit a package installation (without user interaction) if there's an error or to avoid interaction when you remove packages with the `pkgrm` command.

You can also assign a special installation directory for a package, which you might do if you wanted to maintain multiple versions of a package on a system. To do so, set an alternate base directory in the administration file (by using the `basedir` keyword), which specifies where the package will be installed. For more information, see `admin`(4).

Using a Response File

A response file contains your answers to specific questions that are asked by an *interactive package*. An interactive package includes a `request` script that asks you questions prior to package installation, such as whether or not optional pieces of the package should be installed.

If prior to installation, you know that the package you want to install is an interactive package, and you want to store your answers to prevent user interaction during future installations of this package, you can use the `pkgask` command to save your response. For more information on this command, see `pkgask`(1M).

Once you have stored your responses to the questions asked by the `request` script, you can use the `pkgadd -r` command to install the package without user interaction.

CHAPTER 23

Managing Software (Tasks)

This chapter describes how to add, verify, and remove software packages.

This is a list of step-by-step instructions in this chapter.

- "How to Install Software With the Solaris Web Start Program" on page 275
- "How to View Installed or Uninstalled Software Information With the Product Registry (GUI Tasks)" on page 277
- "How to Install Software With the Product Registry (GUI Tasks)" on page 278
- "How to Uninstall Software With the Product Registry (GUI Tasks)" on page 279
- "How to View Installed or Uninstalled Software Information With the Product Registry (`prodreg` CLI Tasks)" on page 280
- "How to View Software Attributes With the Product Registry (`prodreg` CLI Tasks)" on page 283
- "How to Check Dependencies Between Software Components (`prodreg` CLI Tasks)" on page 285
- "How to Identify Damaged Software Products (`prodreg` CLI Tasks)" on page 286
- "How to Uninstall Software in the Solaris Product Registry (`prodreg` CLI Tasks)" on page 289
- "How to Uninstall Damaged Software With the Product Registry (`prodreg` CLI Tasks)" on page 293
- "How to Reinstall Damaged Software Components (`prodreg` CLI Tasks)" on page 295
- "How to Add Software Packages With Admintool" on page 298
- "How to Remove Software Packages With Admintool" on page 299
- "How to Add Software Packages (`pkgadd`)" on page 300
- "How to Add Software Packages to a Spool Directory (`pkgadd`)" on page 302
- "How to List Information About All Installed Packages (`pkginfo`)" on page 304
- "How to Check the Integrity of Installed Software Packages (`pkgchk`)" on page 305
- "How to Remove Software Packages (`pkgrm`)" on page 306

Commands for Managing Software Packages

The following table lists the commands to use for adding, removing, and checking the installation of software packages after the Solaris release is installed.

TABLE 23–1 Tools or Commands for Managing Software Packages

Tool or Command	Man Page	Description
`admintool`	`admintool`(1M)	Installs or removes a software package with a graphical tool
`installer`	`installer`(1M)	Installs or removes a software package with an installer
`pkgadd`	`pkgadd`(1M)	Installs a software package
`pkgchk`	`pkgchk`(1M)	Checks the installation of a software package
`pkginfo`	`pkginfo`(1)	Lists software package information
`pkgparam`	`pkgparam`(1)	Displays software package parameter values
`pkgrm`	`pkgrm`(1M)	Removes a software package
`prodreg`	`prodreg`(1M)	Browse, unregister, and uninstall software in the Solaris Product Registry

Adding Software With the Solaris Web Start Program

This section describes how to use the Solaris Web Start program to add software to a system on which you have installed the Solaris operating environment. The Solaris Web Start program installs only the components of the software groups that you skipped when you initially installed the Solaris operating environment. You cannot upgrade to another software group after installing or upgrading. For a description of the four software groups, see *Solaris 9 Installation Guide*.

▼ How to Install Software With the Solaris Web Start Program

Note – This procedure assumes that the system is running volume management (`vold`). If your system is not running volume management, see Chapter 18 for information on accessing removable media without volume management.

1. **Become superuser or assume an equivalent role.**
2. **Decide to install from a CD, a DVD, or from the network. Select one of the following:**
   - If you are installing from a CD, insert the CD into the CD-ROM drive.

     If you insert the Solaris 9 Languages CD, the Solaris Web Start program starts automatically. Proceed to Step 6.
   - If you are installing from a DVD, insert the DVD into the DVD-ROM drive.
   - If you are installing from the network, locate the net image of the software you want to install.
3. **Change directories to find the Solaris Web Start installer.**

   Solaris Web Start installers are located in various directories on the CDs and on the DVD.
   - Solaris 9 Software 1 of 2 CD.
   - Solaris 9 Software 2 of 2 CD.
   - Solaris 9 Documentation CD.
   - Solaris 9 Languages CD. The Solaris Web Start program automatically starts when the CD is inserted.

   For specific information about CD and DVD structures, see "Organization of Solaris 9 Media (Reference)" in *Solaris 9 Installation Guide*.
4. **Follow the instructions to install the software.**
   - From a file manager, double-click Installer or installer.
   - From the command line, type the following:

     ```
     % ./installer [options]
     ```

`-nodisplay`	Runs the installer without a GUI.

`-noconsole`	Runs without any interactive text console device. Use this option with the `-nodisplay` option when you include the `installer` command in a UNIX script for installing software.

Follow the instructions to install the software.

5. **Double-click Installer or installer.**
   An Installer window is displayed, followed by the Solaris Web Start dialog box.
6. **Follow the directions on the screen to install the software.**
7. **When you have finished adding software, click Exit.**
   The Solaris Web Start program exits.

Adding and Removing Software With the Solaris Product Registry (GUI Tasks)

The Solaris Product Registry is a tool to help you manage installed software. After you have installed the software, Product Registry provides a list of all the installed software by using the Solaris Web Start program 3.0 or the Solaris `pkgadd` command.

You can use the Solaris Product Registry in a GUI or with a command-line interface (CLI). For more information on how to use the Solaris Product Registry CLI, see "Verifying and Removing Software With the Solaris Product Registry Command-Line Interface (`prodreg`)" on page 279.

The Solaris Product Registry GUI interface enables you to do the following:

- View a list of installed and registered software and some software attributes
- View all Solaris system products that you installed in their localized version in the System Software Localizations directory
- Find and launch an installer
- Install additional software products
- Uninstall software and individual software packages

The Solaris Product Registry GUI main window consists of three areas of information:

- Installed, registered, and removed software
- Standard attributes of the currently selected software
- Attributes that are customized and attributes that are internal to the registered software

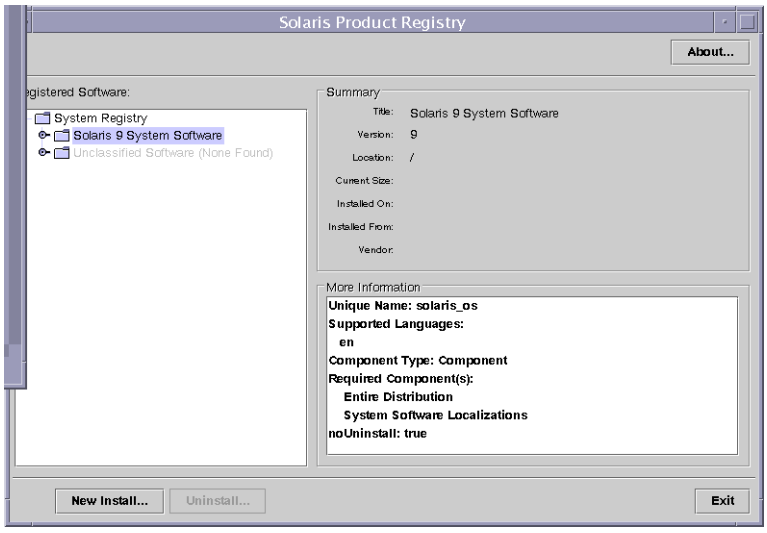


FIGURE 23–1 Solaris Product Registry Window

▼ How to View Installed or Uninstalled Software Information With the Product Registry (GUI Tasks)

1. **Become superuser or assume an equivalent role.**
2. **Start the Product Registry tool.**

   ```
   # prodreg &
   ```

   The Solaris Product Registry main window is displayed.
3. **Click the turner control to the left of the `System registry` directory in the Registered Software box.**

   Notice that the turner control changes from pointing to the right to pointing down. You can expand or collapse any item in the Registry, except an item that has a text file icon to its left.

   The Software Installed in Registered Software box always contains the following:

   - The configuration software group that you chose when installing the Solaris release. Software groups that can be displayed include Core, End User System Support, Developer System Support, Entire Distribution, or Entire Distribution Plus OEM Support.
   - Additional system software, which is Solaris products that are not part of the software group you chose.
   - Unclassified software, which is any package that you installed by using the `pkgadd` command that is not a Solaris product or part of the software group.

4. **Select directories until you find a software application to view.**

   The list expands as you open directories.

5. **To view the attributes, select a directory or file.**

   The Product Registry displays attribute information in the System Registry box.

   - For software products that were installed with the Solaris Web Start program, the Product Registry contains values for at least the following: Title, Version, Location, and Installed on. Items in an expanded list under a product or software group inherit the version information of the product.
   - If all or part of the product was removed with the `pkgrm` command, a cautionary icon appears next to the software product's name.

▼ How to Install Software With the Product Registry (GUI Tasks)

You can use Product Registry to find software and launch the Solaris Web Start program, which leads you through the installation of that software.

1. **Become superuser or assume an equivalent role.**

2. **Start the Product Registry tool.**

   ```
   # prodreg
   ```

   The Solaris Product Registry window is displayed.

3. **Decide if you are installing from a CD, a DVD, or from the network. Select one of the following:**

   - If you are installing from a CD, insert the CD into the CD-ROM drive.
   - If you are installing from a DVD, insert the DVD into the DVD-ROM drive.
   - If you are installing from the network, locate the net image of the software that you want to install.

4. **To view the list of installed and registered software, click the turner control.**

5. **Click the New Install button at the bottom of the Solaris Product Registry window.**

   The Product Registry displays the Select Installer dialog box, which initially points to the `/cdrom` directory or the directory you are in.

6. **Select directories to find the Solaris Web Start program installer.**

   Solaris Web Start installers are located in various directories on the CDs and on the DVD. For specific information about CD and DVD structures, see "Organization of Solaris 9 Media (Reference)" in *Solaris 9 Installation Guide*.

   - Solaris 9 Software 1 of 2 and 2 of 2 CD.
   - Solaris 9 Software 2 of 2 CD.

- Solaris 9 Documentation CD.
- Solaris 9 Languages CD. The Solaris Web Start program automatically starts when the CD is inserted.

7. **When you find the installer you want, select its name in the Files box.**
8. **Click OK.**
   The installer you selected is launched.
9. **Follow the directions that are displayed by the installer to install the software.**

▼ How to Uninstall Software With the Product Registry (GUI Tasks)

1. **Become superuser or assume an equivalent role.**
2. **Start the Product Registry tool.**

   ```
   # prodreg
   ```

   The Solaris Product Registry window is displayed.
3. **To view the list of installed and registered software, click the turner control.**
4. **Select directories until you find the name of the software that you want to uninstall.**
5. **Read the software attributes to make sure that this software is the software that you want to uninstall.**
6. **Click the Uninstall** *software-product-name* **button at the bottom of the Solaris Product Registry window.**
   The software product you selected is uninstalled.

Verifying and Removing Software With the Solaris Product Registry Command-Line Interface (`prodreg`)

The `prodreg` command is the command-line interface (CLI) to the Solaris Product Registry. The `prodreg` command supports several subcommands that enable you to manage the software on your system.

You can use the `prodreg` command in a terminal window to perform the following tasks.

- View a list of installed and registered software and software attributes
- View all Solaris system products that you installed in their localized version in the System Software Localizations directory
- Identify damaged software
- Remove software entries from the Solaris Product Registry
- Uninstall software and individual software packages

For more information on how to manage the Solaris Product Registry by using the command-line interface, see the man page `prodreg`(1M).

▼ How to View Installed or Uninstalled Software Information With the Product Registry (`prodreg` CLI Tasks)

You can view information about software in the Solaris Product Registry in a terminal window by using the `browse` subcommand to the `prodreg` command.

1. **Open a terminal window.**
2. **Browse the Solaris Product Registry.**

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software
```

The `browse` subcommand to the `prodreg` command displays the following information about registered software.

`BROWSE #`	When you use the `prodreg browse` command, the Solaris Product Registry generates a browse number for each registered software component. This number can be used as an argument to either the `prodreg browse` command or the `info` subcommand to descend the hierarchy of specific registered components. **Note –** Browse numbers might change when you reboot or reinstall your system. Do not store browse numbers in scripts or attempt to reuse them between separate login sessions.
`+/-/.`	This field indicates if a software component has additional software component children registered in the Solaris Product Registry. The following characters are displayed in this field. ■ `+` indicates that the software component has additional children components that are not currently displayed. ■ `-` indicates that the software component has additional children components that are currently displayed. ■ `.` indicates that the software component does not have children components.
`UUID`	This field lists the software's unique identifier in the Solaris Product Registry.
`#`	This field indicates the instance number of the software component on the system. If the system contains multiple instances of a software component, the Solaris Product Registry assigns a separate instance number to each instance of the component.
`NAME`	This field lists the localized name of the software. The name of the Solaris operating environment in this sample output is the Solaris 9 4/03 release.

3. Browse the information for one of the software components that are listed in the Solaris Product Registry.

```
% prodreg browse  -m "name"
```

`-m` "*name*"	Displays information on the software component with the name *name*

If the system contains multiple instances of *name* software, type the following command to browse the Solaris Product Registry.

```
% prodreg browse  -u name-UUID -i instance
```

`-u` *name-UUID*	Displays information on the *name* software component with the unique identifier *name-UUID*

`-i` *instance*	Displays information on *name* software component with the instance number *instance*

4. **Repeat Step 3 for each software component you want to browse.**

Example—Viewing Software Information by Component Name

The following example shows how to view software information by referencing the component's name.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg browse -m "Solaris 9 4/03 System Software"
```

Example—Viewing Software Information by Component Browse Number

The following example shows how to use the `-n` option with the `prodreg browse` command to view software information by referencing the component's browse number.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg browse -n 2
```

Example—Viewing Software Information by Component UUID

The following example shows how to use the `-u` option with the `prodreg browse` command to view software information by referencing the component's UUID.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg browse -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
```

▼ How to View Software Attributes With the Product Registry (`prodreg` CLI Tasks)

You can view specific software attributes by using the `info` subcommand to the `prodreg` command. The `prodreg info` command displays a variety of information about registered software, including the following items:

- Name of software component
- Software component description
- Required components of the software
- Other components that require the software
- Base directory of the software
- Path to the software component

1. **Open a terminal window.**
2. **Browse the Solaris Product Registry.**

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software
```

3. **View the attributes for one of the software components that are listed in the Solaris Product Registry.**

   You can view the attributes for one of the software components that are listed in the output of the `prodreg info` command in the following ways.

   ```
   % prodreg info  -m "name"
   ```

`-m` *"name"*	Displays the attributes of the software component with the name *name*

4. **Repeat Step 3 for each software component you want to view.**

Example—Viewing Software Attributes by Component Name

The following example shows how to view software attributes by referencing the component's name.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg info -m "Solaris 9 4/03 System Software"
```

Example—Viewing Software Attributes by Component Browse Number

The following example shows how to use the `-n` option with the `prodreg info` command to view software attributes by referencing the component's browse number.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software
```

```
% prodreg info -n 2
```

Example—Viewing Software Attributes by Component UUID

The following example shows how to use the `-u` option with the `prodreg info` command to view software attributes by referencing the component's UUID.

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software

% prodreg info -u a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b
```

▼ How to Check Dependencies Between Software Components (`prodreg` CLI Tasks)

You can use the `prodreg info` command to view the components that depend on a specific software component. You might want to check dependencies between software products before you uninstall specific components.

1. **Open a terminal window.**
2. **Browse the Solaris Product Registry.**

```
% prodreg browse
    BROWSE # +/-/.  UUID                                  #  NAME
    ======== =====  ====================================  =  ============
    1        -      root                                  1  System
                                                             Registry
    2         +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                             System
                                                             Software
    3         +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                             Software
```

Repeat the `prodreg browse` command until the software component you want to check is displayed in the Solaris Product Registry. See "How to View Installed or Uninstalled Software Information With the Product Registry (`prodreg` CLI Tasks)" on page 280 for more information on browsing the Solaris Product Registry with the `prodreg browse` command.

3. **View the dependencies of a specific software component.**

```
% prodreg info -m "name" -a "Dependent Components"
```

`-m` "*name*"	Displays the attributes of the software component with the name *name*.
`-a` "Dependent Components"	Displays the components that depend on *name* software by displaying the values of the Dependent Components attribute.

This command outputs a list of the software components that depend on *name* software.

Example—Viewing Components That Depend on Other Software Products

The following example shows how to view the components that depend on the software product that is named ExampleSoft.

```
% prodreg -m "ExampleSoft" -a "Dependent Components"
Dependent Components:
Name                          UUID                                  #
---------------------------  ------------------------------------  -
ExampleSoftA                  7f49ecvb-1ii2-11b2-a3f1-0800119u7e8e  1
```

▼ How to Identify Damaged Software Products (`prodreg` CLI Tasks)

If you remove installed software files or packages without using the appropriate uninstaller, you can damage the software on your system. If software is damaged, the software might not function properly. You can use the `info` subcommand to the `prodreg` command to help you determine if a software product is damaged.

1. **View the Solaris Product Registry information on the software you want to check.**

```
% prodreg browse -m name
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                        Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                        System
                                                        Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                        Software
4          -     name-UUID                              1  name
233         .    component-a-pkg                        1  component-a
```

```
234          .     component-b-pkg                    1
```

`-m` "*name*"	Displays information on the software component with the name *name*.
name-UUID	Specifies the UUID of the *name* software component.
component-a-pkg	Specifies the package name of the *component-a* component that depends on *name* software.
component-a	Specifies the name of a component that depends on *name* software.
component-b-pkg	Specifies the package name of the *component-b* component that depends on *name* software.

In the previous sample output, the *component-b-pkg* entry does not have an associated name in the `Name` column. If a software component name is not displayed in the Solaris Product Registry, the component might be damaged.

2. Verify that the software component is damaged.

```
% prodreg info -u name-UUID -i 1 -d
isDamaged=TRUE
```

`-u` *name-UUID*	Displays information on the *name* software component.
`-i` 1	Displays information on the first instance of the *name* software component.
`-d`	Displays the value of the `isDamaged` attribute of the *name* software component.

The `isDamaged=TRUE` output indicates that the *name* software component is damaged.

3. Identify the packages that form the *name-UUID* software component.

```
% prodreg info -u name-UUID -i 1 -a PKGS
pkgs:
component-a-pkg  component-b-pkg
```

4. Verify that these packages are installed on the system.

```
% pkginfo component-a-pkg
application component-a-pkg  component-a

% pkginfo component-b-pkg
ERROR: information on "component-b-pkg" was not found
```

The error message output of the `pkginfo` *component-b-pkg* command indicates that the *component-b-pkg* package has been removed from the system. The *name* software

component might not work without the *component-b-pkg* package.

Example—Identifying Damaged Software Components

The following example shows how to determine if the ExampleSoft software component is damaged.

```
% prodreg browse -m Examplesoft
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
4          -     95842091-725a-8501-ef29-0472985982be  1  ExampleSoft
233         .    90209809-9785-b89e-c821-0472985982be  1  Example Doc
234         .    EXSOzzt                               1
235         .    EXSOblob                              1  Example Data
```

The ExampleSoft child component EXSOzzt does not have an entry in the `NAME` column of the Solaris Product Registry. The ExampleSoft software might be damaged. Use the `prodreg info` command with the `-u`, `-i`, and `-d` options to determine if the ExampleSoft software is damaged.

```
% prodreg info -u 95842091-725a-8501-ef29-0472985982be -i 1 -d
isDamaged=TRUE
```

The output of the previous command indicates that the ExampleSoft software is damaged. Use the `-a PKGS` option to the `prodreg info` command to identify the ExampleSoft software packages.

```
% prodreg info
     -u 95842091-725a-8501-ef29-0472985982be
     -i 1 -a PKGS
pkgs:
EXSOzzt EXSOblob
```

Use the `pkginfo` command to verify that the EXSOzzt and EXSOblob packages are installed on the system.

```
% pkginfo EXSOzzt
ERROR: information for "EXSOzzt" was not found

% pkginfo EXSOblob
application EXSOblob       Example Data
```

The output of the `pkginfo` command indicates that the EXSOzzt package is not installed on the system.

▼ How to Uninstall Software in the Solaris Product Registry (`prodreg` CLI Tasks)

You can use the `uninstall` subcommand to the `prodreg` command to remove software from your system. When you uninstall software with the `prodreg uninstall` command, you remove a specified software and all the child components associated with that software. Before you remove software, verify that other software does not depend on the software you want to uninstall. See "How to Check Dependencies Between Software Components (`prodreg` CLI Tasks)" on page 285 for instructions on how to check software dependencies.

After you uninstall a software component, you can remove the software and all the child components of that software from the Solaris Product Registry by using the `prodreg unregister -r` command.

1. **Become superuser or assume an equivalent role.**
2. **View the information on the software you want to uninstall.**

```
# prodreg browse -u name-UUID
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
1423       -     name-UUID                              1  name
1436        .    component-a-UUID                       1   component-a
1437        -    component-b-UUID                       1   component-b
1462          .  component-c-UUID                       1   component-c
```

`-u` *name-UUID*	Displays information on the software component with the unique identifier *name-UUID*.
name	Specifies the name of the software component you want to uninstall with the unique identifier *name-UUID*.
. *component-a-UUID*	Specifies the unique identifier of the *component-a* software component that is required by *name* software.
component-a	Specifies the name of a component that is required by *name* software.
- *component-b-UUID*	Specifies the unique identifier of the *component-b* component that is required by

	name software. The - symbol indicates that *component-b* requires an additional software component.
component-b	Specifies the name of a software component that is required by *name* software.
. *component-c-UUID*	Specifies the unique identifier of the *component-b* software component that is required by *component-b* software.
component-c	Specifies the name of a software component that is required by *component-b* software.

3. **Uninstall the software.**

```
# prodreg uninstall -u name-UUID
```

4. **Check the dependencies for the software that you want to uninstall.**

```
# prodreg info -u name-UUID
Title: name
.
.
.
Child Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
component-a                  component-a-UUID                     1
component-b                  component-b-UUID                     1

Required Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
component-a                  component-a-UUID                     1
component-b                  component-b-UUID                     1
```

Check the following information in the output of the `prodreg info` command.

- Child Components – Lists the software components that are associated with the *name* software component. When you unregister the *name* software, you also unregister the child components of *name* software. If the output of the previous `prodreg info` command lists any child components, verify that you want to unregister these child components.
- Required Components – Lists the software components that are required by the *name* software component. Software components might require other components that are not child components. When you uninstall and unregister a component, only child components are unregistered and uninstalled.
- Dependent Components – Lists the components that require *name* software to run. When you unregister the *name* software, you also unregister the dependent components of *name* software. If the output of the previous `prodreg info` command lists any dependent components, verify that you want to unregister

these dependent components.

In the previous sample output, *name* software does not have any dependent components.

5. **Check the dependencies of *name* software's child components.**

```
# prodreg info -u component-a-UUID -i 1 -a "Dependent Components"
Dependent Components:
Name                          UUID                                  #
---------------------------  ------------------------------------  -
name                          name-UUID                             1

# prodreg info -u component-b-UUID -i 1 -a "Dependent Components"
Dependent Components:
Name                          UUID                                  #
---------------------------  ------------------------------------  -
name                          name-UUID                             1

# prodreg info -u component-c-UUID -i 1 -a "Dependent Components"
Dependent Components:
Name                          UUID                                  #
---------------------------  ------------------------------------  -
component-b                   component-b-UUID                      1
```

In the previous sample output, no other software depends on the child components of *name* software.

6. **Unregister the software and child components.**

```
# prodreg unregister -r -u name-UUID -i 1
```

`-r`	Recursively unregisters software with the unique identifier *name-UUID* and all the child components of this software.
`-u` *name-UUID*	Specifies the unique identifier of the software you want to unregister.
`-i` 1	Specifies the instance of the software you want to unregister.

Example—Uninstalling Software Components

The following example shows how to uninstall ExampleSoft software and all the child components of ExampleSoft software.

```
# prodreg browse -m "ExampleSoft"
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                        Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
```

```
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
1423       -     95842091-725a-8501-ef29-0472985982be  1  ExampleSoft
1436        .    90209809-9785-b89e-c821-0472985982be  1  Example Doc
1437        -    EXSOzzt                               1  Example Data
1462         .   EXSOblob                              1  Example Data

# prodreg uninstall -u 95842091-725a-8501-ef29-0472985982be -i 1

# prodreg info -u 95842091-725a-8501-ef29-0472985982be
Title: ExampleSoft Software
.
.
.
Child Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
Example Doc                 90209809-9785-b89e-c821-0472985982be  1
Example Data                EXSOzzt                               1

Required Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
Example Doc                 90209809-9785-b89e-c821-0472985982be  1
Example Data                EXSOzzt                               1

# prodreg info -u 90209809-9785-b89e-c821-0472985982be -i 1
    -a "Dependent Components"
Dependent Components:
Name                         UUID                                  #
---------------------------  ------------------------------------  -
ExampleSoft                  95842091-725a-8501-ef29-0472985982be  1

# prodreg info -u EXSOzzt -i 1 -a "Dependent Components"
Dependent Components:
Name                         UUID                                  #
---------------------------  ------------------------------------  -
ExampleSoft                  95842091-725a-8501-ef29-0472985982be  1

# prodreg info -u EXSOblob -i 1 -a "Dependent Components"
Dependent Components:
Name                        UUID                                  #
--------------------------  ------------------------------------  -
Example Data                EXSOzzt                               1

# prodreg unregister -r -u 95842091-725a-8501-ef29-0472985982be -i 1
```

▼ How to Uninstall Damaged Software With the Product Registry (`prodreg` CLI Tasks)

If you try to uninstall a damaged software component by using the `prodreg uninstall` command, the command might fail. This failure can occur if the uninstaller program for the software component has been removed from the system.

Follow these steps to uninstall a software component with no associated uninstaller program on the system.

1. **Become superuser or assume an equivalent role.**

2. **View the information on the software you want to uninstall.**

```
# prodreg browse -m "name"
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
4           -    UUID                                  1  name
1436        .    component-a-UUID                      1  component-a
1437        .    component-b-UUID                      1
```

`-m` *"name"*	Displays information on the *name* software component you want to uninstall.
UUID	Specifies the UUID of the software component you want to uninstall.
. *component-a-UUID*	Specifies the UUID of the *component-a* software component.
component-a	Specifies the name of a child software component of *name* software.
. *component-b-UUID*	Specifies the UUID of a child software component of *name* software.

The *component-b-UUID* entry does not have an associated component name. The missing name value might indicate that this component is damaged.

3. **Uninstall the software.**

```
# prodreg uninstall -u UUID -i 1
The install program requested could not be found
```

`-u` *UUID*	Specifies the UUID of the software component you want to uninstall.

`-i 1`	Specifies the instance of the software you want to uninstall.

The error message indicates that the uninstaller program is not on the system.

4. **Identify the uninstaller program for the software component.**

```
# prodreg info -m "name" -a uninstallprogram
uninstallprogram: /usr/bin/java -mx64m -classpath
uninstaller-location uninstall_name
```

`-m` "*name*"	Displays information on the *name* software component.
`-a` uninstallprogram	Displays information on the uninstaller program that is associated with the *name* software component.
uninstaller-location	Specifies the registered location of the uninstaller program for the *name* software component.

5. **Determine if the uninstaller is in the registered location.**

```
# ls uninstaller-location
uninstaller-location:
No such file or directory
```

The output of the `ls` command indicates that the uninstaller program is not in the registered location.

6. **Remove the software from the system.**

You can remove the software in one of the following ways.

- If you have a system backup available, follow these steps.
  a. Load the uninstaller program from the backup.
  b. Run the uninstaller program from a shell command-line interface such as a terminal window.
- If you do not have access to the uninstaller program on a backup, follow these steps.
  a. Unregister the software component.

```
# prodreg unregister -u UUID -i 1
```

  b. Remove any remaining registered components that are required by the software you want to remove.

```
# pkgrm component-a-UUID
```

Example—Uninstalling Damaged Software

The following example shows how to uninstall the damaged ExampleSoft software. In this example, the uninstaller program is not readily available on a system backup.

```
# prodreg browse -m Examplesoft
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
4          -     95842091-725a-8501-ef29-0472985982be  1  ExampleSoft
233         .    90209809-9785-b89e-c821-0472985982be  1  Example Doc
234         .    EXSOzzt                               1
235         .    EXSOblob                              1  Example Data

# prodreg uninstall -u 95842091-725a-8501-ef29-0472985982be -i 1
The install program requested could not be found

# prodreg info -m "ExampleSoft" -a uninstallprogram
uninstallprogram: /usr/bin/java -mx64m -classpath
/var/sadm/prod/org.example.ExampleSoft/987573587 uninstall_ExampleSoft

# ls /var/sadm/prod/org.example.ExampleSoft/987573587
/var/sadm/prod/org.example.ExampleSoft/987573587:
No such file or directory

# prodreg unregister -u 95842091-725a-8501-ef29-0472985982be -i 1

# pkgrm EXSOblob
```

▼ How to Reinstall Damaged Software Components (`prodreg` CLI Tasks)

If other software depends on a damaged software component, you might want to reinstall the damaged component, rather than uninstall the component and the other dependent software. You can use the `-f` option with the `prodreg unregister` to perform a forced unregister of the damaged component, and then reinstall the component.

1. **Become superuser or assume an equivalent role.**

2. **View the information on the software you want to reinstall.**

```
# prodreg browse -m "name"
BROWSE #  +/-/.  UUID                                  #  NAME
```

```
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                        Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                        System
                                                        Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                        Software
4           .    UUID                                  1  name
```

`-m` "*name*"	Displays information on the *name* software component you want to reinstall.
UUID	Specifies the UUID of the software component you want to reinstall.

3. Identify the software that depends on the software you want to reinstall.

```
# prodreg info -m "name" -a "Dependent Components"
Dependent Components:
Name                         UUID                                  #
---------------------------  ------------------------------------  -
component-a                   component-a-UUID                     1
```

`-m` "*name*"	Specifies the name of the software component you want to reinstall.
`-a` "Dependent Components"	Displays the components that depend on *name* software.
component-a	Specifies the name of a software component that depends on *name* software.
component-a-UUID	Specifies the UUID of the *component-a* software component.

The *component-a* software component depends on the software you want to reinstall. To reinstall *name* software and not unregister *component-a*, you must perform a forced unregister of *name* software, then reinstall *name* software.

4. Unregister only the software component you want to reinstall.

```
# prodreg unregister -f -u UUID
```

5. Reinstall the software component.

```
# /usr/bin/java -cp /usr/installers/installer
```

installer	Specifies the name of the installer program for *name* software.

Example—Reinstalling Damaged Software Components

The following example shows how to reinstall the damaged software component ComponentSoft without unregistering or uninstalling the dependent component ExampleSoft.

```
# prodreg browse -m "ComponentSoft"
BROWSE #  +/-/.  UUID                                  #  NAME
========  =====  ====================================  =  ============
1         -      root                                  1  System
                                                          Registry
2          +     a01ee8dd-1dd1-11b2-a3f2-0800209a5b6b  1  Solaris 9 4/03
                                                          System
                                                          Software
3          +     8f64eabf-1dd2-11b2-a3f1-0800209a5b6b  1  Unclassified
                                                          Software
4            .   86758449-554a-6531-fe90-4352678362fe  1  ComponentSoft

# prodreg info -m "ComponentSoft" -a "Dependent Components"
Dependent Components:
Name                         UUID                                  #
---------------------------  ------------------------------------  -
ExampleSoft                  95842091-725a-8501-ef29-0472985982be  1

# prodreg unregister -f -u 86758449-554a-6531-fe90-4352678362fe -i 1

# /usr/bin/java -cp /usr/installers/org.example.componentsoft
```

Adding and Removing Software Packages With Admintool

The Solaris operating environment includes Admintool, which is a graphical user interface for performing several administration tasks, including adding and removing software packages. Specifically, you can use Admintool to do the following:

- Add software packages to a local system
- Remove software packages from a local system
- View software already installed on the local system
- Customize software packages to be installed
- Specify an alternate installation directory for a software package

▼ How to Add Software Packages With Admintool

1. **Become superuser.**

   Unless you are a member of the `sysadmin` group (group 14), you must become superuser or assume an equivalent role to add or remove software packages with Admintool.

2. **Load a Solaris 9 Software CD or DVD into the drive.**

   Volume Manager automatically mounts the CD.

3. **Start Admintool.**

   ```
   # admintool &
   ```

   The Users window is displayed.

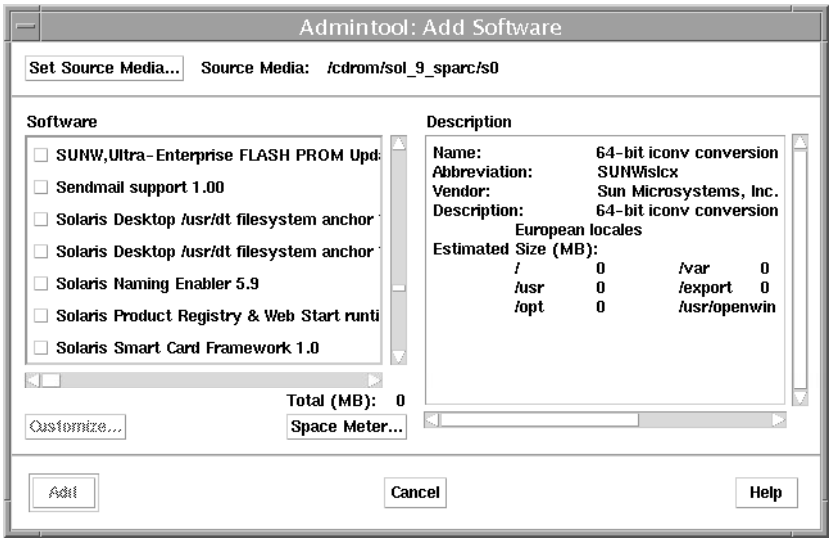
4. **Choose Software from the Browse menu.**

   The Software window is displayed.

5. **Choose Add from the Edit menu.**

   The Set Source Media window might appear. If so, specify the path to the installation media and click OK. The default path is a mounted Solaris CD.

   The Add Software window is displayed.



6. **Select the software you want to install on the local system.**

   In the Software portion of the window, click the check boxes that correspond to the software you want to install.

7. **Click Add.**

   A Command Tool window appears for each package being installed, displaying the installation output.

   The Software window is refreshed to display the packages just added.

▼ How to Remove Software Packages With Admintool

1. **Become superuser.**

   Unless you are a member of the `sysadmin` group (group 14), you must become superuser or assume an equivalent role to add or remove software packages with Admintool.

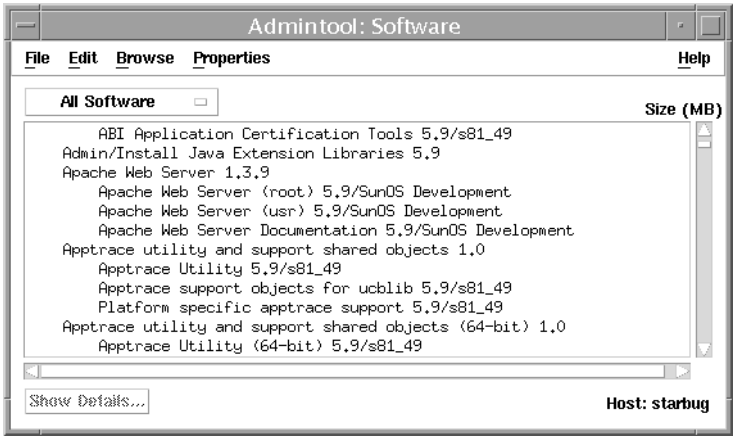
2. **Start Admintool.**

   ```
   # admintool &
   ```

   The Users window is displayed.

3. **Choose Software from the Browse menu.**

   The Software window is displayed.



4. **Select the software you want to delete from the local system.**

5. **Choose Delete from the Edit menu.**

   A warning pop-up window is displayed to confirm whether you really want to delete the software.

6. **Click Delete to confirm that you want to delete the software.**

   For each package that is being deleted, a Command Tool window is displayed that asks for confirmation, again, before deleting the software. Type `y`, `n`, or `q`. If you choose to delete the software, the output from the removal process is displayed.

Adding and Removing Software Packages With Package Commands

This section describes how to add, check, and remove packages with the package commands.

▼ How to Add Software Packages (`pkgadd`)

1. **Become superuser or assume an equivalent role.**
2. **Remove any already installed packages with the same names as the ones you are adding.**

   This step ensures that the system keeps a proper record of software that has been added and removed. There might be times when you want to maintain multiple versions of the same application on the system. For strategies on maintaining multiple software copies, see "Guidelines for Removing Packages" on page 269, and for task information, see "How to Remove Software Packages (`pkgrm`)" on page 306.
3. **Add a software package to the system.**

   # **`pkgadd -a`** *admin-file* **`-d`** *device-name pkgid* `...`

`-a` *admin-file*	(Optional) Specifies an administration file that the `pkgadd` command should consult during the installation. For details about using an administration file, see "Using an Administration File" on page 270 in the previous chapter.
`-d` *device-name*	Specifies the absolute path to the software packages. *device-name* can be the path to a device, a directory, or a spool directory. If you do not specify the path where the package resides, the `pkgadd` command checks the default spool directory (`/var/spool/pkg`). If the package is not there, the package installation fails.

pkgid	(Optional) Is the name of one or more packages, separated by spaces, to be installed. If omitted, the `pkgadd` command installs all available packages.

If the `pkgadd` command encounters a problem during installation of the package, it displays a message related to the problem, followed by this prompt:

```
Do you want to continue with this installation?
```

Respond with `yes`, `no`, or `quit`. If more than one package has been specified, type `no` to stop the installation of the package being installed. The `pkgadd` command continues to install the other packages. Type `quit` to stop the installation.

4. **Verify that the package has been installed successfully.**

   # **pkgchk -v** *pkgid*

   If no errors occur, a list of installed files is returned. Otherwise, the `pkgchk` command reports the error.

Example—Adding Software Packages From a Mounted CD

The following example shows how install the `SUNWpl5u` package from a mounted Solaris 9 CD. The example also shows how to verify that the package files were installed properly.

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product SUNWpl5u
    .
    .
    .
Installation of <SUNWpl5u> was successful.
# pkgchk -v SUNWpl5u
/usr
/usr/bin
/usr/bin/perl
/usr/perl5
/usr/perl5/5.00503
.
.
.
```

Example—Installing Software Packages From a Remote Package Server

If the packages you want to install are available from a remote system, you can manually mount the directory that contains the packages (in package format) and install packages on the local system.

The following example shows how to install software packages from a remote system. In this example, assume that the remote system named `package-server` has software packages in the `/latest-packages` directory. The `mount` command mounts the packages locally on `/mnt`, and the `pkgadd` command installs the `SUNWpl5u` package.

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt SUNWpl5u
    .
    .
    .
Installation of <SUNWpl5u> was successful.
```

If the automounter is running at your site, you do not need to mount the remote package server manually. Instead, use the automounter path, in this case, `/net/package-server/latest-packages`, as the argument to the `-d` option.

```
# pkgadd -d /net/package-server/latest-packages SUNWpl5u
    .
    .
    .
Installation of <SUNWpl5u> was successful.
```

The following example is similar to the previous example, except that it uses the `-a` option and specifies an administration file named `noask-pkgadd`, which is illustrated in "Avoiding User Interaction When Adding Packages" on page 270. In this example, assume that the `noask-pkgadd` administration file is in the default location, `/var/sadm/install/admin`.

```
# pkgadd -a noask-pkgadd -d /net/package-server/latest-packages SUNWpl5u
    .
    .
    .
Installation of <SUNWpl5u> was successful.
```

Adding a Software Package to a Spool Directory

For convenience, you can copy frequently installed packages to a spool directory. If you copy packages to the default spool directory, `/var/spool/pkg`, you do not need to specify the source location of the package (`-d` *device-name* argument) when you use the `pkgadd` command. The `pkgadd` command, by default, checks the `/var/spool/pkg` directory for any packages specified on the command line. Note that copying packages to a spool directory is not the same as installing the packages on a system.

▼ How to Add Software Packages to a Spool Directory (`pkgadd`)

1. **Become superuser or assume an equivalent role.**

2. **Remove any already spooled packages with the same names as the packages you are adding.**

   For information on removing spooled packages, see "Example—Removing a Spooled Software Package" on page 307.

3. **Add a software package to a spool directory.**

   # **pkgadd -d** *device-name* **-s** *spooldir pkgid* ...

-d *device-name*	Specifies the absolute path to the software packages. *device-name* can be the path to a device, a directory, or a spool directory.
-s *spooldir*	Specifies the name of the spool directory where the package will be spooled. You must specify a *spooldir*.
pkgid	(Optional) Is the name of one or more packages, separated by spaces, to be added to the spool directory. If omitted, the pkgadd command copies all available packages.

4. **Verify that the package has been copied successfully to the spool directory.**

   $ **pkginfo -d** *spooldir* | **grep** *pkgid*

   If *pkgid* is copied correctly, the pkginfo command returns a line of information about the *pkgid*. Otherwise, the pkginfo command returns the system prompt.

Example—Setting Up a Spool Directory From a Mounted CD

The following example shows how to transfer the SUNWman package from a mounted SPARC Solaris 9 CD to the default spool directory (/var/spool/pkg).

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product -s /var/spool/pkg SUNWman
Transferring <SUNWman> package instance
```

Example—Setting Up a Spool Directory From a Remote Software Package Server

If packages you want to copy are available from a remote system, you can manually mount the directory that contains the packages, in package format, and copy them to a local spool directory.

The following example shows the commands to do this scenario. In this example, assume that the remote system named package-server has software packages in the /latest-packages directory. The mount command mounts the package directory locally on /mnt, and the pkgadd command copies the SUNWpl5p package from /mnt to the default spool directory (/var/spool/pkg).

```
# mount -F nfs -o ro package-server:/latest-packages /mnt
# pkgadd -d /mnt -s /var/spool/pkg SUNWpl5p
```

```
Transferring <SUNWpl5p> package instance
```

If the automounter is running at your site, you do not have to mount the remote package server manually. Instead, use the automounter path, in this case, `/net/package-server/latest-packages`, as the argument to the `-d` option.

```
# pkgadd -d /net/package-server/latest-packages -s /var/spool/pkg SUNWpl5p
Transferring <SUNWpl5p> package instance
```

Example—Installing Software Packages From the Default Spool Directory

The following example shows how to install the `SUNWpl5p` package from the default spool directory. When no options are used, the `pkgadd` command searches the `/var/spool/pkg` directory for the named packages.

```
# pkgadd SUNWpl5p
    .
    .
    .
Installation of <SUNWpl5p> was successful.
```

▼ How to List Information About All Installed Packages (`pkginfo`)

List information about installed packages with the `pkginfo` command.

```
$ pkginfo
```

Example—Listing All Packages Installed

The following example shows the `pkginfo` command to list all packages installed on a local system, whether that system is a standalone or server. The output shows the primary category, package name, and the description of the package.

```
$ pkginfo
system      SUNWaccr       System Accounting, (Root)
system      SUNWaccu       System Accounting, (Usr)
system      SUNWadmap      System administration applications
system      SUNWadmc       System administration core libraries
.
.
.
```

Example—Displaying Detailed Information About Software Packages

```
$ pkginfo -l SUNWcar
   PKGINST:  SUNWcar
      NAME:  Core Architecture, (Root)
```

```
  CATEGORY:  system
      ARCH:  sparc.sun4u
   VERSION:  11.9.0,REV=2001.10.16.17.05
   BASEDIR:  /
    VENDOR:  Sun Microsystems, Inc.
      DESC:  core software for a specific hardware platform group
    PSTAMP:  crash20011016171723
  INSTDATE:  Nov 02 2001 08:53
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:    111 installed pathnames
                36 shared pathnames
                40 directories
                56 executables
              17626 blocks used (approx)
```

▼ How to Check the Integrity of Installed Software Packages (pkgchk)

1. **Become superuser or assume an equivalent role.**
2. **Check the status of an installed package.**

```
# pkgchk -a| -c -v pkgid ...
# pkgchk -d spooldir pkgid ...
```

`-a`	Specifies to audit only the file attributes, that is, the permissions, rather than the file attributes and contents, which is the default.
`-c`	Specifies to audit only the file contents, rather than the file contents and attributes, which is the default.
`-v`	Specifies verbose mode, which displays file names as they are processed.
`-d` *spooldir*	Specifies the absolute path of the spool directory.
pkgid	(Optional) Is the name of one or more packages, separated by spaces. If you do not specify a *pkgid*, all the software packages installed on the system are checked.

Example—Checking the Contents of Installed Software Packages

The following example shows how to check the contents of a package.

```
# pkgchk -c SUNWbash
```

If no errors occur, the system prompt is returned. Otherwise, the `pkgck` command reports the error.

Example—Checking the File Attributes of Installed Software Packages

The following example shows how to check the file attributes of a package.

```
# pkgchk -a SUNWbash
```

If no errors occur, the system prompt is returned. Otherwise, the `pkgck` command reports the error.

Example—Checking Software Packages Installed in a Spool Directory

The following example shows how to check a software package that was copied to a spool directory (`/export/install/packages`).

```
# pkgchk -d /export/install/packages
## checking spooled package <SUNWadmap>
## checking spooled package <SUNWadmfw>
## checking spooled package <SUNWadmc>
## checking spooled package <SUNWsadml>
```

Note – The checks made on a spooled package are limited because not all information can be audited until a package is installed.

Removing Software Packages

Use the associated tool that you used to add or install a software package to remove or uninstall a software package. For example, if you used the Web Start installer to install software, use the Web Start uninstaller to uninstall sofware.

Caution – Do not use the `rm` command to remove software packages.

▼ How to Remove Software Packages (`pkgrm`)

1. **Become superuser or assume an equivalent role.**
2. **Remove an installed package.**

   # **pkgrm** *pkgid* . . .

pkgid identifies the name of one or more packages, separated by spaces, to be removed. If omitted, `pkgrm` removes all available packages.

Example—Removing Software Packages

This example shows how to remove a package.

```
# pkgrm SUNWctu

The following package is currently installed:
   SUNWctu         Netra ct usr/platform links (64-bit)
                   (sparc.sun4u) 11.9.0,REV=2001.07.24.15.53

Do you want to remove this package? y

## Removing installed package instance <SUNWctu>
## Verifying package dependencies.
## Processing package information.
## Removing pathnames in class <none>
.
.
.
```

Example—Removing a Spooled Software Package

This example shows how to remove a spooled package.

```
# pkgrm -s /export/pkg SUNWdmfex.u
The following package is currently spooled:
   SUNWdmfex.u           Sun Davicom 10/100Mb Ethernet Driver (64-bit)
                         (sparc.sun4u) 11.9.0,REV=2001.07.24.15.53

Do you want to remove this package? y
Removing spooled package instance <SUNWdmfex.u>
```

CHAPTER 24

Managing Solaris Patches (Overview)

Patch management involves listing or adding Solaris patches from a system running the Solaris release. Patch management might also involve removing unwanted or faulty patches. Removing patches is also called *backing out* patches.

This is a list of the overview information in this chapter.

- "What Is a Patch?" on page 309
- "Accessing Solaris Patches" on page 312
- "Tools for Managing Solaris Patches" on page 313

For step-by-step instructions on adding a patch to your system, see "High-Level View of Managing Patches in the Solaris Environment (Task Map)" on page 317.

For information on adding patches to diskless client systems, see "Patching Diskless Client OS Services" on page 134.

What Is a Patch?

A patch is a collection of files and directories that replace or update existing files and directories that are preventing proper execution of the existing software. The existing software is derived from a specified *package* format, which conforms to the Application Binary Interface. For details about packages, see Chapter 22.

What Is a Signed Patch?

A *signed* patch is a patch with a digital signature. A patch with a valid digital signature ensures that the patch has not been modified since the signature was applied to the patch. Using signed patches is a secure method of downloading or adding patches because the patches include a digital signature that can be verified before the patch is added to your system.

Patches that are available for the Solaris 2.6, 7, 8, and 9 releases include a digital signature. Patches without a digital signature, or *unsigned patches*, are also available, but eventually, all patches will be *signed patches*. A valid digital signature ensures that the patch has not been modified since the signature was applied.

Signed patches are stored in Java archive format (JAR) files and are available from the SunSolve Online[SM] web site.

Using Sun's Certificates to Verify Signed Patches

Digital certificates, issued and authenticated by Sun Microsystems, are used to verify that the downloaded patch archive with the digital signature has not been compromised. These certificates are imported into your system's keystore. A keystore is a protected database that stores the keys and certificates from Sun. The `keytool` command is used to import the certificates into your system's keystore. For information on using the `keytool` command, see "How to Import Sun Certificates Into the Keystore" on page 323.

Access to a keystore is protected by a special password that you specify when you import the Sun certificates into your system's keystore.

The `SUNWcert` package contains Sun's certificate authority (CA) that you need to verify a patch's signatures. You can obtain the `SUNWcert` package from the following methods:

- **For Solaris 2.6, 7, and 8 releases –** The `SUNWcert` package is automatically installed when you download and install the Solaris 2.6, 7, or 8 patch management tools.
- **For Solaris 9 releases –** The `SUNWcert` package is automatically installed when you download and install the Solaris 9 patch management tools.
- Download from one of the following secure sites:
  - `https://uk.sunsolve.sun.com/SUNWcert/`
  - `https://de.sunsolve.sun.com/SUNWcert/`
  - `https://se.sunsolve.sun.com/SUNWcert/`
  - `https://ch.sunsolve.sun.com/SUNWcert/`
  - `https://fr.sunsolve.sun.com/SUNWcert/`
  - `https://sunsolve.sun.com/SUNWcert/`
  - `https://sunsolve.sun.co.jp/SUNWcert/`

You can verify that the certificates in the SUNWcert package match the certificate information at http://www.sun.com/pki/index.html.

SunPKI Registration Authorities

Sun Public Key Infrastructure (SunPKI) architecture is designed with one top-level certificate, called the Root CA (Certificate Authority) and a subordinate CA, which is the Sun Microsystems Inc., CA (Class B) certificate. An additional certificate issued by Sun Enterprise Services, called the patch management certificate, is used to verify the digital signatures on signed patches.

The Sun Root CA, Sun Class B CA, and the patch signing certificate are included in the SUNWcert package.

These three certificates provide a *certificate chain* of trust in the patch verification process whereby the Sun Root CA trusts the Class B CA, and the Class B CA trusts the patch management certificate. And ultimately, the GTE CyberTrust CA trusts the Sun Root CA.

Sun certificates are issued by Baltimore Technologies, who recently bought GTE CyberTrust.

A certification authority certifies the relationship between public keys that are used to decrypt the digital signature with the patch and the owner of the public keys.

The Sun CA process means the following:

- Sun has issued and authenticated the digital certificates.
- The public key in the certificate is paired with a private key that is held by Sun.
- These certificates can be used for business purposes only and can be revoked or suspended if the certificate user violates Sun's certificate policy.

For more information about Sun's certificate policy, go to http://www.sun.com/pki/cps.html.

Revoked Sun Certificates

If the Sun Root or Class B certificates are stolen or lost, a revoked certificate list is posted at http://www.sun.com/pki/ca/pkismica.crl.html.

You should view this site occasionally to verify that your imported certificates are still valid. If your imported certificates are revoked, remove them from your keystore and import the replaced certificates.

If the patch signing certificate is revoked, the existing signed patches on the SunSolve web site are removed and replaced by patches with a new digital signature.

Accessing Solaris Patches

All Sun customers can access patches through the SunSolve Online℠ web site. The following table describes the various ways to access Solaris patches.

TABLE 24–1 Ways to Access Solaris Patches

Customer Type	Description
SunSpectrum contract customer	You have access to the SunSolve database of patches and patch information. They are available from the SunSolve Online web site or by using anonymous `ftp`. These patches are updated nightly.
Not a SunSpectrum contract customer	You have access to a general set of security patches and other recommended patches. These patches are available through SunSolve Online.

You can access Solaris patches from a web site or by using anonymous `ftp`.

To access patches from a web site, you need a system that is:

- Connected to the Internet
- Capable of running a web browser such as the Netscape™ software.

To access patches by anonymous `ftp`, you need a system that is:

- Connected to the Internet
- Capable of running the `ftp` program

Access patches from the SunSolve Online℠ web site by using the following URL:

```
http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access
```

You can install either a patch cluster of recommended patches or individual patches that are freely available. Patch reports are also available.

Solaris Patch Numbering

Patches are identified by unique alphanumeric strings, with the patch base code first, a hyphen, and a number that represents the patch revision number. For example, patch 108528-10 is a *patch ID* for the SunOS 5.8 kernel update patch.

Tools for Managing Solaris Patches

The following table summarizes Solaris patch management features.

Feature	`patchadd`/`patchrm` Commands	Solaris 2.6, 7, and 8 Patch Management Tools	Solaris 9 Patch Management Tools	PatchPro Interactive or PatchPro Expert
How do I get this tool?	Bundled in Solaris release (`SUNWswmt`)	Must dowload tool from http://www.sun.com/PatchPro	Must download tool from http://www.sun.com/PatchPro	Run tool from http://www.sun.com/PatchPro
Solaris release availability	Solaris 2.6, 7, 8, and 9	Solaris 2.6, 7, and 8	Solaris 9	Solaris 2.6, 7, 8, and 9
Adds signed patches?	Yes*	Yes, and automatically verifies the signed patch when it is downloaded	Yes, and automatically verifies the signed patch when it is downloaded	No
Adds unsigned patches?	Yes	No	Yes	Yes
GUI available?	No	No	Yes	No
Analyzes system for required patches and downloads signed or unsigned patches	No	Yes, both signed and unsigned patches	Yes, both signed and unsigned patches	Yes, unsigned patches only
Local and remote system patch support	Local	Local	Local and Remote	No
RBAC support?	No	No	Yes	No

*You can unpack a signed patch and add it to your system with the `patchadd` command, but the digital signature will be lost. For information on manually verifying a signed patch and adding it with the `patchadd` command, see `http://sunsolve.Sun.COM/patches/spag.pdf`.

Detailed information about how to install and back out a patch is provided in the `patchadd`(1M) and `patchrm`(1M) man pages. Each patch also contains a `README` file that contains information about the patch.

Solaris Patch Management Tools for Signed Patches

Solaris Patch Manager Base Version 1.0, which is the `smpatch` command, is used to manage signed patches on systems running the Solaris 2.6, 7, and 8 releases. You can use the `smpatch` command with PatchPro 2.1 to manage signed patches on systems running the Solaris 9 release.

Both signed patch tools provide the following capabilities:

- They analyze patch requirements and download signed patches on the local system only. Similar to PatchPro Expert, this tool reads the `/etc/patchpro_hdw.conf` file to determine what hardware is installed. Other than this feature, the two tools are entirely independent.
- They apply one or more signed patches in JAR format, which also authenticates the patch or patches to be added.
- They remove one or more patches, which checks patch dependencies before removing the patch or patches.
- You can set up a default patch policy that allows the installation of various patch types such as `clientroot`, `clientusr`, `rebootafter`, or `standard` patches.
- If you upgrade to the Solaris 9 release, the `smpatch` command is automatically upgraded to the lastest version.

The `patchadd` command is still available to add unsigned patches to systems running the Solaris 2.6, 7, 8, and 9 releases. You cannot use Patch Manager Base Version 1.0 to add unsigned patches on these systems.

Restrictions When Using Solaris 2.6, 7, or 8 Signed Patch Tools

The Solaris 2.6, 7, and 8 signed patch tools limitations are:

- You cannot install signed patches to alternate boot environments nor to diskless clients.
- You cannot install patches that do not have a digital signature.
- You cannot install patches with the `rebootimmediate`, `reconfigimmediate`, or `nonconforming` attributes.

Package Requirements for Solaris Patch Management Tools

When you install the patch management tools, several Solaris packages are added to your system, including some Java packages, that are required for the tools to run. In addition, several packages must be installed on your system before you can install the patch tools. These packages are as follows:

- **Solaris 2.6 release –** Core cluster plus the `SUNWmfrun`, `SUNWlibC`, and `SUNWxcu4` packages.
- **Solaris 7 and 8 releases –** Core cluster plus the `SUNWmfrun` and `SUNWlibC` packages.
- **Solaris 9 release –** Developer cluster (`SUNWprog`) is required if you are using the Solaris Management Console Patches Tool with PatchPro 2.1.

For information on verifying whether the required Solaris packages are installed on your system, see "How to Verify Package Requirements for Signed Patch Tools" on page 321.

Downloading the Solaris Patch Management Tools

You can download the Solaris patch management tools from the following location:

`https://www.sun.com/PatchPro`

Follow the links for your Solaris release and select the appropriate `tar` file.

Selecting the Best Method for Adding Signed Patches

After you have installed a patch management tool, you can use several different methods of downloading or adding a signed patch or patches to your system. Use the following table to determine which method is best for your needs.

Command or Tool	Description	For More Information
`smpatch update`	Use this command to identify required patches, and then, automatically download and add the patches to your system.	`smpatch`(1M)
`smpatch analyze`	Use this command to identify required patches and display a list of required patch IDs for your system. Then, you could use the `smpatch download` and `smpatch add` commands to download and add the patches to your system.	`smpatch(1M)`
`smpatch download` and `smpatch add`	Use these commands to download and add a patch or patches to your system. These commands also download and add any prerequisite patches.	"Examples—Downloading and Adding a Signed Patch on a Solaris System (`smpatch` Command)" on page 327
`ftp` and `smpatch add`	Use the `ftp` command to transfer a patch or patches to your system. Then, use the `smpatch add` command to add the patch or patches to your system.	"Examples—Downloading and Adding a Signed Patch on a Solaris System (`smpatch` Command)" on page 327

Command or Tool	Description	For More Information
Solaris Management Console Patches Tool	**For Solaris 9 systems only –** Use this tool when you want the convenience of a GUI tool to manage signed patches.	Solaris Management Console online help

CHAPTER 25

Managing Solaris Patches (Tasks)

This chapter provides step-by-step instructions for managing patches in the Solaris environment.

This is a list of the task maps in this chapter.

- "High-Level View of Managing Patches in the Solaris Environment (Task Map)" on page 317
- "Preparation for Managing Signed Patches (Task Map)" on page 319
- "Managing Signed Patches on a Solaris System (Task Map)" on page 326
- "Managing Unsigned Solaris Patches (Task Map)" on page 332

For overview information about managing patches in the Solaris environment, see Chapter 24.

For information on troubleshooting problems with signed patches, see `http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/spfaq`.

High-Level View of Managing Patches in the Solaris Environment (Task Map)

Use this task map to identify all the tasks for managing patches in the Solaris environment. Each task points to a series of additional tasks such as managing signed or unsigned patches.

Task	Description	For Instructions
Identify disk space requirements for patches	Identify whether your system has enough disk space to download or spool patches.	"Identifying Disk Space Requirements for Patches" on page 318
Determine if adding signed or unsigned patches	Determine whether adding signed or unsigned patches is best for your environment.	"Selecting Signed or Unsigned Patches for Your Environment" on page 319
Add a signed or unsigned patch to your system	Select one of the following:	
	Add a signed patch to a your system.	"Managing Signed Patches on a Solaris System (Task Map)" on page 326
	Add an unsigned patch to your system.	"Managing Unsigned Solaris Patches (Task Map)" on page 332

Identifying Disk Space Requirements for Patches

Keep the following disk space considerations in mind before you begin downloading or spooling patches:

- The default download directory for signed patches is `/var/sadm/spool`. Unsigned patches that are spooled are also stored in `/var/sadm/spool`.
- The patch download process might use more disk space than anticipated because multiple patches can be downloaded, if prerequisite patches are required by the patch that you downloaded.
- Signed patches are unpacked in the `/var/sadm/spool` directory before they are installed. Be sure you have enough disk space in the `/var` directory for this process.
- If your `/var` directory is not large enough to support the downloading and unpacking of signed patches, you can use the `smpatch` command with the `-d` option to specify an alternate patch download directory.
- You can safely remove the patches from the `/var/sadm/spool` directory after they are successfully downloaded and added to your system to reclaim disk space in the `/var` directory.

Selecting Signed or Unsigned Patches for Your Environment

The key factor in determining when to add signed or unsigned patches is whether or not the secure download of patches is important in your environment. If the secure download of patches is important in your environment, then add signed patches to your system.

Preparation for Managing Signed Patches (Task Map)

Use this map to identify all the preparation tasks that are required before you can add signed patches to your system.

Task	Description	For Instructions
1. Verify Solaris package requirements	Verify that the required Solaris packages are installed on your system to support the patch tools.	"How to Verify Package Requirements for Signed Patch Tools" on page 321
2. Download and install a Solaris patch management tool	Select a Solaris patch management tool based on your Solaris release.	"How to Download and Install the Solaris Patch Management Tools" on page 321
3. Import Sun certificates into the keystore	Import and accept the Sun certificates that are used to verify a patch's signature. The `SUNWcert` package is automatically installed when you install the signed patches tool. Do not install the `SUNWcert` package separately if you have already installed a signed patches tool.	"How to Import Sun Certificates Into the Keystore" on page 323
4. (Optional) Change the keystore password	Change the password to keep the keystore secure.	"How to Change the Keystore Password" on page 324
5. Set up your patch environment	Set up your system for adding signed patches.	"How to Set Up Your Patch Environment" on page 324

Using the Solaris Patch Management Tools

Keep the following key points in mind when using the Solaris patch management tools:

- Make sure your systems are currently up-to-date with patches, including the appropriate kernel update patches, Java patches, and the recommended patch clusters.
- You will have to manually import the Sun certificates used to verify a patch's signature after installing the Solaris patch management tools.
- **Solaris 2.6, 7, or 8 only** – If you have previous versions of the PatchPro software on your system, the older versions will be upgraded when Solaris Patch Manager Base Version 1.0 is installed.
- Install patches on a quiet system, preferably in single-user mode.
- Signed patches are verified when they are downloaded with the `smpatch download` command.

  However, on a Solaris 9 system, no patch signature validation message is displayed during the patch download, even if the patch signature is successfully verified. If the patch signature verification fails, then the patch is not downloaded to your system.
- **Solaris 9 only** – The `smpatch` command prompts you for authentication information if you do not specify the authentication information in the `smpatch` command line.

  For example, you can specify authentication information to the `smpatch` command using the following syntax:

  ```
  # smpatch add -p mypassword -u root -- -i patch-ID
  ```

  The `smpatch` subcommands (`add`, `analyze`, `download`, or `remove`) and the authentication options and arguments are separated by the subcommand arguments with --. Or, you can let the `smpatch` command prompt you for the authentication information. For example:

  ```
  # /usr/sadm/bin/smpatch add -i patch-ID
  Authenticating as user: root

  Type /? for help, pressing <enter> accepts the default denoted by [ ]
  Please enter a string value for: password ::
  Loading Tool: com.sun.admin.patchmgr.cli.PatchMgrCli from starbug
  Login to starbug as user root was successful.
  Download of com.sun.admin.patchmgr.cli.PatchMgrCli from starbug was
  successful.
  ```
- Use the `/opt/SUNWppro/bin/uninstallpatchpro` script if you need to uninstall PatchPro 2.1. Do not attempt to remove PatchPro2.1 using this script if your current directory is `/opt/SUNWppro/bin`. Set your path as described in "How to Set Up Your Patch Environment" on page 324 and then run the `uninstallpatchpro` script from the root (`/`) directory, for example.

How to Verify Package Requirements for Signed Patch Tools

Make sure that you have the required Solaris packages installed on your system before you install the signed patch tools. If you are running the Solaris 2.6, 7, or 8 release, you need a minimal system configuration plus some additional packages. If you are running the Solaris 9 release, you must have the Developer cluster (SUNWCprog) installed on your system to use the signed patch tools.

1. **Identify your Solaris release and select one of the following:**

   a. **If you are running the Solaris 2.6 release, identify whether the required packages are installed on your system:**

   ```
   # pkginfo | grep SUNWmfrun
   system      SUNWmfrun      Motif RunTime Kit
   # pkginfo | grep SUNWlibC
   system      SUNWlibC       Sun Workshop Compilers Bundled libC
   # pkginfo | grep SUNWxcu4
   system      SUNWxcu4       XCU4 Utilities
   ```

   b. **If you are running the Solaris 7 or 8 releases, identify whether the required packages are installed on your system:**

   ```
   # pkginfo | grep SUNWmfrm
   system      SUNWmfrun      Motif RunTime Kit
   # pkginfo | grep SUNWlibC
   system      SUNWlibC       Sun Workshop Compilers Bundled libC
   ```

   c. **If you are running the Solaris 9 release, verify that the required Developer cluster is installed on your system:**

   ```
   # cat /var/sadm/system/admin/CLUSTER
   CLUSTER=SUNWCprog
   ```

2. **If the `pkginfo` commands do not return any output, you need to install the required packages.**

▼ How to Download and Install the Solaris Patch Management Tools

1. **Become superuser.**

2. **Follow the links and download the appropriate tar file for your Solaris release from the following location:**

   `http://www.sun.com/PatchPro`

3. **Select one of the following to unpack the patch tool package:**

a. If you are running the Solaris 2.6 or 7 release, uncompress and unpack the package by using the following commands:

```
# uncompress SUNWpkg-name.tar.Z
# tar xvf SUNWpkg-name.tar
```

b. If you are running the Solaris 8 or 9 release, unpack the package by using the following command:

```
# gunzip -dc SUNWpkg-name.tar.gz | tar xvf -
```

4. Run the install script.

```
# cd unzipped-pkg-dir
# ./setup
```

If there are errors while running the install script, see "Identifying Problems With Signed Patches" on page 330.

Examples—Downloading and Installing Solaris Patch Management Tools

This example shows how to download and install the Solaris 2.6 patch management tools.

```
# uncompress pproSunOSsparc5.6jre2.1.tar.Z
# tar xvf pproSunOSsparc5.6jre2.1.tar
.
.
.
# cd pproSunOSsparc5.6jre2.1
# ./setup
.
.
.
```

This example shows how to download and install the Solaris 9 patch management tools.

```
# gunzip -dc pproSunOSsparc5.9jre2.1.tar.gz | tar xvf -
.
.
# cd pproSunOSsparc5.9jre2.1
# ./setup
.
.
.
```

▼ How to Import Sun Certificates Into the Keystore

Use the `keytool` command to import and verify the Sun certificates that are used to verify the signed patches you want to add to your system. You must do this task even if you imported the certificates from a previous installation.

Note – The `SUNWcert` package is automatically installed when you install the signed patches tool. Do not install the `SUNWcert` package separately if you have already installed a signed patches tool.

---

1. **Verify that you have completed the prerequisite task, which is to download one of the Solaris patch management tools.**
2. **Become superuser.**
3. **Determine the fingerprints of your Sun root certificate and Sun class B certificate.**

```
# /usr/j2se/bin/keytool -printcert -file /etc/certs/SUNW/smirootcacert.b64
# /usr/j2se/bin/keytool -printcert -file /etc/certs/SUNW/smicacert.b64
```

4. **Verify that the output of these commands matches the Sun root and class B certificate fingerprints displayed at:**

```
https://www.sun.com/pki/ca/
```

5. **Accept the Sun class B certificate by importing it into your system:**

```
# /usr/j2se/bin/keytool -import -alias smicacert -file /etc/certs/SUNW/
smicacert.b64  -keystore /usr/j2se/jre/lib/security/cacerts
Enter keystore password:  changeit
Owner: O=Sun Microsystems Inc, CN=Sun Microsystems Inc CA (Class B)
Issuer: CN=Sun Microsystems Inc Root CA, O=Sun Microsystems Inc, C=US
Serial number: 1000006
Valid from: Mon Nov 13 12:23:10 MST 2000 until: Fri Nov 13 12:23:10 MST 2009
Certificate fingerprints:
         MD5:  B4:1F:E1:0D:80:7D:B1:AB:15:5C:78:CB:C8:8F:CE:37
         SHA1: 1E:38:11:02:F0:5D:A3:27:5C:F9:6E:B1:1F:C4:79:95:E9:6E:D6:DF
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

6. **Accept the Sun root certificate by importing it into your system:**

```
# /usr/j2se/bin/keytool -import -alias smirootcacert -file /etc/certs/SUNW/
smirootcacert.b64 -keystore /usr/j2se/jre/lib/security/cacerts
Enter keystore password:  changeit
Owner: CN=Sun Microsystems Inc Root CA, O=Sun Microsystems Inc, C=US
Issuer: CN=GTE CyberTrust Root, O=GTE Corporation, C=US
Serial number: 200014a
Valid from: Tue Nov 07 15:39:00 MST 2000 until: Thu Nov 07 16:59:00 MST 2002
Certificate fingerprints:
         MD5:  D8:B6:68:D4:6B:04:B9:5A:EB:34:23:54:B8:F3:97:8C
```

```
         SHA1: BD:D9:0B:DA:AE:91:5F:33:C4:3D:10:E3:77:F0:45:09:4A:E8:A2:98
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

7. **Accept the patch signing certificate by importing it into your system:**

```
# /usr/j2se/bin/keytool -import -alias patchsigning -file /opt/SUNWppro/
etc/certs/patchsigningcert.b64 -keystore /usr/j2se/jre/lib/security/cacerts
Enter keystore password:  changeit
Owner: CN=Enterprise Services Patch Management, O=Sun Microsystems Inc
Issuer: O=Sun Microsystems Inc, CN=Sun Microsystems Inc CA (Class B)
Serial number: 1400007b
Valid from: Mon Sep 24 14:38:53 MDT 2001 until: Sun Sep 24 14:38:53 MDT 2006
Certificate fingerprints:
         MD5:  6F:63:51:C4:3D:92:C5:B9:A7:90:2F:FB:C0:68:66:16
         SHA1: D0:8D:7B:2D:06:AF:1F:37:5C:0D:1B:A0:B3:CB:A0:2E:90:D6:45:0C
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

▼ How to Change the Keystore Password

1. **Become superuser.**
2. **Change the keystore password.**

```
# /usr/j2se/bin/keytool -storepasswd -keystore /usr/j2se/jre/lib/security/
cacerts
Enter keystore password:  changeit
New keystore password:  new-password
Re-enter new keystore password:  new-password
```

▼ How to Set Up Your Patch Environment

1. **Become superuser.**
2. **Add patch tool directories to your path.**

```
# PATH=/usr/sadm/bin:/opt/SUNWppro/bin:$PATH
# export PATH
```

3. **(Optional) Identify the hardware on your system so that you can use the `smpatch analyze` command to determine whether you need specific patches based on your hardware configuration.**

```
# pprosetup -H

Change Hardware Configuration.
Analyzing this computer.
..............
```

This command only identifies Sun's Network Storage products.

4. **Identify the types of patches that you will be adding to the system.**

   ```
   # pprosetup -i standard:singleuser:rebootafter:reconfigafter
   ```

   This command establishes the default patch policy for your system.

5. **(Optional) If you want to add contract signed patches to your system, do the following steps to define your SunSolve username and password.**

   a. **Define your SunSolve username.**

      ```
      # pprosetup -u username
      ```

   b. **Define your SunSolve password by adding the password to the following file:**

      ```
      /opt/SUNWppro/lib/.sunsolvepw
      ```

6. **Identify a proxy server so that the patch tool can download patches to your system.**

   a. **If your system is behind a firewall, you need to define a proxy server that can access the `patchpro.sun.com` server and one of the following Sun patch servers that are used to download patches:**

      - `americas.patchmanager.sun.com` (default)
      - `emea.patchmanager.sun.com`
      - `japan.patchmanager.sun.com`

   b. **Identify the selected proxy server by using the following command:**

      ```
      # pprosetup -x proxy-server:proxy-port
      ```

      For example, if you selected `webaccess.corp.net.com` as the proxy server, the `pprosetup` command would look like this:

      ```
      # pprosetup -x webaccess.corp.net.com:8080
      ```

Where to Go From Here

If you have completed all the signed patch preparation tasks, you can now add signed patches with the patch management tools.

Managing Signed Patches on a Solaris System (Task Map)

Task	Description	For Instructions
1. Perform signed patches preparation tasks	Perform all of the signed patches preparation tasks: ■ Verify package requirements for signed patch tools. ■ Download and install the Solaris patch management tools. ■ Import Sun certificates into the keystore. ■ Change the keystore password. ■ Set up your system for adding signed patches.	"Preparation for Managing Signed Patches (Task Map)" on page 319
2. Download and add a signed patch or patches	Download and add a signed patch with the `smpatch` command.	"How to Download and Add a Signed Patch on a Solaris System" on page 326
3. (Optional) Remove a signed patch	If necessary, remove an unsigned patch from your system.	"How to Remove a Signed Patch on a Solaris System" on page 329

▼ How to Download and Add a Signed Patch on a Solaris System

1. **Make sure you have completed the preparation tasks before downloading and adding a signed patch to your system. For more information, see "Preparation for Managing Signed Patches (Task Map)" on page 319.**
2. **Become superuser.**
3. **Solaris 9 system only – Notify the Solaris Management Console server that the PatchPro packages were added to the system.**

   ```
   # /etc/init.d/init.wbem stop
   # /etc/init.d/init.wbem start
   ```

4. **Download a signed patch or patches from the SunSolve web site.**

   ```
   # smpatch download -i patch-ID

   Requested patches:
   ```

patch-ID

```
Downloading the requested patches

/var/sadm/spool/patch-ID.jar has been validated.

For downloaded patch(es) see /var/sadm/spool
```

5. **Add the signed patch.**

```
# smpatch add -i patch-ID
```

Examples—Downloading and Adding a Signed Patch on a Solaris System (`smpatch` Command)

The following example shows how to download and add a signed patch with the `smpatch` command on a Solaris 9 system.

```
# /usr/sadm/bin/smpatch download -i 111711-01
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: xxx
Loading Tool: com.sun.admin.patchmgr.cli.PatchMgrCli from starbug
Login to starbug as user root was successful.
Download of com.sun.admin.patchmgr.cli.PatchMgrCli from starbug was
successful.


        Requested patches:
                111711-01

        Downloading the requested patches ...


For downloaded patch(es) see /var/sadm/spool.
# smpatch add -i 111711-01
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password :: xxx
Loading Tool: com.sun.admin.patchmgr.cli.PatchMgrCli from starbug
Login to starbug as user root was successful.
Download of com.sun.admin.patchmgr.cli.PatchMgrCli from starbug was
successful.

        On machine starbug ...
                Installing patch 111711-01
#
```

The following example shows how to download and add patch 105081–45 with the `smpatch` command on a Solaris 2.6 system.

```
# smpatch download -i 105407-01

Requested patches:

    105407-01

Downloading the requested patches



/var/sadm/spool/105407-01.jar has been validated.

For downloaded patch(es) see /var/sadm/spool
# smpatch add -i 105407-01

On machine "earth/172.20.27.27" ...


Installing patch 105407-01 ...
Purging /var/sadm/spool/105407-01
/var/sadm/spool/README.txt has been moved to
/var/sadm/spool/patchproSequester
```

The following example shows how to download and add patch 107081–45 with the `smpatch` command on a Solaris 7 system. This patch has two patch dependencies, which are automatically downloaded and verified.

```
# smpatch download -i 107081-45

Requested patches:

    107081-45

Downloading the requested patches

The following patches were added due to patch dependencies:
    108376-37
    107656-09

/var/sadm/spool/108376-37.jar has been validated.

/var/sadm/spool/107656-09.jar has been validated.

/var/sadm/spool/107081-45.jar has been validated.

For downloaded patch(es) see /var/sadm/spool
# smpatch add -i 108376-37 -i 107656-09 -i 107081-45

On machine "venus/172.20.27.26" ...

Installing patch 108376-37 ...
Installing patch 107656-09 ...
Installing patch 107081-45 ...
Purging /var/sadm/spool/108376-37
Purging /var/sadm/spool/107656-09
```

```
Purging /var/sadm/spool/107081-45
```

The following example shows how to use the `ftp` command to get a signed Solaris 8 patch from the SunSolve Online web site and then use the `smpatch add` command to add the signed patch to the system.

```
# ftp sunsolve.sun.com
Connected to sunsolve.sun.com.
220-
220-Welcome to the SunSolve Online FTP server.
220-
220-Public users may log in as anonymous.
.
.
.
Name (sunsolve.sun.com:root): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: xxx
230-
230-SUN MICROSYSTEMS, INC.
.
.
.230 Guest login ok, access restrictions apply.
ftp> cd signed_patches
250 CWD command successful.
ftp> get 112846-01.jar /var/sadm/spool/112846-01.jar
200 PORT command successful.
150 Opening ASCII mode data connection for 112846-01.jar (22524 bytes).
226 Transfer complete.
local: /var/sadm/spool/112846-01 remote: 112846-01.jar
22613 bytes received in 0.065 seconds (341.70 Kbytes/s)
ftp> quit
# smpatch add -i 112846-01
On machine "earth/172.20.27.27" ...


Installing patch 112846-01 ...
Purging /var/sadm/spool/112846-01
```

▼ How to Remove a Signed Patch on a Solaris System

1. **Become superuser.**
2. **Remove the signed patch.**

   # **smpatch remove -i** *patch-ID*

   You cannot remove multiple patches in the same command.

Examples—Removing a Signed Patch on a Solaris System

The following example shows how to remove a signed patch on a system running the Solaris 9 release.

```
# /usr/sadm/bin/smpatch remove -- -i 111711-01
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.patchmgr.cli.PatchMgrCli from starbug
Login to starbug as user root was successful.
Download of com.sun.admin.patchmgr.cli.PatchMgrCli from starbug was
successful.

          On machine starbug ...
                Removing patch 111711-01
```

The following example shows how to remove a signed patch on a system running the Solaris 2.6 release.

```
# smpatch remove -i 105407-01

On machine "earth/172.20.27.27" ...

Removing patch 105407-01

Checking installed patches...

Backing out patch 105407-01...

Patch 105407-01 has been backed out.
```

Identifying Problems With Signed Patches

For up-to-date information on troubleshooting signed patch problems or error messages, see `http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/spfaq`.

Viewing Patch Tool Log Files

Various log files on the system can identify problems with installing patch management tools or adding signed patches.

By default, PatchPro writes to the system log file.The syslog configuration file, `/etc/syslog.conf`, identifies where the system log file resides on the system. You can instruct PatchPro to write messages to a different file on the local file system by updating the `patchpro.log.file` property in the PatchPro configuration file, `/opt/SUNWppro/etc/patchpro.conf`.

For example, if you want PatchPro to write to the `/var/tmp/patchpro.log` file, assign `/var/tmp/patchpro.log` to the `patchpro.log.file` property.

Use the following table to determine which log file might contain information about a failed installation of a patch management tool or a signed patch.

Log File	Description
`/var/tmp/ppro_install_log.`*nnn*	Identifies the success or failure of the installation of PatchPro packages and patches.
`/var/tmp/log/patchpro.log`	Identifies problems when adding a signed patch with the various patch tools.
`/var/adm/messages`	Can identify problems when adding a signed patch with the various patch tools or when the patch tools did not initialize properly.
Solaris Management Console 's Log Viewer on a Solaris 9 system	Identifies the success or failure of adding a signed patch with the Solaris Management Console's Patches tool.

▼ How to Resolve a Sequestered Patch

A patch might not install successfully if it requires prerequisite patches or if a system reboot is required to install the patch. Patches that cannot be installed by PatchPro are sequestered in the `/var/spool/pkg/patchproSequester` directory.

Review the patch `README` file to find out if there are any prerequisite patches, which are listed in a section called `REQUIRED PATCHES`.

You can either view a copy of the patch `README` from the SunSolve Online website or extract the `README` file from the JAR archive. Do not expand the JAR archive to avoid any tampering with the digital signature. Use the following procedure to safely extract the patch `README` file.

You should also review the contents of the `/var/tmp/log/patchpro.log` file to find out why a patch did not install successfully.

1. **Verify that a patch or patches were not installed by viewing the contents of the `/var/spool/pkg/patchproSequester` directory.**

```
# cd /var/spool/pkg/patchproSequester; ls
```

2. **Extract the `README` file from the JAR archive:**

   a. **First, identify the name of the `README` file. For example:**

      ```
      # /usr/j2se/bin/jar tvf 107058-01.jar | grep README
      ```

   b. **Then, extract the `README` file. For example:**

      ```
      # /usr/j2se/bin/jar xvf 107058-01.jar 107058-01/README.107058-01
      extracted: 107058-01/README.107058-01
      ```

3. **View the `README` file.**

   For example:

   ```
   # more 107058-01/README.107058-01
   ```

Managing Unsigned Solaris Patches (Task Map)

Task	Description	For Instructions
1. (Optional) Display information about unsigned patches	Display information about unsigned patches already installed on your system.	"How to Display Information About Solaris Patches" on page 333
2. Download an unsigned patch	Download an unsigned patch to your system.	"How to Download an Unsigned Solaris Patch" on page 334
3. Add an unsigned patch	Add an unsigned patch to your system.	"How to Add a Unsigned Solaris Patch" on page 334
4. (Optional) Remove an unsigned patch	If necessary, remove an unsigned patch from your system.	"How to Remove an Unsigned Solaris Patch" on page 335

Displaying Information About Unsigned Solaris Patches

Before installing patches, you might want to know more about patches that have previously been installed. The following table describes commands that provide useful information about patches that are already installed on a system.

TABLE 25–1 Commands for Solaris Patch Management

Command	Description
`patchadd -p`, `showrev -p`	Shows all patches that have been applied to a system.
`pkgparam` *pkgid* `PATCHLIST`	Shows all patches that have been applied to the package identified by *pkgid*, the name of the package. For example, `SUNWadmap`.
`patchadd -S` *Solaris-OS* `-p`	Shows all the `/usr` patches installed on an OS server.

▼ How to Display Information About Solaris Patches

Use the `patchadd -p` command to display information about patches installed on your system.

```
$ patchadd -p
```

Use the following command to verify whether a specific patch is installed on your system. For example:

```
$ patchadd -p | grep 111879
```

Adding an Unsigned Solaris Patch

You can use the `patchadd` command to add unsigned patches to servers or standalone systems. If you need to add a patch to a diskless client system, see "Patching Diskless Client OS Services" on page 134.

When you add a patch, the `patchadd` command calls the `pkgadd` command to install the patch packages from the patch directory to a local system's disk. More specifically, the `patchadd` command:

- Determines the Solaris version number of the managing host and the target host
- Updates the patch package's `pkginfo` file with information about patches obsoleted by the patch being installed, other patches required by this patch, and patches incompatible with this patch

During patch installation, the `patchadd` command keeps a log of the patch installation in the `/var/sadm/patch/`*patch-ID*`/log` file for current Solaris versions.

The `patchadd` command will not install a patch under the following conditions:

- The package is not fully installed on the host.
- The patch packages architecture differs from the system's architecture.

- The patch packages version does not match the installed package's version.
- A patch with the same base code and a higher version number is already installed.
- The patch is incompatible with another, already installed patch. Each installed patch keeps this information in its `pkginfo` file.
- The patch being installed requires another patch that is not installed

▼ How to Download an Unsigned Solaris Patch

1. **(Optional) Log in to the system where the patch will be applied.**

   Or, you can download the patch and use the `ftp` command to copy the patch to the target system.

2. **Open a web browser and go to the SunSolve Online web site:**

   `http://sunsolve.Sun.COM/pub-cgi/show.pl?target=patches/patch-access`

3. **Determine if you are going to download a specific patch or patch cluster. Then select one of the following:**

   a. **Type the patch number (*patch-ID*) in the "Find Patch" search field. Then, click on Find Patch.**

      Entering *patch-ID* downloads the latest patch revision.

      If this patch is freely available, the patch `README` is displayed. If this patch is not freely available, an `ACCESS DENIED` message is displayed.

      There are different patch numbers for SPARC and x86 systems, which are listed in the displayed patch `README`. Make sure you install the patch that matches your system architecture.

   b. **Click on a recommended patch cluster based on the Solaris release running on the system to be patched.**

4. **Click the Download Patch (*n* bytes) HTTP or FTP button.**

   After the patch or patches are downloaded successfully, close the web browser.

5. **Change to the directory that contains the downloaded patch package, if necessary.**

6. **Unzip the patch package.**

   `% unzip` *patch-ID-revision*

▼ How to Add a Unsigned Solaris Patch

1. **Become superuser.**

2. **Add the patch or patches.**

   `# patchadd` *patch-ID-revision*

3. **Verify that the patch was added successfully.**

```
# patchadd -p | grep patch-ID-revision
```

Example—Adding an Unsigned Solaris Patch

In the following example, the Solaris 8 patch, 111879–01, is added to the system. The patch had already been downloaded to the system previously.

```
# patchadd /export/Sol8patch/111879-01

Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

Patch number 111879-01 has been successfully installed.
See /var/sadm/patch/111879-01/log for details

Patch packages installed:
  SUNWwsr
# patchadd -p | grep 111879-01
Patch: 111879-01 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWwsr
```

Removing an Unsigned Solaris Patch

When you back out a patch, the `patchrm` command restores all files modified by that patch, unless:

- The patch was installed with the `patchadd -d` option, which instructs `patchadd` to not save copies of files being updated or replaced.
- The patch has been obsoleted by a later patch.
- The patch is required by another patch.

The `patchrm` command calls the `pkgadd` command to restore packages that were saved from the initial patch installation.

During the patch removal process, `patchrm` keeps a log of the back out process in `/tmp/backoutlog.`*process_id*. This log file is removed if the patch backs out successfully.

▼ How to Remove an Unsigned Solaris Patch

Use the `patchrm` command if you need to remove an unsigned Solaris patch.

1. **Become superuser.**
2. **Remove the patch.**

```
# patchrm patch-ID-revision
```

3. **Verify that the patch was removed.**

```
# patchadd -p | grep patch-ID-revision
```

Example—Removing an Unsigned Solaris Patch

The following example shows how to remove the Solaris 8 patch, 111879–01.

```
# patchrm 111879-01

Checking installed patches...

Backing out patch 111879-01...

Patch 111874-02 has been backed out.

# showrev -p | grep 111879-01
#
```

CHAPTER 26

Managing Devices Topics

This topic map lists the chapters that provide information on managing devices.

Chapter 27	Provides a high-level overview of device configuration and step-by-step instructions for displaying device information on your system.
Chapter 28	Provides step-by-step instructions for configuring devices.
Chapter 29	Provides a high-level overview of USB devices and step-by-step instructions for using USB devices.
Chapter 30	Provides an overview of device naming conventions and instructions for accessing devices.

CHAPTER 27

Managing Devices (Tasks)

This chapter provides overview information and step-by-step instructions for managing peripheral devices, such as disks, CD-ROMs, and tape devices, in the Solaris environment.

This is a list of the step-by-step instructions in this chapter.

- "How to Display System Configuration Information" on page 343
- "How to Display Device Information" on page 345
- "How to Add a Device Driver" on page 348
- "How to Add a Peripheral Device" on page 346

This is a list of the overview information in this chapter.

- "Where to Find Device Management Tasks" on page 339
- "About Device Drivers" on page 340
- "Automatic Configuration of Devices" on page 340
- "Displaying Device Configuration Information" on page 342

For information about accessing peripheral devices, see Chapter 30.

Device management in the Solaris environment usually involves adding and removing peripheral devices from systems, possibly adding a third-party device driver to support a device, and displaying system configuration information.

Where to Find Device Management Tasks

The following table describes where to find step-by-step instructions for hot-plugging devices and adding serial devices, such as printers and modems, and peripheral devices, such as a disk, CD-ROM, or tape devices.

TABLE 27–1 Where to Find Instructions for Adding a Device

Device Management Task	For More Information
Adding a disk that is not hot-pluggable	Chapter 34 or Chapter 35
Hot-plugging a SCSI or PCI device	"SCSI Hot-Plugging With the `cfgadm` Command" on page 354 or "x86: PCI Hot-Plugging With the `cfgadm` Command" on page 364
Hot-plugging a USB device	"Hot-Plugging USB Devices" on page 386
Adding a CD-ROM or tape device	"How to Add a Peripheral Device" on page 346
Adding a modem	"Managing Terminals and Modems (Overview)" in *System Administration Guide: Advanced Administration*
Adding a printer	"Managing Printing Services (Overview)" in *System Administration Guide: Advanced Administration*

About Device Drivers

A computer typically uses a wide range of peripheral and mass-storage devices. Your system, for example, probably has a disk drive, a keyboard and a mouse, and some kind of magnetic backup medium. Other commonly used devices include CD-ROM drives, printers and plotters, light pens, touch-sensitive screens, digitizers, and tablet-and-stylus pairs.

The Solaris software does not directly communicate with all these devices. Each type of device requires different data formats, protocols, and transmission rates.

A *device driver* is a low-level program that allows the operating system to communicate with a specific piece of hardware. The driver serves as the operating system's "interpreter" for that piece of hardware.

Automatic Configuration of Devices

The kernel, consisting of a small generic core with a platform-specific component and a set of modules, is configured automatically in the Solaris environment.

A kernel module is a hardware or software component that is used to perform a specific task on the system. An example of a *loadable* kernel module is a device driver that is loaded when the device is accessed.

The platform-independent kernel is `/kernel/genunix`. The platform-specific component is `/platform/‘uname -m‘/kernel/unix`.

The kernel modules are described in the following table.

TABLE 27–2 Description of Kernel Modules

Location	Directory Contents
`/platform/‘uname -m‘/kernel`	Platform-specific kernel components
`/kernel`	Kernel components common to all platforms that are needed for booting the system
`/usr/kernel`	Kernel components common to all platforms within a particular instruction set

The system determines what devices are attached to it at boot time. Then, the kernel configures itself dynamically, loading needed modules into memory. At this time, device drivers are loaded when devices, such as disk and tape devices, are accessed. This process is called *autoconfiguration* because all kernel modules are loaded automatically when they are needed.

You can customize the way in which kernel modules are loaded by modifying the `/etc/system` file. For instructions on modifying this file, see `system`(4).

Features and Benefits of Autoconfiguration

The benefits of autoconfiguration are as follows:

- Main memory is used more efficiently because modules are loaded when needed.
- There is no need to reconfigure the kernel when new devices are added to the system.
- Drivers can be loaded and tested without having to rebuild the kernel and reboot the system.

You will use autoconfiguration is used by a system administrator when you add a new device (and driver) to the system. At this time, you will perform a reconfiguration boot so that the system will recognize the new device.

What You Need for Unsupported Devices

Device drivers needed to support a wide range of standard devices are included in the Solaris environment. These drivers can be found in the `/kernel/drv` and `/platform/‘uname -m‘/kernel/drv` directories.

However, if you've purchased an unsupported device, the manufacturer should provide the software that is needed for the device to be properly installed, maintained, and administered.

At a minimum, this software includes a device driver and its associated configuration (`.conf`) file. The `.conf` files reside in the `drv` directories. This software might also include custom maintenance and administrative utilities since the device might be incompatible with Solaris utilities.

Contact your device manufacturer for more information.

Displaying Device Configuration Information

Three commands are used to display system and device configuration information.

Command	Man Page	Description
`prtconf`	`prtconf`(1M)	Displays system configuration information, including total amount of memory and the device configuration as described by the system's device hierarchy. The output displayed by this command depends upon the type of system.
`sysdef`	`sysdef`(1M)	Displays device configuration information including system hardware, pseudo devices, loadable modules, and selected kernel parameters.
`dmesg`	`dmesg`(1M)	Displays system diagnostic messages as well as a list of devices attached to the system since the last reboot.

For information on the device names that are used to identify devices on the system, see "Device Naming Conventions" on page 406.

`driver not attached` Message

The following driver-related message might be displayed by the `prtconf` and `sysdef` commands:

device, instance #*number* (driver not attached)

This message does not always mean that a driver is unavailable for this device. This message means that no driver is *currently* attached to the device instance because there is no device at this node or the device is not in use. Drivers are loaded automatically when the device is accessed and unloaded when the device is not in use.

Identifying a System's Devices

Use the output of the `prtconf` and `sysdef` commands to identify which disk, tape, and CD-ROM devices are connected to the system. The output of these commands display the `driver not attached` messages next to the device instances. Since these devices are always being monitored by some system process, the `driver not attached` message is usually a good indication that there is no device at that device instance.

For example, the following `prtconf` output identifies a device at `instance #3` and `instance #6`, which is probably a disk device at target 3 and a CD-ROM device at target 6 of the first SCSI host adapter (`esp, instance #0`).

```
$ /usr/sbin/prtconf
.
.
.

esp, instance #0
            sd (driver not attached)
            st (driver not attached)
            sd, instance #0 (driver not attached)
            sd, instance #1 (driver not attached)
            sd, instance #2 (driver not attached)
            sd, instance #3
            sd, instance #4 (driver not attached)
            sd, instance #5 (driver not attached)
            sd, instance #6
.
.
.
```

You can glean the same device information from the `sysdef` output.

▼ How to Display System Configuration Information

Use the `prtconf` command to display system configuration information.

```
# /usr/sbin/prtconf
```

Use the `sysdef` command to display system configuration information that include pseudo devices, loadable modules, and selected kernel parameters.

```
# /usr/sbin/sysdef
```

Examples—Displaying System Configuration Information

The following `prtconf` output is displayed on a SPARC based system.

```
# prtconf
System Configuration:  Sun Microsystems  sun4u
Memory size: 128 Megabytes
System Peripherals (Software Nodes):
SUNW,Ultra-5_10
    packages (driver not attached)
        terminal-emulator (driver not attached)
        deblocker (driver not attached)
        obp-tftp (driver not attached)
        disk-label (driver not attached)
        SUNW,builtin-drivers (driver not attached)
        sun-keyboard (driver not attached)
        ufs-file-system (driver not attached)
    chosen (driver not attached)
    openprom (driver not attached)
        client-services (driver not attached)
    options, instance #0
    aliases (driver not attached)
    memory (driver not attached)
    virtual-memory (driver not attached)
    pci, instance #0
        pci, instance #0
            ebus, instance #0
                auxio (driver not attached)
                power, instance #0
                SUNW,pll (driver not attached)
                se, instance #0
                su, instance #0
                su, instance #1
                ecpp (driver not attached)
                fdthree, instance #0
.
.
.
```

The following `sysdef` output is displayed from an x86 based system.

```
# sysdef
* Hostid
*
  29f10b4d
*
* i86pc Configuration
*
*
* Devices
*
```

```
+boot (driver not attached)
memory (driver not attached)
aliases (driver not attached)
chosen (driver not attached)
i86pc-memory (driver not attached)
i86pc-mmu (driver not attached)
openprom (driver not attached)
options, instance #0
packages (driver not attached)
delayed-writes (driver not attached)
itu-props (driver not attached)
isa, instance #0
    motherboard (driver not attached)
    pnpADP,1542, instance #0
    asy, instance #0
    asy, instance #1
    lp, instance #0 (driver not attached)
    fdc, instance #0
        fd, instance #0
        fd, instance #1 (driver not attached)
    kd (driver not attached)
    kdmouse (driver not attached)
.
.
.
```

▼ How to Display Device Information

Display device information with the `dmesg` command.

```
# /usr/sbin/dmesg
```

The `dmesg` output is displayed as messages on the system console and identifies which devices are connected to the system since the last reboot.

Examples—Displaying Device Information

The following `dmesg` output is displayed from a SPARC based system.

```
# dmesg
Jan  3 08:44:41 starbug genunix: [ID 540533 kern.notice] SunOS Release 5.9 ...
Jan  3 08:44:41 starbug genunix: [ID 913631 kern.notice] Copyright 1983-2002 ...
Jan  3 08:44:41 starbug genunix: [ID 678236 kern.info] Ethernet address = ...
Jan  3 08:44:41 starbug unix: [ID 389951 kern.info] mem = 131072K (0x8000000)
Jan  3 08:44:41 starbug unix: [ID 930857 kern.info] avail mem = 121888768
Jan  3 08:44:41 starbug rootnex: [ID 466748 kern.info] root nexus =
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
.
.
.
```

```
#
```

The following `dmesg` output is displayed from an x86 based system.

```
# dmesg
Jan  2 07:21:46 naboo genunix: [ID 540533 kern.notice] SunOS Release 5.9 Version ...
Jan  2 07:21:46 naboo genunix: [ID 913631 kern.notice] Copyright 1983-2002 ...
Jan  2 07:21:46 naboo genunix: [ID 897550 kern.info] Using default device ...
Jan  2 07:21:46 naboo unix: [ID 168242 kern.info] mem = 130684K (0x7f9f000)
Jan  2 07:21:46 naboo unix: [ID 930857 kern.info] avail mem = 116547584
Jan  2 07:21:46 naboo rootnex: [ID 466748 kern.info] root nexus = i86pc
Jan  2 07:21:46 naboo rootnex: [ID 349649 kern.info] pci0 at root: ...
Jan  2 07:21:46 naboo genunix: [ID 936769 kern.info] pci0 is /pci@0,0
Jan  2 07:21:46 naboo genunix: [ID 678236 kern.info] Ethernet address = ...
.
.
.
```

Adding a Peripheral Device to a System

Adding a new (non-hot-pluggable) peripheral device usually involves the following:

- Shutting down the system
- Connecting the device to the system
- Rebooting the system

Use the "How to Add a Peripheral Device" on page 346 procedure to add the following devices that are not hot-pluggable to a system:

- CD-ROM
- Secondary disk drive
- Tape drive
- SBUS card

In some cases, you might have to add a third-party device driver to support the new device.

For information on hot-plugging devices, see Chapter 28.

▼ How to Add a Peripheral Device

1. **Become superuser.**
2. **Follow steps 2 and 3 of "How to Add a Device Driver" on page 348 if you need to add a device driver to support the device.**
3. **Create the `/reconfigure` file.**

```
# touch /reconfigure
```

The `/reconfigure` file will cause the Solaris software to check for the presence of any newly installed devices the next time you turn on or boot your system.

4. **Shut down the system.**

```
# shutdown -i0 -g30 -y
```

`-i0`	Brings the system to the 0 init state, which is the appropriate state for turning the system power off for adding and removing devices.
`-g30`	Shuts the system down in 30 seconds. The default is 60 seconds.
`-y`	Continues the system shutdown without user intervention. Otherwise, you are prompted to continue the shutdown process.

5. **Select one of the following to turn off power to the system after it is shut down.**

   a. **For SPARC platforms, it is safe to turn off power if the `ok` prompt is displayed.**

   b. **For x86 platforms, it is safe to turn off power if the `type any key to continue` prompt is displayed.**

   Refer to the hardware installation guide that accompanies your system for the location of the power switch.

6. **Turn off power to all external devices.**

   For the location of power switches on any peripheral devices, refer to the hardware installation guides that accompany your peripheral devices.

7. **Install the peripheral device, making sure that the device you are adding has a different target number than the other devices on the system.**

   You often will find a small switch located at the back of the disk for selecting the target number.

   Refer to the hardware installation guide that accompanies the peripheral device for information on installing and connecting the device.

8. **Turn on the power to the system.**

   The system boots to multiuser mode and the login prompt is displayed.

9. **Verify that the peripheral device has been added by attempting to access the device.**

   For information on accessing the device, see Chapter 30.

▼ How to Add a Device Driver

This procedure assumes that the device has already been added to the system. If not, see "What You Need for Unsupported Devices" on page 341.

1. **Become superuser.**
2. **Place the tape, diskette, or CD-ROM into the drive.**
3. **Install the driver.**

```
# pkgadd -d device  package-name
```

`-d` *device*	Identifies the device path name that contains the package.
package-name	Identifies the package name that contains the device driver.

4. **Verify that the package has been added correctly.**

```
# pkgchk package-name
#
```

The system prompt returns with no response if the package is installed correctly.

Example—Adding a Device Driver

The following example shows how to install and verify a package called `XYZdrv`.

```
# pkgadd XYZdrv
(licensing messages displayed)
.
.
.
Installing XYZ Company driver as <XYZdrv>
.
.
.
Installation of <XYZdrv> was successful.
# pkgchk XYZdrv
#
```

CHAPTER 28

Dynamically Configuring Devices (Tasks)

This chapter provides instructions for dynamically configuring devices in the Solaris environment. You can add, remove, or replace devices in the Solaris environment while the system is still running, if the system components support hot-plugging. If the system components do not support hot-plugging, you can reboot the system to reconfigure the devices.

For information on the procedures associated with dynamically configuring devices, see the following:

- "SCSI Hot-Plugging With the `cfgadm` Command (Task Map)" on page 353
- "PCI Hot-Plugging With the `cfgadm` Command (Task Map)" on page 363
- "Application Developer RCM Script (Task Map)" on page 370
- "System Administrator RCM Script (Task Map)" on page 370

For information on hot-plugging USB devices with the `cfgadm` command, see "Hot-Plugging USB Devices With the `cfgadm` Command" on page 399.

For information about accessing devices, see Chapter 30.

Dynamic Reconfiguration and Hot-Plugging

Hot-plugging is the ability to physically add, remove, or replace system components while the system is running. *Dynamic reconfiguration* refers to the ability to hot-plug system components. This term also refers to the general ability to move system resources (both hardware and software) around in the system or to disable them in some way without physically removing them from the system.

You can hot-plug the following devices with the `cfgadm` command:

- USB devices on SPARC and x86 platforms
- SCSI devices on SPARC and x86 platforms
- PCI devices on x86 platforms

Features of the `cfgadm` command include the following:

- Displaying system component status
- Testing system components
- Changing component configurations
- Displaying configuration help messages

The benefit of using the `cfgadm` command to reconfigure systems components is that you can add, remove, or replace components while the system is running. An added benefit is that the `cfgadm` command guides you through the steps needed to add, remove, or replace system components.

For step-by-step instructions on hot-plugging SCSI components, see `cfgadm`(1M) and "SCSI Hot-Plugging With the `cfgadm` Command" on page 354. For step-by-step instructions on hot-plugging PCI adapter cards on x86 based systems, see "x86: PCI Hot-Plugging With the `cfgadm` Command" on page 364.

Note – Not all SCSI and PCI controllers support hot-plugging with the `cfgadm` command.

As part of Sun's high availability strategy, dynamic reconfiguration is expected to be used in conjunction with additional layered products, such as alternate pathing or fail-over software. Both products provide fault tolerance in the event of a device failure.

Without any high availability software, you can replace a failed device by manually stopping the appropriate applications, unmounting noncritical file systems, and then proceeding with the add or remove operations.

Note – For information about hot-plugging devices on your specific hardware configuration, such as enterprise-level systems, please refer to your hardware configuration documentation.

Attachment Points

The `cfgadm` command displays information about *attachment points*, which are locations in the system where dynamic reconfiguration operations can occur.

An attachment point consists of the following:

- An *occupant*, which represents a hardware component that can be configured into the system
- A *receptacle*, which is the location that accepts the occupant

Attachment points are represented by logical and physical attachment point IDs (`Ap_Id`s). The physical `Ap_Id` is the physical pathname of the attachment point. The logical `Ap_Id` is a user-friendly alternative for the physical `Ap_Id`. For more information on `Ap_Id`s, refer to `cfgadm`(1M).

The logical `Ap_Id` for a SCSI Host Bus Adapter (HBA), or SCSI controller, is usually represented by the controller number, such as `c0`.

In cases where no controller number has been assigned to a SCSI HBA, then an internally-generated unique identifier is provided. An example of a unique identifier for a SCSI controller is the following:

```
fas1:scsi
```

The logical `Ap_Id` for a SCSI device usually looks like this:

HBA-logical-apid::device-identifier

In the following example, `c0` is the logical `Ap_Id` for the SCSI HBA:

```
c0::dsk/c0t3d0
```

The device identifier is typically derived from the logical device name for the device in the `/dev` directory. For example, a tape device with logical device name, `/dev/rmt/1`, has the following logical `Ap_Id`:

```
c0::rmt/1
```

If a logical `Ap_Id` of a SCSI device cannot be derived from the logical name in the `/dev` directory, then an internally-generated unique identifier is provided. An example of an identifier for the `/dev/rmt/1` tape device is the following:

```
c0::st4
```

For more information on SCSI `Ap_Id`s, refer to `cfgadm_scsi`(1M).

The `cfgadm` command represents all resources and dynamic reconfiguration operations in terms of a common set of states (such as configured, unconfigured) and operations (connect, configure, unconfigure, and so on). For more information on these common states and operations, see `cfgadm`(1M).

The receptacle and occupant states for the SCSI HBA attachment points are as follows:

Receptacle State	Description	Occupant State	Description
`empty`	N/A for SCSI HBA	`configured`	One or more devices configured on the bus
`disconnected`	Bus quiesced	`unconfigured`	No devices configured
`connected`	Bus active		

Receptacle and occupant states for SCSI device attachment points are as follows:

Receptacle State	Description	Occupant State	Description
`empty`	N/A for SCSI devices	`configured`	Device is configured
`disconnected`	Bus quiesced	`unconfigured`	Device is not configured
`connected`	Bus active		

The state of SCSI attachment points is unknown unless there is special hardware to indicate otherwise. For instructions on displaying SCSI component information, see "How to Display Information About SCSI Devices" on page 354.

x86: Detaching PCI Adapter Cards

A PCI adapter card that is hosting nonvital system resources can be removed if the device driver supports hot-plugging. A PCI adapter card is not detachable if it is a vital system resource. For a PCI adapter card to be detachable the following conditions must be met:

- The device driver must support hot-plugging.
- Critical resources must be accessible through an alternate pathway.

For example, if a system has only one Ethernet card installed in it, the Ethernet card cannot be detached without losing the network connection. This detachment requires additional layered software support to keep the network connection active.

x86: Attaching PCI Adapter Cards

A PCI adapter card can be added to the system as long as the following conditions are met:

- There are slots available.
- The device driver supports hot-plugging for this adapter card.

For step-by-step instructions on adding or removing a PCI adapter card, see "x86: PCI Hot-Plugging With the `cfgadm` Command" on page 364.

SCSI Hot-Plugging With the `cfgadm` Command (Task Map)

Task	Description	For Instructions
1. Display information about SCSI devices	Display information about SCSI controllers and devices.	"How to Display Information About SCSI Devices" on page 354
2. Unconfigure a SCSI controller	Unconfigure a SCSI controller.	"How to Unconfigure a SCSI Controller" on page 355
3. Configure a SCSI controller	Configure a SCSI controller that was previously unconfigured.	"How to Configure a SCSI Controller" on page 355
4. Configure a SCSI device	Configure a specific SCSI device.	"How to Configure a SCSI Device" on page 356
5. Disconnect a SCSI controller	Disconnect a specific SCSI controller.	"How to Disconnect a SCSI Controller" on page 357
6. Connect a SCSI controller	Connect a specific SCSI controller that was previously disconnected.	"How to Connect a SCSI Controller" on page 358
7. Add a SCSI device to a SCSI bus	Add a specific SCSI device to a SCSI bus.	"SPARC: How to Add a SCSI Device to a SCSI Bus" on page 358
8. Replace an identical device on a SCSI controller	Replace a device on the SCSI bus with another device of the same type.	"SPARC: How to Replace an Identical Device on a SCSI Controller" on page 359
9. Remove a SCSI device	Remove a SCSI device from the system.	"SPARC: How to Remove a SCSI Device" on page 360
10. Troubleshooting SCSI configuration problems	Resolve a failed SCSI unconfigure operation.	"How to Resolve a Failed SCSI Unconfigure Operation" on page 363

SCSI Hot-Plugging With the `cfgadm` Command

This section describes various SCSI hot-plugging procedures that you can perform with the `cfgadm` command.

These procedures use specific devices as examples to illustrate how to use the `cfgadm` command to hot-plug SCSI components. The device information that you supply, and that the `cfgadm` command displays, depends on your system configuration.

▼ How to Display Information About SCSI Devices

The following procedure uses SCSI controllers `c0` and `c1` and the devices that are attached to them as examples of the type of device configuration information that you can display with the `cfgadm` command.

Note – If the SCSI device is not supported by the `cfgadm` command, it does not display in the `cfgadm` command output.

---

1. **Become superuser.**
2. **Display information about attachment points on the system.**

```
# cfgadm -l
Ap_Id                Type         Receptacle   Occupant     Condition
c0                   scsi-bus     connected    configured   unknown
c1                   scsi-bus     connected    configured   unknown
```

   In this example, `c0` and `c1` represent two SCSI controllers.

3. **Display information about a system's SCSI controllers and their attached devices.**

```
# cfgadm -al
Ap_Id                Type         Receptacle   Occupant     Condition
c0                   scsi-bus     connected    configured   unknown
c0::dsk/c0t0d0       disk         connected    configured   unknown
c0::rmt/0            tape         connected    configured   unknown
c1                   scsi-bus     connected    configured   unknown
c1::dsk/c1t3d0       disk         connected    configured   unknown
c1::dsk/c1t4d0       unavailable  connected    unconfigured unknown
```

Note – The `cfgadm -l` commands displays information about SCSI HBAs but not SCSI devices. Use the `cfgadm -al` command to display information about SCSI devices such as disk and tapes.

In the following procedures, only SCSI attachment points are listed. The attachment points that are displayed on your system depend on your system configuration.

▼ How to Unconfigure a SCSI Controller

The following procedure uses SCSI controller `c1` as an example of unconfiguring a SCSI controller.

1. **Become superuser.**
2. **Unconfigure a SCSI controller.**

```
# cfgadm -c unconfigure c1
```

3. **Verify that the SCSI controller is unconfigured.**

```
# cfgadm -al
Ap_Id                  Type         Receptacle   Occupant     Condition
c0                     scsi-bus     connected    configured   unknown
c0::dsk/c0t0d0         disk         connected    configured   unknown
c0::rmt/0              tape         connected    configured   unknown
c1                     scsi-bus     connected    unconfigured unknown
```

Notice that the `Occupant` column for `c1` specifies `unconfigured`, indicating that the SCSI bus has no configured occupants.

If the unconfigure operation fails, see "How to Resolve a Failed SCSI Unconfigure Operation" on page 363.

▼ How to Configure a SCSI Controller

The following procedure uses SCSI controller `c1` as an example of configuring a SCSI controller.

1. **Become superuser.**
2. **Configure a SCSI controller.**

```
# cfgadm -c configure c1
```

3. **Verify that the SCSI controller is configured.**

```
# cfgadm -al
Ap_Id                Type         Receptacle   Occupant     Condition
```

```
c0                  scsi-bus     connected    configured   unknown
c0::dsk/c0t0d0      disk         connected    configured   unknown
c0::rmt/0           tape         connected    configured   unknown
c1                  scsi-bus     connected    configured   unknown
c1::dsk/c1t3d0      disk         connected    configured   unknown
c1::dsk/c1t4d0      unavailable  connected    unconfigured unknown
```

The previous unconfigure procedure removed all devices on the SCSI bus. Now all the devices are configured back into the system.

▼ How to Configure a SCSI Device

The following procedure uses SCSI disk `c1t4d0` as an example of configuring a SCSI device.

1. **Become superuser.**
2. **Identify the device to be configured.**

```
# cfgadm -al
Ap_Id               Type          Receptacle   Occupant     Condition
c0                  scsi-bus      connected    configured   unknown
c0::dsk/c0t0d0      disk          connected    configured   unknown
c0::rmt/0           tape          connected    configured   unknown
c1                  scsi-bus      connected    configured   unknown
c1::dsk/c1t3d0      disk          connected    configured   unknown
c1::dsk/c1t4d0      unavailable   connected    unconfigured unknown
```

3. **Configure the SCSI device.**

```
# cfgadm -c configure c1::dsk/c1t4d0
```

4. **Verify that the SCSI device is configured.**

```
# cfgadm -al
Ap_Id               Type          Receptacle   Occupant     Condition
c0                  scsi-bus      connected    configured   unknown
c0::dsk/c0t0d0      disk          connected    configured   unknown
c0::rmt/0           tape          connected    configured   unknown
c1                  scsi-bus      connected    configured   unknown
c1::dsk/c1t3d0      disk          connected    configured   unknown
c1::dsk/c1t4d0      disk          connected    configured   unknown
```

▼ How to Disconnect a SCSI Controller

Caution – Disconnecting a SCSI device must be done with caution, particularly when you are dealing with controllers for disks that contain critical file systems such as root (/), `usr`, `var`, and the `swap` partition. The dynamic reconfiguration software cannot detect all cases where a system hang might result. Use this procedure with caution.

The following procedure uses SCSI controller `c1` as an example of disconnecting a SCSI device.

1. **Become superuser.**

2. **Verify that the device is connected before you disconnect it.**

```
# cfgadm -al
Ap_Id                Type          Receptacle   Occupant     Condition
c0                   scsi-bus      connected    configured   unknown
c0::dsk/c0t0d0       disk          connected    configured   unknown
c0::rmt/0            tape          connected    configured   unknown
c1                   scsi-bus      connected    configured   unknown
c1::dsk/c1t3d0       disk          connected    configured   unknown
c1::dsk/c1t4d0       disk          connected    configured   unknown
```

3. **Disconnect the SCSI controller.**

```
# cfgadm -c disconnect c1
WARNING: Disconnecting critical partitions may cause system hang.
Continue (yes/no)? y
```

Caution – This command suspends all I/O activity on the SCSI bus until the `cfgadm -c connect` command is used. The `cfgadm` command does some basic checking to prevent critical partitions from being disconnected, but it cannot detect all cases. Inappropriate use of this command can result in a system hang and could require a system reboot.

4. **Verify that the SCSI bus is disconnected.**

```
# cfgadm -al
Ap_Id                Type          Receptacle   Occupant     Condition
c0                   scsi-bus      connected    configured   unknown
c0::dsk/c0t0d0       disk          connected    configured   unknown
c0::rmt/0            tape          connected    configured   unknown
c1                   unavailable   disconnected configured   unknown
c1::dsk/c1t10d0      unavailable   disconnected configured   unknown
c1::dsk/c1t4d0       unavailable   disconnected configured   unknown
```

The controller and all the devices that are attached to it are disconnected from the system.

▼ How to Connect a SCSI Controller

The following procedure uses SCSI controller `c1` as an example of connecting a SCSI controller.

1. **Become superuser.**
2. **Verify that the device is disconnected before you connect it.**

```
# cfgadm -al
Ap_Id                Type          Receptacle   Occupant     Condition
c0                   scsi-bus      connected    configured   unknown
c0::dsk/c0t0d0       disk          connected    configured   unknown
c0::rmt/0            tape          connected    configured   unknown
c1                   unavailable   disconnected configured   unknown
c1::dsk/c1t10d0      unavailable   disconnected configured   unknown
c1::dsk/c1t4d0       unavailable   disconnected configured   unknown
```

3. **Connect the SCSI controller.**

```
# cfgadm -c connect c1
```

4. **Verify that the SCSI controller is connected.**

```
# cfgadm -al
Ap_Id                Type          Receptacle   Occupant     Condition
c0                   scsi-bus      connected    configured   unknown
c0::dsk/c0t0d0       disk          connected    configured   unknown
c0::rmt/0            tape          connected    configured   unknown
c1                   scsi-bus      connected    configured   unknown
c1::dsk/c1t3d0       disk          connected    configured   unknown
c1::dsk/c1t4d0       disk          connected    configured   unknown
```

▼ SPARC: How to Add a SCSI Device to a SCSI Bus

SCSI controller `c1` provides an example of how to add a SCSI device to a SCSI bus.

Note – When you add devices, you specify the `Ap_Id` of the SCSI HBA (controller) to which the device is attached, not the `Ap_Id` of the device itself.

---

1. **Become superuser.**
2. **Identify the current SCSI configuration.**

```
# cfgadm -al
Ap_Id                Type          Receptacle   Occupant     Condition
c0                   scsi-bus      connected    configured   unknown
c0::dsk/c0t0d0       disk          connected    configured   unknown
c0::rmt/0            tape          connected    configured   unknown
```

```
c1                  scsi-bus     connected    configured   unknown
c1::dsk/c1t3d0      disk         connected    configured   unknown
```

3. **Add the SCSI device to the SCSI bus.**

```
# cfgadm -x insert_device c1
Adding device to SCSI HBA: /devices/sbus@1f,0/SUNW,fas@1,8800000
This operation will suspend activity on SCSI bus: c1
```

   a. **Type `y` at the `Continue (yes/no)?` prompt to proceed.**

```
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

   I/O activity on the SCSI bus is suspended while the hot-plug operation is in progress.

   b. **Connect the device and then power it on.**

   c. **Type `y` at the `Enter y if operation is complete or n to abort (yes/no)?` prompt.**

```
Enter y if operation is complete or n to abort (yes/no)? y
```

4. **Verify that the device has been added.**

```
# cfgadm -al
Ap_Id               Type         Receptacle   Occupant     Condition
c0                  scsi-bus     connected    configured   unknown
c0::dsk/c0t0d0      disk         connected    configured   unknown
c0::rmt/0           tape         connected    configured   unknown
c1                  scsi-bus     connected    configured   unknown
c1::dsk/c1t3d0      disk         connected    configured   unknown
c1::dsk/c1t4d0      disk         connected    configured   unknown
```

A new disk has been added to controller `c1`.

▼ SPARC: How to Replace an Identical Device on a SCSI Controller

The following procedure uses SCSI disk `c1t4d0` as an example of replacing an identical device on a SCSI controller.

1. **Become superuser.**

2. **Identify the current SCSI configuration.**

```
# cfgadm -al
Ap_Id               Type         Receptacle   Occupant     Condition
c0                  scsi-bus     connected    configured   unknown
c0::dsk/c0t0d0      disk         connected    configured   unknown
c0::rmt/0           tape         connected    configured   unknown
```

```
c1                  scsi-bus     connected    configured   unknown
c1::dsk/c1t3d0      disk         connected    configured   unknown
c1::dsk/c1t4d0      disk         connected    configured   unknown
```

3. **Replace a device on the SCSI bus with another device of the same type.**

```
# cfgadm -x replace_device c1::dsk/c1t4d0
Replacing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
This operation will suspend activity on SCSI bus: c1
```

   a. **Type `y` at the `Continue (yes/no)?` prompt to proceed.**

      I/O activity on the SCSI bus is suspended while the hot-plug operation is in progress.

```
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

   b. **Power off the device to be removed and remove it.**

   c. **Add the replacement device. Then, power it on.**

      The replacement device should be of the same type and at the same address (target and lun) as the device to be removed

   d. **Type `y` at the `Enter y if operation is complete or n to abort (yes/no)?` prompt.**

```
Enter y if operation is complete or n to abort (yes/no)? y
```

4. **Verify that the device has been replaced.**

```
# cfgadm -al
Ap_Id               Type         Receptacle   Occupant     Condition
c0                  scsi-bus     connected    configured   unknown
c0::dsk/c0t0d0      disk         connected    configured   unknown
c0::rmt/0           tape         connected    configured   unknown
c1                  scsi-bus     connected    configured   unknown
c1::dsk/c1t3d0      disk         connected    configured   unknown
c1::dsk/c1t4d0      disk         connected    configured   unknown
```

▼ SPARC: How to Remove a SCSI Device

The following procedure uses SCSI disk `c1t4d0` as an example of removing a device on a SCSI controller.

1. **Become superuser.**

2. **Identify the current SCSI configuration.**

```
# cfgadm -al
Ap_Id               Type         Receptacle   Occupant     Condition
c0                  scsi-bus     connected    configured   unknown
```

```
c0::dsk/c0t0d0       disk          connected    configured   unknown
c0::rmt/0            tape          connected    configured   unknown
c1                   scsi-bus      connected    configured   unknown
c1::dsk/c1t3d0       disk          connected    configured   unknown
c1::dsk/c1t4d0       disk          connected    configured   unknown
```

3. **Remove the SCSI device from the system.**

```
# cfgadm -x remove_device c1::dsk/c1t4d0
Removing SCSI device: /devices/sbus@1f,0/SUNW,fas@1,8800000/sd@4,0
This operation will suspend activity on SCSI bus: c1
```

   a. **Type `y` at the `Continue (yes/no)?` prompt to proceed.**

```
Continue (yes/no)? y
SCSI bus quiesced successfully.
It is now safe to proceed with hotplug operation.
```

      I/O activity on the SCSI bus is suspended while the hot-plug operation is in progress.

   b. **Power off the device to be removed and remove it.**

   c. **Type `y` at the `Enter y if operation is complete or n to abort (yes/no)?` prompt.**

```
Enter y if operation is complete or n to abort (yes/no)? y
```

4. **Verify that the device has been removed from the system.**

```
# cfgadm -al
Ap_Id                Type          Receptacle   Occupant     Condition
c0                   scsi-bus      connected    configured   unknown
c0::dsk/c0t0d0       disk          connected    configured   unknown
c0::rmt/0            tape          connected    configured   unknown
c1                   scsi-bus      connected    configured   unknown
c1::dsk/c1t3d0       disk          connected    configured   unknown
```

SPARC: Troubleshooting SCSI Configuration Problems

This section provides error messages and possible solutions for troubleshooting SCSI configuration problems. For more information on troubleshooting SCSI configuration problems, see `cfgadm`(1M).

Error Message

```
cfgadm: Component system is busy, try again: failed to offline:
     device path
          Resource              Information
     ------------------  -------------------------
     /dev/dsk/c1t0d0s0   mounted filesystem "/file-system"
```

Cause
You attempted to remove or replace a device with a mounted file system.

Solution
Unmount the file system that is listed in the error message and retry the `cfgadm` operation.

If you use the `cfgadm` command to remove a system resource, such as a swap device or a dedicated dump device, an error messages similar to the following is displayed if the system resource is still active.

Error Message

```
cfgadm: Component system is busy, try again: failed to offline:
     device path
          Resource              Information
     ------------------  -------------------------
     /dev/dsk/device-name   swap area
```

Cause
You attempted to remove or replace one or more configured swap areas.

Solution
Unconfigure the swap areas on the device that is specified and retry the `cfgadm` operation.

Error Message

```
cfgadm: Component system is busy, try again: failed to offline:
     device path
          Resource              Information
     ------------------  -------------------------
     /dev/dsk/device-name   dump device (swap)
```

Cause
You attempted to remove or replace a dump device that is configured on a swap area.

Solution
Unconfigure the dump device that is configured on the swap area and retry the `cfgadm` operation.

Error Message

```
cfgadm: Component system is busy, try again: failed to offline:
     device path
          Resource              Information
     ------------------  -------------------------
     /dev/dsk/device-name   dump device (dedicated)
```

Cause
You attempted to remove or replace a dedicated dump device.

Solution
Unconfigure the dedicate dump device and retry the `cfgadm` operation.

▼ How to Resolve a Failed SCSI Unconfigure Operation

Use this procedure if one or more target devices are busy and the SCSI unconfigure operation fails. Otherwise, future dynamic reconfiguration operations on this controller and target devices will fail with a `dr in progress` message.

1. **Become superuser, if not done already.**
2. **Type the following command to reconfigure the controller.**

   # **`cfgadm -c configure`** *device-name*

PCI Hot-Plugging With the `cfgadm` Command (Task Map)

Task	Description	For Instructions
1. Display PCI Slot Configuration Information	Display the status of PCI hot-pluggable devices and slots on the system.	"x86: How to Display PCI Slot Configuration Information" on page 364
2. Remove a PCI adapter card	Unconfigure the card, disconnect power from the slot, and remove the card from the system.	"x86: How to Remove a PCI Adapter Card" on page 365
3. Add a PCI adapter card	Insert the adapter card into a hot-pluggable slot, connect power to the slot, and configure the card.	"x86: How to Add a PCI Adapter Card" on page 365
4. Troubleshooting PCI configuration problems	Identify error message and possible solutions to resolve PCI configuration problems.	"x86: Troubleshooting PCI Configuration Problems" on page 366

x86: PCI Hot-Plugging With the `cfgadm` Command

This section provides step-by-step instructions for hot-plugging PCI adapter cards on x86 based systems.

In the examples, only PCI attachment points are listed, for brevity. The attachment points that are displayed on your system depend on your system configuration.

▼ x86: How to Display PCI Slot Configuration Information

The `cfgadmin` command displays the status of PCI hot-pluggable devices and slots on a system. For more information, see `cfgadm`(1M).

1. **Become superuser.**
2. **Display PCI slot configuration information.**

```
# cfgadm
Ap_Id                Type         Receptacle   Occupant     Condition
pci1:hpc0_slot0      unknown      empty        unconfigured unknown
pci1:hpc0_slot1      unknown      empty        unconfigured unknown
pci1:hpc0_slot2      unknown      empty        unconfigured unknown
pci1:hpc0_slot3      ethernet/hp  connected    configured   ok
pci1:hpc0_slot4      unknown      empty        unconfigured unknown
```

Display specific PCI device information.

```
# cfgadm -s "cols=ap_id:type:info" pci
Ap_Id                Type         Information
pci1:hpc0_slot0      unknown      Slot 7
pci1:hpc0_slot1      unknown      Slot 8
pci1:hpc0_slot2      unknown      Slot 9
pci1:hpc0_slot3      ethernet/hp  Slot 10
pci1:hpc0_slot4      unknown      Slot 11
```

The logical `Ap_Id`, `pci1:hpc0_slot0`, is the logical `Ap_Id` for hot-pluggable slot, `Slot 7`. The component `hpc0` indicates the hot-pluggable adapter card for this slot, and `pci1` indicates the PCI bus instance. The `Type` field indicates the type of PCI adapter card that is present in the slot.

▼ x86: How to Remove a PCI Adapter Card

1. **Become superuser.**
2. **Determine which slot the PCI adapter card is in.**

```
# cfgadm
Ap_Id                Type         Receptacle   Occupant     Condition
pci1:hpc0_slot0      unknown      empty        unconfigured unknown
pci1:hpc0_slot1      unknown      empty        unconfigured unknown
pci1:hpc0_slot2      unknown      empty        unconfigured unknown
pci1:hpc0_slot3      ethernet/hp  connected    configured   ok
pci1:hpc0_slot4      unknown      empty        unconfigured unknown
```

3. **Stop the application that has the device open.**

   For example, if the device is an Ethernet card, use the `ifconfig` command to bring down the interface and unplumb the interface.

4. **Unconfigure the device.**

```
# cfgadm -c unconfigure pci1:hpc0_slot3
```

5. **Confirm that the device has been unconfigured.**

```
# cfgadm
Ap_Id                Type         Receptacle   Occupant     Condition
pci1:hpc0_slot0      unknown      empty        unconfigured unknown
pci1:hpc0_slot1      unknown      empty        unconfigured unknown
pci1:hpc0_slot2      unknown      empty        unconfigured unknown
pci1:hpc0_slot3      ethernet/hp  connected    unconfigured unknown
pci1:hpc0_slot4      unknown      empty        unconfigured unknown
```

6. **Disconnect the power to the slot.**

```
# cfgadm -c disconnect pci1:hpc0_slot3
```

7. **Confirm that the device has been disconnected.**

```
# cfgadm
Ap_Id                Type         Receptacle   Occupant     Condition
pci1:hpc0_slot0      unknown      empty        unconfigured unknown
pci1:hpc0_slot1      unknown      empty        unconfigured unknown
pci1:hpc0_slot2      unknown      empty        unconfigured unknown
pci1:hpc0_slot3      ethernet/hp  disconnected unconfigured unknown
pci1:hpc0_slot4      unknown      empty        unconfigured unknown
```

8. **Open the slot latches and remove the PCI adapter card.**

▼ x86: How to Add a PCI Adapter Card

1. **Become superuser.**
2. **Identify the hot-pluggable slot and open latches.**

3. **Insert the PCI adapter card into a hot-pluggable slot.**

4. **Determine which slot the PCI adapter card is in once it is inserted. Close the latches.**

```
# cfgadm
Ap_Id                Type         Receptacle   Occupant     Condition
pci1:hpc0_slot0      unknown      empty        unconfigured unknown
pci1:hpc0_slot1      unknown      empty        unconfigured unknown
pci1:hpc0_slot2      unknown      empty        unconfigured unknown
pci1:hpc0_slot3      ethernet/hp  disconnected unconfigured unknown
pci1:hpc0_slot4      unknown      empty        unconfigured unknown
```

5. **Connect the power to the slot.**

```
# cfgadm -c connect pci1:hpc0_slot3
```

6. **Confirm that the slot is connected.**

```
# cfgadm
Ap_Id                Type         Receptacle   Occupant     Condition
pci1:hpc0_slot0      unknown      empty        unconfigured unknown
pci1:hpc0_slot1      unknown      empty        unconfigured unknown
pci1:hpc0_slot2      unknown      empty        unconfigured unknown
pci1:hpc0_slot3      ethernet/hp  connected    unconfigured unknown
pci1:hpc0_slot4      unknown      empty        unconfigured unknown
```

7. **Configure the PCI adapter card.**

```
# cfgadm -c configure pci1:hpc0_slot3
```

8. **Verify the configuration of the PCI adapter card in the slot.**

```
# cfgadm
Ap_Id                Type         Receptacle   Occupant     Condition
pci1:hpc0_slot0      unknown      empty        unconfigured unknown
pci1:hpc0_slot1      unknown      empty        unconfigured unknown
pci1:hpc0_slot2      unknown      empty        unconfigured unknown
pci1:hpc0_slot3      ethernet/hp  connected    configured   unknown
pci1:hpc0_slot4      unknown      empty        unconfigured unknown
```

9. **Configure any supporting software if this device is a new device.**

   For example, if this device is an Ethernet card, use the `ifconfig` command to set up the interface.

x86: Troubleshooting PCI Configuration Problems

Error Message

```
cfgadm: Configuration operation invalid: invalid transition
```

Cause

An invalid transition was attempted.

Solution
Check whether the `cfgadm -c` command was issued appropriately. Use the `cfgadm` command to check the current receptacle and occupant state and to make sure that the `Ap_Id` is correct.

Error Message

```
cfgadm: Attachment point not found
```

Cause
The specified attachment point was not found.

Solution
Check whether the attachment point is correct. Use the `cfgadm` command to display a list of available attachment points. Also check the physical path to see if the attachment point is still there.

Note – In addition to the `cfgadm` command, several other commands are helpful during hot-pluggable operations. The `prtconf` command displays whether Solaris recognizes the hardware. After adding hardware, use the `prtconf` command to verify that the hardware is recognized. After a configure operation, use the `prtconf -D` command to verify that the driver is attached to the newly installed hardware device.

Reconfiguration Coordination Manager (RCM) Script Overview

The Reconfiguration Coordination Manager (RCM) is the framework that manages the dynamic removal of system components. By using RCM, you can register and release system resources in an orderly manner.

You can use the new RCM script feature to write your own scripts to shut down your applications, or to cleanly release the devices from your applications during dynamic reconfiguration. The RCM framework launches a script automatically in response to a reconfiguration request, if the request impacts the resources that are registered by the script.

You can also release resources from applications manually before you could dynamically remove the resource. Or, you could use the `cfgadm` command with the `-f` option to force a reconfiguration operation, but this option might leave your applications in an unknown state. Also, the manual release of resources from applications commonly causes errors.

The RCM script feature simplifies and better controls the dynamic reconfiguration process. By creating an RCM script, you can do the following:

- Automatically release a device when you dynamically remove a device. This process also closes the device if the device is opened by an application.
- Run site-specific tasks when you dynamically remove a device from the system.

What Is an RCM Script?

An RCM script is as follows:

- An executable shell script (Perl, `sh`, `csh`, or `ksh`) or binary program that the RCM daemon runs. Perl is the recommended language.
- A script that runs in its own address space by using the user ID of the script file owner.
- A script that is run by the RCM daemon when you use the `cfgadm` command to dynamically reconfigure a system resource.

What Can an RCM Script Do?

You can use an RCM script to release a device from an application when you dynamically remove a device. If the device is currently open, the RCM script also closes the device.

For example, an RCM script for a tape backup application can inform the tape backup application to close the tape drive or shut down the tape backup application.

How Does the RCM Script Process Work?

You can invoke a script as follows:

`$` *`script-name command`* `[`*`args`* `...]`

A script performs the following basic steps:

1. Takes the RCM command from command-line arguments.
2. Executes the command.
3. Writes the results to `stdout` as name-value pairs.
4. Exits with the appropriate exit status.

The RCM daemon runs one instance of a script at a time. For example, if a script is running, the RCM daemon does not run the same script until the first script exits.

RCM Script Commands

You must include the following RCM commands in an RCM script:

- `scriptinfo` - Gathers script information
- `register` - Registers interest in resources
- `resourceinfo` - Gathers resource information

You might include some or all of the following RCM commands:

- `queryremove` - Queries whether the resource can be released
- `preremove` - Releases the resource
- `postremove` - Provides post-resource removal notification
- `undoremove` - Undoes the actions done in `preremove`

For a complete description of these RCM commands, see `rcmscript`(4).

RCM Script Processing Environment

When you dynamically remove a device, the RCM daemon runs the following:

- The script's `register` command to gather the list of resources (device names) that are identified in the script.
- The script's `queryremove/preremove` commands prior to removing the resource if the script's registered resources are affected by the dynamic remove operation.
- The script's `postremove` command if the remove operation succeeds. However, if the remove operation fails, the RCM daemon runs the script's `undoremove` command.

RCM Script Tasks

The following sections describe the RCM script tasks for application developers and system administrators.

Application Developer RCM Script (Task Map)

The following task map describes the tasks for an application developer who is creating an RCM script.

Task	Description	For Instructions
1. Identify resources your application uses	Identify the resources (device names) your application uses that you could potentially dynamically remove.	`cfgadm`(1M)
2. Identify commands to release the resource	Identify the commands for notifying the application to cleanly release the resource from the application.	Application documentation
3. Identify commands for post-removal of the resource	Include the commands for notifying the application of the resource removal.	`rcmscript`(4)
4. Identify commands if the resource removal fails	Include the commands for notifying the application of the available resource.	`rcmscript`(4)
5. Write the RCM script	Write the RCM script based on the information identified in the previous tasks.	"Tape Backup RCM Script Example" on page 373
6. Install the RCM script	Add the script to the appropriate script directory.	"How to Install an RCM Script" on page 372
7. Test the RCM script	Test the script by running the script commands manually and by initiating a dynamic reconfiguration operation.	"How to Test an RCM Script" on page 372

System Administrator RCM Script (Task Map)

The following task map describes the tasks for a system administrator who is creating an RCM script to do site customization.

Task	Description	For Instructions
1. Identify resources to be dynamically removed	Identify the resources (device names) to be potentially removed by using the `cfgadm -l` command.	`cfgadm`(1M)

Task	Description	For Instructions
2. Identify applications to be stopped	Identify the commands for stopping the applications cleanly.	Application documentation
3. Identify commands for pre-removal and post-removal of the resource	Identify the actions to be taken before and after the resource is removed.	`rcmscript`(4)
4. Write the RCM script	Write the RCM script based on the information identified in the previous tasks.	"Tape Backup RCM Script Example" on page 373
5. Install the RCM script	Add the script to the appropriate script directory.	"How to Install an RCM Script" on page 372
6. Test the RCM script	Test the script by running the script commands manually and by initiating a dynamic reconfiguration operation.	"How to Test an RCM Script" on page 372

Naming an RCM Script

A script must be named as *vendor,service* where the following applies:

vendor	Is the stock symbol of the vendor that provides the script, or any distinct name that identifies the vendor.
service	Is the name of the service that the script represents.

Installing or Removing an RCM Script

You must be superuser (root) to install or remove an RCM script. Use this table to determine where you should install your RCM script.

TABLE 28–1 RCM Script Directories

Directory Location	Script Type
`/etc/rcm/scripts`	Scripts for specific systems
`/usr/platform/‘uname -i‘/lib/rcm/scripts`	Scripts for a specific hardware implementation
`/usr/platform/‘uname -m‘/lib/rcm/scripts`	Scripts for a specific hardware class

TABLE 28–1 RCM Script Directories *(Continued)*

Directory Location	Script Type
`/usr/lib/rcm/scripts`	Scripts for any hardware

▼ How to Install an RCM Script

1. **Become superuser.**
2. **Copy the script to the appropriate directory as described in Table 28–1.**
   For example:

   ```
   # cp SUNW,sample.pl /usr/lib/rcm/scripts
   ```

3. **Change the user ID and the group ID of the script to the desired values.**

   ```
   # chown user:group /usr/lib/rcm/scripts/SUNW,sample.pl
   ```

4. **Send `SIGHUP` to the RCM daemon.**

   ```
   # pkill -HUP -x -u root rcm_daemon
   ```

▼ How to Remove an RCM Script

1. **Become superuser.**
2. **Remove the script from the RCM script directory.**
   For example:

   ```
   # rm /usr/lib/rcm/scripts/SUNW,sample.pl
   ```

3. **Send `SIGHUP` to the RCM daemon.**

   ```
   # pkill -HUP -x -u root rcm_daemon
   ```

▼ How to Test an RCM Script

1. **Set environment variables, such as `RCM_ENV_FORCE`, in the command-line shell before running your script.**
   For example, in the Korn shell, use the following:

   ```
   $ export RCM_ENV_FORCE=TRUE
   ```

2. **Test the script by running the script commands manually from the command line.**
   For example:

   ```
   $ script-name scriptinfo
   $ script-name register
   $ script-name preremove resource-name
   ```

```
$ script-name postremove resource-name
```

3. **Make sure each RCM script command in your script prints appropriate output to `stdout`.**
4. **Install the script in the appropriate script directory.**
   For more information, see "How to Install an RCM Script" on page 372.
5. **Test the script by initiating a dynamic remove operation:**
   For example, assume your script registers the device, `/dev/dsk/c1t0d0s0`. Try these commands.

```
$ cfgadm -c unconfigure c1::dsk/c1t0d0
$ cfgadm -f -c unconfigure c1::dsk/c1t0d0
$ cfgadm -c configure c1::dsk/c1t0d0
```

Caution – Make sure that you are familiar with these commands because they can alter the state of the system and can cause system failures.

Tape Backup RCM Script Example

This example illustrates how to use an RCM script for tape backups.

What the Tape Backup RCM Script Does

The tape backup RCM script performs the following steps:

1. Sets up a dispatch table of RCM commands.
2. Calls the dispatch routine that corresponds to the specified RCM command and exits with status 2 for unimplemented RCM commands.
3. Sets up the `scriptinfo` section:

   ```
   rcm_script_func_info=Tape backup appl script for DR
   ```
4. Registers all tape drives in the system by printing all tape drive device names to `stdout`.

   ```
   rcm_resource_name=/dev/rmt/$f
   ```
   If an error occurs, prints the error information to `stdout`.

   ```
   rcm_failure_reason=$errmsg
   ```
5. Sets up the resource information for the tape device.

   ```
   rcm_resource_usage_info=Backup Tape Unit Number $unit
   ```

6. Sets up the `preremove` information by checking if the backup application is using the device. If the backup application is not using the device, the dynamic reconfiguration operation continues. If the backup application is using the device, the script checks `RCM_ENV_FORCE`. If `RCM_ENV_FORCE` is set to `FALSE`, the script denies the dynamic reconfiguration operation and prints the following message:

   ```
   rcm_failure_reason=tape backup in progress pid=...
   ```

   If `RCM_ENV_FORCE` is set to `TRUE`, the backup application is stopped, and the reconfiguration operation proceeds.

Outcomes of the Tape Backup Reconfiguration Scenarios

Here are the various outcomes if you use the `cfgadm` command to remove a tape device without the RCM script.

- If you use the `cfgadm` command and the backup application is not using the tape device, the operation succeeds.
- If you use the `cfgadm` command and the backup application is using the tape device, the operation fails.

Here are the various outcomes if you use the `cfgadm` command to remove a tape device with the RCM script.

- If you use the `cfgadm` command and the backup application is not using the tape device, the operation succeeds.
- If you use the `cfgadm` command without the `-f` option and the backup application is using the tape device, the operation fails with an error message similar to the following:

  ```
  tape backup in progress pid=...
  ```

- If you use the `cfgadm -f` command and the backup application is using the tape device, the script stops the backup application and the `cfgadm` operation succeeds.

Example—Tape Backup RCM Script

```
#! /usr/bin/perl -w
   #
   # A sample site customization RCM script.
   #
   # When RCM_ENV_FORCE is FALSE this script indicates to RCM that it cannot
   # release the tape drive when the tape drive is being used for backup.
   #
   # When RCM_ENV_FORCE is TRUE this script allows DR removing a tape drive
   # when the tape drive is being used for backup by killing the tape
   # backup application.
   #
```

```
    use strict;

    my ($cmd, %dispatch);
    $cmd = shift(@ARGV);
# dispatch table for RCM commands
    %dispatch = (
            "scriptinfo"    =>      \&do_scriptinfo,
            "register"      =>      \&do_register,
            "resourceinfo"  =>      \&do_resourceinfo,
            "queryremove"   =>      \&do_preremove,
            "preremove"     =>      \&do_preremove
    );


    if (defined($dispatch{$cmd})) {
            &{$dispatch{$cmd}};
    } else {
            exit (2);
    }

    sub do_scriptinfo
    {
            print "rcm_script_version=1\n";
            print "rcm_script_func_info=Tape backup appl script for DR\n";
            exit (0);
    }

    sub do_register
{
            my ($dir, $f, $errmsg);

            $dir = opendir(RMT, "/dev/rmt");
            if (!$dir) {
                 $errmsg = "Unable to open /dev/rmt directory: $!";
                 print "rcm_failure_reason=$errmsg\n";
                 exit (1);
            }

            while ($f = readdir(RMT)) {
                # ignore hidden files and multiple names for the same device
                if (($f !~ /^\./) && ($f =~ /^[0-9]+$/)) {
                        print "rcm_resource_name=/dev/rmt/$f\n";
                    }

            }

            closedir(RMT);
            exit (0);
    }
sub do_resourceinfo
    {
      my ($rsrc, $unit);

      $rsrc = shift(@ARGV);
      if ($rsrc =~ /^\/dev\/rmt\/([0-9]+)$/) {
```

```
            $unit = $1;
            print "rcm_resource_usage_info=Backup Tape Unit Number $unit\n";
            exit (0);
        } else {
            print "rcm_failure_reason=Unknown tape device!\n";
             exit (1);
         }
    }

    sub do_preremove
    {
            my ($rsrc);

            $rsrc = shift(@ARGV);

            # check if backup application is using this resource
            #if (the backup application is not running on $rsrc) {
                    # allow the DR to continue
            #        exit (0);
            #}
            #
            # If RCM_ENV_FORCE is FALSE deny the operation.
            # If RCM_ENV_FORCE is TRUE kill the backup application in order
            # to allow the DR operation to proceed
            #
            if ($ENV{RCM_ENV_FORCE} eq 'TRUE') {
                 if ($cmd eq 'preremove') {
                         # kill the tape backup application
                 }
                 exit (0);
            } else {
               #
               # indicate that the tape drive can not be released
               # since the device is being used for backup by the
               # tape backup application
               #
               print "rcm_failure_reason=tape backup in progress pid=...\n"
;
               exit (3);
            }
    }
```

CHAPTER 29

Using USB Devices (Overview/Tasks)

This chapter provides an overview of Universal Serial Bus (USB) devices and step-by-step instructions for using USB devices in the Solaris environment.

For information on the procedures associated with using USB devices, see the following:

- "Using USB Mass Storage Devices (Task Map)" on page 384
- "Using USB Audio Devices (Task Map)" on page 392
- "Hot-Plugging USB Devices With the `cfgadm` Command (Task Map)" on page 398

This is a list of the overview information in this chapter.

- "Overview of USB Devices" on page 377
- "About USB in the Solaris Environment" on page 381
- "Using USB Mass Storage Devices" on page 385
- "Hot-Plugging USB Devices" on page 386
- "Using USB Audio Devices" on page 392
- "Troubleshooting USB Audio Device Problems" on page 397
- "Hot-Plugging USB Devices With the `cfgadm` Command" on page 399

For general information about dynamic reconfiguration and hot-plugging, see Chapter 28.

For information on configuring USB printers, see "What's New in Printing?" in *System Administration Guide: Advanced Administration*.

Overview of USB Devices

Universal Serial Bus (USB) was developed by the PC industry to provide a low-cost solution for attaching peripheral devices, such as keyboards, mouse devices, and printers, to a system.

USB connectors are designed to fit only one type of cable, one way. Devices can connect to hub devices, which connect several devices, including other hub devices. The primary design motivation for USB was to alleviate the need for multiple connector types for different devices. This design reduces the clutter on the back panel of a system. Additional advantages of using USB devices are as follows:

- USB devices are hot-pluggable. For more information, see "Hot-Plugging USB Devices" on page 386.
- Supports a maximum of 126 devices per host controller in the Solaris environment.
- Supports a maximum of 12 Mbit/sec data transfer.
- Supports low speed (1.5 Mbit/sec) and full speed (12 Mbit/sec) devices.
- Supports Solaris Ready USB PCI controllers. For more information, see `http://www.sun.com/io`.
- The bus can be easily extended by adding low-cost external hubs. Hubs can be connected to hubs to form a tree topology.

Sun Microsystems support for USB devices includes the following:

- USB 1.1 devices are supported in the Solaris 9 environment.
- Sun Blade™ 100 and Sun Blade 1000 systems that run the Solaris 8 10/00, 1/01, 4/01, 7/01, 10/01, 2/02 and the Solaris 9 releases provide USB device support.
- Netra™ X1/T1 and Sunfire 280R systems that run the Solaris 9 release.
- x86 based systems that run the Solaris 8 and Solaris 9 x86 Platform Editions provide USB support also. For more information, see `scsa2usb`(7D).

This table lists specific USB devices that are supported in the Solaris environment.

USB Devices	Systems Supported
HID control on audio devices	SPARC based and x86 based systems.
Hubs	SPARC based and x86 based systems.
Keyboards and mouse devices	SPARC based systems with Sun USB support based on the `ohci` controller.
	x86 based systems with a USB bus based on the `uhci` controller.
	Only on-board USB controllers are supported.
Mass storage devices	SPARC based and x86 based systems.
Printers	SPARC based and x86 based systems.
Speakers and microphones	SPARC based and x86 based systems.

Commonly Used USB Acronyms

The following table describes the USB acronyms that are used in the Solaris environment. For a complete description of USB components and acronyms, go to `http://www.usb.org`.

Acronym	Definition
USB	Universal Serial Bus
USBA	Universal Serial Bus Architecture (Solaris)
USBAI	USBA Client Driver Interface (Solaris)
HCD	USB host controller driver

USB Bus Description

The USB specification is openly available and free of royalties. The specification defines the electrical and mechanical interfaces of the bus and the connectors.

USB employs a topology in which hubs provide attachment points for USB devices. The host controller contains the root hub, which is the origin of all USB ports in the system. For more information about hubs, see "USB Host Controller and Root Hub" on page 383.

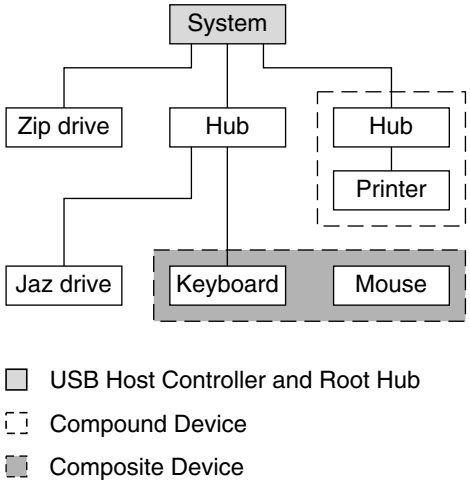


FIGURE 29–1 USB Physical Device Hierarchy

Figure 29–1 shows a system with three active USB ports. The first USB port has a Zip drive that does not have an embedded hub, so you cannot attach additional devices. The second USB port has a hub with a Jaz drive and a composite keyboard and mouse device connected. One port from the secondary hub has a keyboard with an embedded hub where the mouse is attached.

Figure 29–1 also shows an example of a hub and printer as a *compound device*. Both the hub and the printer are enclosed in the same plastic case, but the hub and the printer have separate USB bus addresses. The same diagram shows an example of a *composite device*. The composite keyboard and mouse device are also enclosed in the same plastic case, but they have the same USB bus address. A cable connects the USB mouse to the composite keyboard/controller in this figure.

The device tree path name for some of the devices that are displayed in Figure 29–1 are listed in this table.

Zip drive	`/pci@1f,4000/usb@5/storage@1`
Keyboard	`/pci@1f,4000/usb@5/hub@2/device@1/keyboard@0`
Mouse	`/pci@1f,4000/usb@5/hub@2/device@1/mouse@1`
Jaz drive	`/pci@1f,4000/usb@5/hub@2/storage@3`
Printer	`/pci@1f,4000/usb@5/hub@3/printer@1`

USB Devices and Drivers

USB devices are divided into device classes. Each device class has a corresponding driver. Devices within a class are managed by the same device driver. However, the USB specification also allows for vendor-specific devices that are not part of a specific class. Devices with similar attributes and services are grouped.

The Human Interface Device (HID) class contains devices that are user-controlled such as keyboards, mouse devices, and joysticks. The Communication Device class contains devices that connect to a telephone, such as modems or an ISDN interface. Other device classes include the Audio, Monitor, Printer, and Storage Device classes. Each USB device contains descriptors that reflect the class of the device. A device class specifies how its members should behave in configuration and data transfer. You can obtain additional class information from the `http://www.usb.org` site.

Solaris USB Architecture (USBA)

USB devices are represented as two levels of device tree nodes. A device node represents the entire USB *device*. One or more child *interface* nodes represent the individual USB interfaces on the device. For special cases, the device nodes and interface nodes are *combined* into a single combined node.

Driver binding is achieved by using the compatible name properties. For more information, refer to 3.2.2.1 of the IEEE 1275 USB binding and *Writing Device Drivers*. A driver can either bind to the entire device and control all the interfaces. Or, a driver can bind to just one interface. For example, a keyboard or mouse. If no vendor or class driver claims the entire device, a generic USB multi-interface driver is bound to the device-level node. This driver attempts to bind drivers to each interface by using compatible names properties, as defined in section 3.3.2.1 of the 1275 binding.

The Solaris USB Architecture (USBA) adheres to the USB 1.0 and 1.1 specification plus Solaris driver requirements. The USBA model is similar to Sun Common SCSI Architecture (SCSA). The USBA is a thin layer that provides a generic USB transport-layer abstraction to the client driver.

The differences between SCSA and USBA are that the SCSA relies on `.conf` files to probe the bus, while USB hub drivers are self-probing nexus drivers.

About USB in the Solaris Environment

This section describes information you should know about USB in the Solaris environment.

USB Keyboards and Mouse Devices

Only Sun USB keyboards and mouse devices are officially supported. Additionally, keep only one USB keyboard and mouse on the system at all times because multiple USB keyboards and mouse devices are not supported in the Solaris environment. See the following items for details.

- A keyboard and mouse that are connected anywhere on the bus are configured as a console keyboard and mouse. Booting the system is slower if the keyboard and mouse are not on the root hub.
- You can move a console keyboard and mouse to another hub at any time *after* a system reboot. You cannot move a console keyboard and mouse *during* a reboot or at the `ok` prompt. After you plug in a keyboard and mouse, they are fully functional again.
- **SPARC** – The power key on a USB keyboard behaves differently than the power key on the Sun type 5 keyboard. On a USB keyboard, you can suspend or shut down the system by using the SUSPEND/SHUTDOWN key, but you cannot power up the system.
- The keys just to the left of the keypad do not function on third-party USB keyboards.
- Multiple keyboards are not supported:
  - Multiple keyboards enumerate and are usable, but they are not plumbed as console keyboards.
  - The first keyboard that is probed at boot time becomes the console keyboard. The result of this probing might cause confusion if multiple keyboards are plugged in at boot time.
  - If you unplug the console keyboard, the next available USB keyboard does not become the console keyboard. The next hot-plugged keyboard becomes the console keyboard.
- Multiple mouse devices are not supported:
  - Multiple mouse devices enumerate and are usable, but they are not plumbed as console mouse devices.
  - The first mouse that is probed at boot time becomes the console mouse. The result of this probing might cause confusion if you have multiple mouse devices plugged in at boot time.
  - If you unplug the console mouse, the next available USB mouse does not become the console mouse. The next hot-plugged mouse becomes the console mouse.
- If you have a third-party composite keyboard with a PS/2 mouse, and the composite keyboard/mouse is the first one to be probed, it becomes the console keyboard/mouse even if the PS/2 mouse is not plugged in. Thus, another USB mouse plugged into the system cannot work because it is not configured as the console mouse.

- Only two-button and three-button mouse devices are supported. A wheel-on-wheel mouse acts like a plain-button mouse. A mouse with more than three buttons functions like a three–button mouse.

USB Host Controller and Root Hub

A USB hub is responsible for the following:

- Monitoring the insertion or removal of a device on its ports
- Power-managing individual devices on its ports
- Controlling power to its ports

The USB host controller has an embedded hub called the *root hub*. The ports that are visible at the system's back panel are the ports of the root hub. The USB host controller is responsible for the following:

- Directing the USB bus. Individual devices cannot arbitrate for the bus.
- Polling the devices by using a polling interval that is determined by the device. The device is assumed to have sufficient buffering to account for the time between the polls.
- Sending data between the USB host controller and its attached devices. Peer-to-peer communication is not supported.

USB Hub Devices

- Do not cascade hubs beyond four levels on either SPARC based or x86 based systems. On SPARC systems, the OpenBoot™ PROM cannot reliably probe beyond four levels of devices.
- Do not cascade bus-powered hubs. So, you cannot plug a bus-powered hub into another bus-powered hub. A bus-powered hub does not have its own power supply. A USB diskette device derives all its power from the bus and might not work on a bus-powered hub.

SPARC: USB Power Management

Suspending and resuming USB devices are fully supported on SPARC systems. However, do not suspend a devices that is busy and *never* remove a device when the system is powered off.

If the SPARC based system has power management enabled, the USB framework makes a best effort to power-manage all devices. Power-managing a USB device means that the hub driver suspends the port to which the device is connected. The device might or might not support remote wakeup. If the device supports remote wakeup, it wakes up the hub it is connected to, depending on the event, such as moving the mouse. The host system could also wake up the device if an application sends an I/O to the device.

All HID (keyboard, mouse, speakers, microphones), hub, and storage devices are power-managed by default if they support remote wakeup capability. A USB printer is power-managed only between two print jobs.

When you power-manage to reduce power consumption, USB leaf devices are powered down first. Then, after some delay, the parent hub is powered down. When all devices that are connected to this hub's ports are powered down, the hub is powered down after some delay. To achieve the most efficient power management, do not cascade many hubs.

Guidelines for USB Cables

Never use USB cable extenders that are available in the market. Always use a hub with longer cables to connect devices. Always use fully rated (12 Mbit/sec) 20/28 AWG cables for connecting USB devices.

Using USB Mass Storage Devices (Task Map)

Task	Description	For Instructions
1. Add a hot-pluggable USB mass storage device	Add a hot-pluggable USB mass storage device with `vold` running.	"How to Add a Hot-Pluggable USB Mass Storage Device With `vold` Running" on page 386
	Add a hot-pluggable USB mass storage device without `vold` running.	"How to Add a Hot-Pluggable USB Mass Storage Device Without `vold` Running" on page 387
2. Remove a hot-pluggable USB mass storage device	Remove a hot-pluggable USB mass storage device with `vold` running.	"How to Remove a Hot-Pluggable USB Mass Storage Device With `vold` Running" on page 387
	Remove a hot-pluggable USB mass storage device without `vold` running.	"How to Remove a Hot-Pluggable USB Mass Storage Device Without `vold` Running" on page 387

Task	Description	For Instructions
3. Mount a USB mass storage device	Mount a hot-pluggable USB mass storage device with `vold` running. Mount a hot-pluggable USB mass storage device without `vold` running.	"How to Mount or Unmount a USB Mass Storage Device With `vold` Running" on page 389 "How to Mount or Unmount a USB Mass Storage Device Without `vold` Running" on page 390
4. Add a hot-pluggable USB camera	Add a hot-pluggable USB camera to access digital images.	"How to Add a Hot-Pluggable USB Camera" on page 390

Using USB Mass Storage Devices

Removable mass storage devices such as USB CD-RWs, disks, DVDs, digital cameras, Zip, Jaz, Clik!, SmartMedia, CompactFlash, and ORB are supported, starting with the Solaris 9 release.

For a complete list of USB devices that are supported in the Solaris environment, see `http://www.sun.com/io_technologies/storagesolutions.html`.

These devices can be managed with or without volume management. For information on managing devices with volume management, see `vold`(1M).

Using Non-Compliant USB Mass Storage Devices

If you connect a USB mass storage device to a system running the Solaris 9 release and the system is unable to use it, you can check the `/kernel/drv/scsa2usb.conf` file to see if there is a matching, commented entry for this device. For a listing of recommended USB mass storage devices, go to `http://www.sun.com/io`.

Some devices might be supported by the USB mass storage driver even though they do not identify themselves as compliant with the USB mass storage class or identify themselves incorrectly. The `scsa2usb.conf` file contains an attribute-override-list that lists the vendor ID, product ID, and revision for matching mass storage devices, as well as fields for overriding the default device attributes. The entries in this list are commented out by default, and can be copied and uncommented to enable support of particular devices.

Follow the information given in the `scsa2usb.conf` file to see if a particular device can be supported by using the override information.

For more information, see `scsa2usb`(7D).

Hot-Plugging USB Devices

Hot-plugging a device means the device is added or removed without shutting down the operating system or powering off the system.

When you hot-plug a USB device, the device is immediately seen in the system's device hierarchy, as displayed in the `prtconf` command output. When you remove a USB device, the device is removed from the system's device hierarchy, unless the device is in use.

If the USB device is in use when it is removed, the hot-plug behavior is a little different. If a device is in use when it is unplugged, the device node remains, but the driver controlling this device stops all activity on the device. Any new I/O activity issued to this device is returned with an error.

In this situation, the system prompts you to plug in the original device. To recover from accidentally removing a busy USB device, do the following:

1. Plug the original device into the same port.
2. Stop the application that is using the device.
3. Remove the device.

The USB port remains unusable until the original device has been plugged in again. If the device is no longer available, the port remains unusable until the next reboot.

Note – Data integrity might be impaired if you remove an active or open device. Always close the device before removing, except the console keyboard and mouse, which can be moved while active.

▼ How to Add a Hot-Pluggable USB Mass Storage Device With `vold` Running

This procedure describes how to add a hot-pluggable USB device with `vold` running.

1. **Insert the USB mass storage device.**
2. **Restart `vold`.**

   ```
   # pkill -HUP vold
   ```

3. **Verify that the device has been added.**

```
$ ls device-alias
```

▼ How to Add a Hot-Pluggable USB Mass Storage Device Without vold Running

This procedure describes how to add a hot-pluggable USB device without vold running.

1. **Add a hot-pluggable USB device into the USB port.**
2. **Verify that the USB device has been added.**

```
$ ls /dev/rdsk/cntndnsn
```

▼ How to Remove a Hot-Pluggable USB Mass Storage Device With vold Running

The following procedure uses a Zip drive as an example of removing a hot-pluggable USB device with vold running.

1. **Unmount the device.**

```
$ volrmmount -e zip0
```

2. **Stop any active applications that are using the device.**
3. **Eject the device.**

```
$ eject zip0
```

4. **Become superuser and stop vold.**

```
# /etc/init.d/volmgt stop
```

5. **Remove the USB mass storage device.**
6. **Start vold.**

```
# /etc/init.d/volmgt start
```

▼ How to Remove a Hot-Pluggable USB Mass Storage Device Without vold Running

This procedure describes how to remove a hot-pluggable USB device without vold running.

1. **Become superuser.**

2. **Stop any active applications that are using the device.**
3. **Remove the hot-pluggable USB device.**
   a. **Unmount the device.**

      # **umount** */mount-point*

   b. **Remove the device.**

Mounting USB Mass Storage Devices With or Without `vold` Running

If you are running Solaris Common Desktop Environment (CDE), the USB removable mass storage devices are managed by the Removable Media Manager component of the CDE File Manager. For more information on the CDE File Manager, see `dtfile(1)`.

Note – You must include the `/usr/dt/man` directory in your `MANPATH` variable to display the man pages that are listed in this section. You must also have the `/usr/dt/bin` directory in your path and have CDE running to use these commands, or have a `DISPLAY` variable set to use these commands remotely.

The following table identifies the commands that Removable Media Manager uses to manage storage devices from the CDE environment.

Command	Man Page	Task
`sdtmedia_format`	`sdtmedia_format(1)`	Format and label a device
`sdtmedia_prop`	`sdtmedia_prop(1)`	Display properties of a device
`sdtmedia_prot`	`sdtmedia_prot(1)`	Change device protection
`sdtmedia_slice`	`sdtmedia_slice(1)`	Create or modify slices on a device

After the USB device is formatted, it is usually mounted under the `/rmdisk/`*label* directory. For more information on configuring removable storage devices, see `rmmount.conf`(4) or `vold.conf`(4).

The device nodes are created under the `/vol/dev` directory. For more information, see `scsa2usb(7D)`.

You can use USB mass storage devices without the volume manager (`vold`) running. Here are two ways to avoid using the volume manager.

- Stop `vold` by issuing this command:

  ```
  # /etc/init.d/volmgt stop
  ```

- Keep `vold` running, but do not register the USB mass storage devices with `vold`. Remove volume manager registration of USB mass storage devices by commenting the following line in the `/etc/vold.conf` file, like this:

  ```
  # use rmdisk drive /dev/rdsk/c*s2 dev_rmdisk.so rmdisk%d
  ```

  After this line is commented, restart `vold`.

  ```
  # /etc/init.d/volmgt start
  ```

Caution – If you comment out this line and other SCSI or ATAPI Zip or Jaz removable devices are in the system, `vold` registration for these devices would be disabled as well.

For more information, see `vold.conf`(4).

The following procedures describe how to manage USB mass storage devices without `vold` running. The device nodes are created under the `/dev/rdsk` directory for character devices and under the `/dev/dsk` directory for block devices. For more information, see `scsa2usb`(7D).

How to Mount or Unmount a USB Mass Storage Device With `vold` Running

1. **Display device aliases for all removable mass storage devices, including USB mass storage devices.**

   ```
   $ eject -n
   .
   .
   .
   rmdisk0 -> /vol/dev/rdsk/c4t0d0/clik40     (Generic USB storage)
   cdrom0 -> /vol/dev/rdsk/c0t6d0/audio_cd    (Generic CD device)
   zip1 -> /vol/dev/rdsk/c2t0d0/fat32         (USB Zip device)
   zip0 -> /vol/dev/rdsk/c1t0d0/zip100        (USB Zip device)
   jaz0 -> /vol/dev/rdsk/c3t0d0/jaz1gb        (USB Jaz device)
   ```

2. **Mount a USB mass storage device by using the device aliases listed previously.**

   ```
   $ volrmmount -i device-alias
   ```

   This example shows how to mounts USB Jaz drive (`/rmdisk/jaz0`).

   ```
   $ volrmmount -i jaz0
   ```

3. **Unmount a USB mass storage device.**

```
$ volrmmount -e device-alias
```

This example shows how to unmount a USB Zip drive (`/rmdisk/zip0`).

```
$ volrmmount -e zip0
```

4. **Eject a USB device from a generic USB drive.**

```
$ eject device-alias
```

For example:

```
$ eject rmdisk0
```

Note – The `eject` command also unmounts the device if the device is not unmounted already. The command also terminates any active applications that access the device.

How to Mount or Unmount a USB Mass Storage Device Without `vold` Running

1. **Become superuser.**
2. **Mount a USB mass storage device.**

```
# mount -F fs-type /dev/dsk/cntndnsn /mount-point
```

This command might fail if the device is read-only. Use the following command for CD-ROM devices.

```
# mount -F fs-type -o ro /dev/dsk/cntndnsn /mount-point
```

For example:

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /mnt
```

3. **Unmount a USB mass storage device.**

```
# umount /mount-point
```

4. **Eject the device.**

```
# eject /dev/[r]dsk/cntndnsn
```

▼ How to Add a Hot-Pluggable USB Camera

Use this procedure to add a hot-pluggable USB camera.

1. **Become superuser.**
2. **Plug in and turn on the USB camera.**

The USB subsystem and the SCSA subsystem create a logical device for the camera. After the camera is plugged in, output is written to the `/var/adm/messages` file.

3. **Examine the output that is written to the `/var/adm/messages` file.**

   Examining this output enables you to determine what logical device was created so that you can then use that device to access your images. The output will look similar to the following:

```
# more /var/adm/messages
Jul 15 09:53:35 buffy usba: [ID 349649 kern.info]       OLYMPUS, C-3040ZOOM,
 000153719068
Jul 15 09:53:35 buffy genunix: [ID 936769 kern.info] scsa2usb1 is
/pci@0,0/pci925,1234@7,2/storage@2
Jul 15 09:53:36 buffy scsi: [ID 193665 kern.info] sd3 at scsa2usb1:
target 0 lun 0
```

4. **Mount the USB camera file system.**

   The camera's file system is most likely a PCFS file system. In order to mount the file system on the device created, the slice that represents the disk must be specified. The slice is normally `s0` for a SPARC system, and `p0` for an x86 system.

   For example, to mount the file system on an x86 system, execute the following command:

```
# mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
```

   To mount the file system on a SPARC system, execute the following command:

```
# mount -F pcfs /dev/dsk/c3t0d0s0:c /mnt
```

   For information on mounting file systems, see Chapter 40.

   For information on mounting different PCFS file systems, see `mount_pcfs`(1M).

5. **Verify that the image files are available.**

   For example:

```
# ls /mnt/DCIM/100OLYMP/
P7220001.JPG*  P7220003.JPG*  P7220005.JPG*
P7220002.JPG*  P7220004.JPG*  P7220006.JPG*
```

6. **View and manipulate the image files created by the USB camera.**

```
# /usr/dt/bin/sdtimage &
```

7. **Unmount the file system before disconnecting the camera.**

   For example:

```
# umount /mnt
```

8. **Turn off and disconnect the camera.**

Using USB Audio Devices (Task Map)

Task	Description	For Instructions
1. Add hot-pluggable USB audio devices	Add a hot-pluggable USB microphone and speakers.	"How to Add Hot-Pluggable USB Audio Devices" on page 394
2. Identify your system's primary audio device	Identify which audio device is your primary audio device.	"How to Identify Your System's Primary Audio Device" on page 394
3. Change the primary USB audio device	You might want to make one particular audio device the primary audio device if you remove or change your USB audio devices.	"How to Change the Primary USB Audio Device" on page 395
4. Remove unused USB audio device links	If you remove a USB audio device while the system is powered off, the `/dev/audio` device might be pointing to a `/dev/sound/*` device that doesn't exist.	"How to Remove Unused USB Audio Device Links" on page 397
5. Troubleshoot USB audio device problems	You might have to power cycle USB speakers.	"Solving USB Speaker Problems" on page 398

Using USB Audio Devices

This Solaris release provides USB audio support which is implemented by a pair of cooperating drivers, `usb_ac` and `usb_as`. The audio control driver, `usb_ac`, a USBA (Solaris USB Architecture) compliant client driver provides the controlling interface to user applications. The audio streaming driver, `usb_as`, is provided to process audio data messages during play and record and set sample frequency, precision, and encoding requests from the `usb_ac` driver.

Both drivers comply to the USB audio class 1.0 specification.

Solaris supports external USB audio devices that are play-only or record-only. Onboard USB audio devices are not supported. For supported audio data format information, see `usb_ac`(7D).

- Only USB audio devices with one volume, bass, or treble control are supported. See the USB audio class specification for more information at `http://www.usb.org`.
- USB audio devices are supported on SPARC Ultra and x86 platforms that provide USB connectors.
- Hot-plugging USB audio devices is supported.
- USB audio devices must support a continuous sample rate of between 8000 and 48000 Hz or must support a 48000 Hz sample rate to play or record on the Solaris 8 10/01, Solaris 8 2/02, or Solaris 9 release.

The primary audio device is `/dev/audio`. You can verify that `/dev/audio` is pointing to USB audio by using the following command:

```
% mixerctl
Device /dev/audioctl:
  Name    = USB Audio
  Version = 1.0
  Config  = external

Audio mixer for /dev/audioctl is enabled
```

After you connect your USB audio devices, you access them with the `audioplay` and `audiorecord` command through the following files:

`/dev/sound/`*N*

You can select a specific audio device by setting the `AUDIODEV` environment variable or by specifying the `-d` option to the `audioplay` and `audiorecord` commands. However, setting `AUDIODEV` does not work for applications that have `/dev/audio` hardcoded as the audio file.

When you plug in a USB audio device, it automatically becomes the primary audio device, `/dev/audio`, unless `/dev/audio` is in use. For instructions on changing `/dev/audio` from onboard audio to USB audio and vice versa, refer to "How to Change the Primary USB Audio Device" on page 395 and `usb_ac`(7D).

Hot-Plugging Multiple USB Audio Devices

If a USB audio device is plugged into a system, it becomes the primary audio device, `/dev/audio`. It remains the primary audio device even after the system is rebooted. If additional USB audio devices are plugged in, the last one becomes the primary audio device.

For additional information on troubleshooting USB audio device problems, see `usb_ac(7D)`.

▼ How to Add Hot-Pluggable USB Audio Devices

Use the following procedure to add hot-pluggable USB audio devices.

1. **Plug in the USB speakers and microphone.**

   The primary audio device, `/dev/audio`, usually points to the onboard audio. After you connect USB audio devices, `/dev/audio` points to the USB audio devices that are identified in the `/dev/sound` directory.

2. **Verify that the audio device files have been created.**

```
% ls /dev/sound
0     0ctl  1     1ctl  2     2ctl
```

3. **Test the left and right USB speakers.**

```
% cd /usr/share/audio/samples/au
% audioplay -d /dev/sound/1 -b 100 spacemusic.au
% audioplay -d /dev/sound/1 -b -100 spacemusic.au
```

4. **Test the USB microphone.**

```
% cd $HOME/au
% audiorecord -d /dev/sound/2 -p mic -t 30 test.au
```

▼ How to Identify Your System's Primary Audio Device

This procedure assumes that you have already connected USB audio devices.

1. **Identify the state of your current audio device links.**

   For example:

```
% ls -lt /dev/audio*
lrwxrwxrwx   1 root     root          7 Jul 23 15:41 /dev/audio -> sound/0
lrwxrwxrwx   1 root     root         10 Jul 23 15:41 /dev/audioctl ->
sound/0ctl
% ls -lt /dev/sound/*
lrwxrwxrwx   1 root     other       66 Jul 23 14:21 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx   1 root     other       69 Jul 23 14:21 /dev/sound/0ctl ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioctl
%
```

   The primary audio device, `/dev/audio`, is currently pointing to the onboard audio, which is `/dev/sound/0`.

2. **(Optional) Add a new USB audio device.**

3. **Examine your system's new audio links.**

   For example:

```
% ls -lt /dev/audio*
lrwxrwxrwx   1 root     root          7 Jul 23 15:46 /dev/audio -> sound/1
lrwxrwxrwx   1 root     root         10 Jul 23 15:46 /dev/audioctl ->
sound/1ctl
% ls -lt /dev/sound/*
lrwxrwxrwx   1 root     root         74 Jul 23 15:46 /dev/sound/1 ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,audio
lrwxrwxrwx   1 root     root         77 Jul 23 15:46 /dev/sound/1ctl ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,aud...
lrwxrwxrwx   1 root     other        66 Jul 23 14:21 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx   1 root     other        69 Jul 23 14:21 /dev/sound/0ctl ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioctl
%
```

Notice that the primary audio device, `/dev/audio`, is pointing to the newly plugged in USB audio device, `/dev/sound/1`.

If you remove the USB audio device now, the primary audio device, `/dev/audio`, does not revert back to the onboard audio. See the following procedure for instructions on changing the primary audio device back to the system's onboard audio.

You can also examine your system's USB audio devices with the `prtconf` command and look for the USB device information.

```
% prtconf
.
.
.
usb, instance #0
   hub, instance #0
        mouse, instance #0
        keyboard, instance #1
        device, instance #0
            sound-control, instance #0
            sound, instance #0
            input, instance #0
.
.
.
```

▼ How to Change the Primary USB Audio Device

Follow these steps if you remove or change your USB audio devices and you want to make one particular audio device the primary audio device. The procedure changes the primary audio device to the onboard audio device as an example.

1. **Become superuser.**
2. **Close all audio applications.**
3. **Determine whether the audio and USB drivers are loaded.**

```
# modinfo | grep -i audio
124 780e6a69   bb6e   -   1  audiosup (Audio Device Support 1.12)
# modinfo  | grep -i usb
 48  13dba67  18636 199   1  ohci (USB OpenHCI Driver 1.31)
 49 78020000  1dece   -   1  usba (USBA: USB Architecture 1.37)
 50  12e5f1f    35f 195   1  hubd (USB Hub Driver 1.4)
 51  13ef53d   5e26 194   1  hid (USB HID Client Driver 1.16)
 54  13f67f2   1b42  10   1  usbms (USB mouse streams 1.6)
 56  127bbf0   2c74  11   1  usbkbm (USB keyboard streams 1.17)
#
```

4. **Load and attach the onboard audio driver.**

```
# devfsadm -i audiocs
```

The onboard audio driver is `audiocs` on a Sunblade 1000, and `audiots` on a Sunblade 100.

5. **Verify that the primary audio device link is pointing to the onboard audio.**

```
# ls -lt /dev/audio*
lrwxrwxrwx   1 root     other        7 Jul 23 15:49 /dev/audio -> sound/0
lrwxrwxrwx   1 root     other       10 Jul 23 15:49 /dev/audioctl ->
sound/0ctl
# ls -lt /dev/sound/*
lrwxrwxrwx   1 root     other       66 Jul 23 14:21 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx   1 root     other       69 Jul 23 14:21 /dev/sound/0ctl ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audioctl
#
```

6. **Confirm the onboard audio is the primary audio device.**

```
% audioplay /usr/demo/SOUND/sounds/bark.au
```

The `audioplay` command defaults to the onboard audio device.

7. **(Optional) Unload all the audio drivers that can be unloaded before plugging in another USB audio device.**

   a. **Close all the audio applications.**

   b. **Display the audio driver information to verify that no audio drivers are currently loaded.**

```
# modinfo | grep -i audio
 60 78048000   bb6e   -   1  audiosup (Audio Device Support 1.12)
 61 78152000  39a97   -   1  mixer (Audio Mixer 1.49)
 62 78118000   bf9f   -   1  amsrc1 (Audio Sample Rate Conv. #1 1.3)
128 7805e000  14968  54   1  audiocs (CS4231 mixer audio driver 1.21)
#
```

   c. **Unload the audio drivers.**

```
# modunload -i 0
# modinfo | grep -i audio
 60 78048000   bb6e   -   1  audiosup (Audio Device Support 1.12)
```

```
 61 78152000  39a97   -   1  mixer (Audio Mixer 1.49)
#
```

At this point, `audiocs`, the onboard audio driver, has been unloaded and guaranteed not to be open. However, the primary audio device, `/dev/audio`, does not change if it is held open by an application.

8. **(Optional) Plug in a USB audio device.**

9. **(Optional) Examine the new audio links.**

```
% ls -lt /dev/audio*
lrwxrwxrwx   1 root     root          7 Jul 23 16:12 /dev/audio -> sound/1
lrwxrwxrwx   1 root     root         10 Jul 23 16:12 /dev/audioctl ->
sound/1ctl
% ls -lt /dev/sound/*
lrwxrwxrwx   1 root     root         77 Jul 23 16:12 /dev/sound/1ctl ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,aud...
lrwxrwxrwx   1 root     root         74 Jul 23 16:12 /dev/sound/1 ->
../../devices/pci@1f,4000/usb@5/hub@1/device@3/sound-control@0:sound,aud...
lrwxrwxrwx   1 root     root         66 Jul 23 15:59 /dev/sound/0 ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,audio
lrwxrwxrwx   1 root     root         69 Jul 23 15:59 /dev/sound/0ctl ->
../../devices/pci@1f,4000/ebus@1/SUNW,CS4231@14,200000:sound,aud...
%
```

▼ How to Remove Unused USB Audio Device Links

Use this procedure if a USB audio device is removed while the system is powered off. It is possible that removing the USB audio device while the system is powered off will leave the `/dev/audio` device still pointing to a `/dev/sound/*` device that doesn't exist.

1. **Become superuser.**

2. **Close all audio applications.**

3. **Remove the unused audio links.**

```
# devfsadm -C -c audio
```

Troubleshooting USB Audio Device Problems

This section describes how to troubleshoot USB audio device problems.

Solving USB Speaker Problems

Sometimes USB speakers do not produce any sound even though the driver is attached and the volume is set to high. Hot-plugging the device might not change this behavior.

The workaround is to power cycle the USB speakers.

Key Points of Audio Device Ownership

Keep the following key points of audio device ownership in mind when working with audio devices.

- When you plug in a USB audio device and you are logged in on the console, the console is the owner of the `/dev/*` entries. This situation means you can use the audio device as long as you are logged into the console.
- If you are not logged into the console when you plug in a USB audio device, root becomes the owner of the device. However, if you log into the console and attempt to access the USB audio device, device ownership changes to the console. For more information, see `logindevperm`(4).
- When you remotely login with the `rlogin` command and attempt to access the USB audio device, the ownership does not change. This means that, for example, unauthorized users cannot listen to conversations over a microphone owned by someone else.

Hot-Plugging USB Devices With the `cfgadm` Command (Task Map)

Task	Description	For Instructions
1. Display USB device information	Display information about USB devices and buses.	"How to Display USB Device Information" on page 400
2. Unconfigure a USB device	Logically unconfigure a USB device that is still physically connected to the system.	"How to Unconfigure a USB Device" on page 401
3. Configure a USB device	Configure a USB device that was previously unconfigured.	"How to Configure a USB Device" on page 402

Task	Description	For Instructions
4. Logically disconnect a USB device	You can logically disconnect a USB device if you are not physically near the system.	"How to Logically Disconnect a USB Device" on page 402
5. Logically connect a USB device	Logically connect a USB device that was previously logically disconnected or unconfigured.	"How to Logically Connect a USB Device" on page 402
6. Disconnect a USB device subtree	Disconnect a USB device subtree, which is the hierarchy (or tree) of devices below a hub.	"How to Logically Disconnect a USB Device Subtree" on page 403
7. Reset a USB device	Reset a USB device to logically remove and recreate the device.	"How to Reset a USB Device" on page 403

Hot-Plugging USB Devices With the `cfgadm` Command

You can add and remove a USB device from a running system without using the `cfgadm` command. However, a USB device can also be *logically* hot-plugged without physically removing the device. This scenario is convenient when you are working remotely and you need to disable or reset a non-functioning USB device. The `cfgadm` command also provides a way to display the USB device tree including manufacturer and product information.

The `cfgadm` command displays information about attachment points, which are locations in the system where dynamic reconfiguration operations can occur. An attachment point consists of:

- An occupant, which represents a hardware resource that might be configured into the system, and
- A receptacle, which is the location that accepts the occupant.

Attachment points are represented by logical and physical attachment point IDs (`Ap_Id`s). The physical `Ap_Id` is the physical pathname of the attachment point. The logical `Ap_Id` is a user-friendly alternative for the physical `Ap_Id`. For more information on `Ap_Id`s, see `cfgadm_usb`(1M).

The `cfgadm` command provides the following USB device status information.

Receptacle State	Description
empty/unconfigured	The device is not connected.
disconnected/unconfigured	The device is logically disconnected and unavailable. The devinfo node is removed even though the device could still be physically connected.
connected/unconfigured	The device is logically connected, but unavailable. The devinfo node is present.
connected/configured	The device is connected and available.

The following sections describe how to hot-plugging a USB device with the `cfgadm` command. All of the sample USB device information in these sections has been truncated to focus on relevant information.

▼ How to Display USB Device Information

Use the `prtconf` command to display information about USB devices.

```
$ prtconf
        usb, instance #0
                hub, instance #2
                    device, instance #8
                        interface (driver not attached)
                    printer (driver not attached)
                    mouse, instance #14
                    device, instance #9
                        keyboard, instance #15
                        mouse, instance #16
                    storage, instance #7
                        disk (driver not attached)
                    communications, instance #10
                        modem (driver not attached)
                        data (driver not attached)
                storage, instance #0
                    disk (driver not attached)
                storage, instance #1
                    disk (driver not attached)
```

Use the `cfgadm` command to display USB bus information. For example:

```
% cfgadm
Ap_Id                       Type         Receptacle   Occupant     Condition
usb0/4.5                    usb-hub      connected    configured   ok
usb0/4.5.1                  usb-device   connected    configured   ok
usb0/4.5.2                  usb-printer  connected    configured   ok
usb0/4.5.3                  usb-mouse    connected    configured   ok
usb0/4.5.4                  usb-device   connected    configured   ok
usb0/4.5.5                  usb-storage  connected    configured   ok
```

```
usb0/4.5.6                 usb-communi  connected    configured   ok
usb0/4.5.7                 unknown      empty        unconfigured ok
usb0/4.6                   usb-storage  connected    configured   ok
usb0/4.7                   usb-storage  connected    configured   ok
```

In the preceding example, `usb0/4.5.1` identifies a device connected to port 1 of the second-level external hub, which is connected to port 5 of first-level external hub, which is connected to the first USB controller's root hub, port 4.

Use the following `cfgadm` command to display specific USB device information. For example:

```
% cfgadm -l -s "cols=ap_id:info"
Ap_Id       Information
usb0/4.5.1  Mfg: Inside Out Networks Product: Edgeport/421 NConfigs: 1 Config: 0  : ...
usb0/4.5.2  Mfg: <undef> Product: <undef>   NConfigs: 1 Config: 0  <no cfg str descr>
usb0/4.5.3  Mfg: Mitsumi Product: Apple USB Mouse NConfigs: 1 Config: 0 <no cfg str descr>
usb0/4.5.4  Mfg: NMB  Product: NMB USB KB/PS2 M NConfigs: 1 Config: 0
usb0/4.5.5  Mfg: Hagiwara Sys-Com  Product: SmartMedia R/W  NConfigs: 1 Config: 0 : Default
usb0/4.5.6  Mfg: 3Com Inc.  Product: U.S.Robotics 56000 Voice USB Modem  NConfigs: 2 ...
usb0/4.5.7
usb0/4.6    Mfg: Iomega  Product: USB Zip 250  NConfigs: 1  Config: 0  : Default
usb0/4.7    Mfg: Iomega  Product: USB Zip 100  NConfigs: 1  Config: 0  : Default
#
```

▼ How to Unconfigure a USB Device

You can unconfigure a USB device that is still physically connected to the system, but a driver will never attach to it. After the USB device is unconfigured, the device is visible in the `prtconf` output.

1. **Become superuser.**
2. **Unconfigure the USB device.**

```
# # cfgadm -c unconfigure usb0/4.7
Unconfigure the device: /devices/pci@8,700000/usb@5,3/hub@4:4.7
This operation will suspend activity on the USB device
Continue (yes/no)? y
```

3. **Verify that the device is unconfigured.**

```
# cfgadm
Ap_Id                      Type         Receptacle   Occupant     Condition
usb0/4.5                   usb-hub      connected    configured   ok
usb0/4.5.1                 usb-device   connected    configured   ok
usb0/4.5.2                 usb-printer  connected    configured   ok
usb0/4.5.3                 usb-mouse    connected    configured   ok
usb0/4.5.4                 usb-device   connected    configured   ok
usb0/4.5.5                 usb-storage  connected    configured   ok
usb0/4.5.6                 usb-communi  connected    configured   ok
usb0/4.5.7                 unknown      empty        unconfigured ok
```

```
usb0/4.6                    usb-storage  connected    configured   ok
usb0/4.7                    usb-storage  connected    unconfigured ok
```

▼ How to Configure a USB Device

1. **Become superuser.**
2. **Configure a USB device.**

   ```
   # cfgadm -c configure usb0/4.7
   ```

3. **Verify that the USB device is configured.**

   ```
   # cfgadm usb0/4.7
   Ap_Id                  Type         Receptacle  Occupant    Condition
   usb0/4.7               usb-storage  connected   configured  ok
   ```

▼ How to Logically Disconnect a USB Device

If you want to remove a USB device from the system and the `prtconf` output, but you are not physically near the system, just logically disconnect the USB device. The device is still physically connected, but it is logically disconnected, unusable, and not visible to the system.

1. **Become superuser.**
2. **Disconnect a USB device.**

   ```
   # cfgadm -c disconnect -y usb0/4.7
   ```

3. **Verify that the device is disconnected.**

   ```
   # cfgadm usb0/4.7
   Ap_Id                  Type       Receptacle    Occupant      Condition
   usb0/4.7               unknown    disconnected  unconfigured  ok
   ```

▼ How to Logically Connect a USB Device

Use this procedure to logically connect a USB device that was previously logically disconnected or unconfigured.

1. **Become superuser.**
2. **Connect a USB device.**

   ```
   # cfgadm -c configure usb0/4.7
   ```

3. **Verify that the device is connected.**

```
# cfgadm usb0/4.7
Ap_Id                  Type         Receptacle  Occupant    Condition
usb0/4.7               usb-storage  connected   configured  ok
```

The device is now available and visible to the system.

▼ How to Logically Disconnect a USB Device Subtree

Use this procedure to disconnect a USB device subtree, which is the hierarchy (or tree) of devices below a hub.

1. **Become superuser.**
2. **Remove a USB device subtree.**

   ```
   # cfgadm -c disconnect -y usb0/4
   ```

3. **Verify that the USB device subtree is disconnected.**

   ```
   # cfgadm usb0/4
   Ap_Id                  Type       Receptacle   Occupant     Condition
   usb0/4                 unknown    disconnected unconfigured ok
   ```

▼ How to Reset a USB Device

If a USB device behaves erratically, use the `cfgadm` command to reset the device, which logically removes and recreates the device.

1. **Become superuser.**
2. **Make sure the device is not in use.**
3. **Reset the device.**

   ```
   # cfgadm -x usb_reset -y usb0/4.7
   ```

4. **Verify that the device is connected.**

   ```
   # cfgadm usb0/4.7
   Ap_Id                  Type         Receptacle   Occupant     Condition
   usb0/4.7               usb-storage  connected    configured   ok
   ```

CHAPTER 30

Accessing Devices (Overview)

This chapter provides information about how to access the devices on a system.

This is a list of the overview information in this chapter.

- "Accessing Devices" on page 405
- "Logical Disk Device Names" on page 407
- "Logical Tape Device Names" on page 410
- "Logical Removable Media Device Names" on page 411

For overview information about configuring devices, see Chapter 27.

Accessing Devices

You need to know how to specify device names when using commands to manage disks, file systems, and other devices. In most cases, you can use logical device names to represent devices that are connected to the system. Both logical and physical device names are represented on the system by logical and physical device files.

How Device Information Is Created

When a system is booted for the first time, a device hierarchy is created to represent all the devices connected to the system. The kernel uses the device hierarchy information to associate drivers with their appropriate devices, and provides a set of pointers to the drivers that perform specific operations. For more information on device hierarchy, see *OpenBoot 3.x Command Reference Manual*.

How Devices Are Managed

The `devfsadm` command manages the special device files in the `/dev` and `/devices` directories. By default, the `devfsadm` command attempts to load every driver in the system and attach to all possible device instances. Then, `devfsadm` creates the device files in the `/devices` directory and the logical links in the `/dev` directory. In addition to managing the `/dev` and `/devices` directories, the `devfsadm` command also maintains the `path_to_inst`(4) instance database.

Both reconfiguration boot processing and updating the `/dev` and `/devices` directories in response to dynamic reconfiguration events is handled by `devfsadmd`, the daemon version of the `devfsadm` command. This daemon is started from the `/etc/rc*` scripts when a system is booted.

Since the `devfsadmd` daemon automatically detects device configuration changes generated by any reconfiguration event, there is no need to run this command interactively.

For more information, see `devfsadm`(1M).

Device Naming Conventions

Devices are referenced in three ways in the Solaris environment.

- Physical device name – Represents the full device pathname in the device information hierarchy. Physical device names are displayed by using the following commands:
  - `dmesg`
  - `format`
  - `sysdef`
  - `prtconf`

  Physical device files are found in the `/devices` directory.
- Instance name – Represents the kernel's abbreviation name for every possible device on the system. For example, `sd0` and `sd1` represent the instance names of two disk devices. Instance names are mapped in the `/etc/path_to_inst` file and are displayed by using the following commands:
  - `dmesg`
  - `sysdef`
  - `prtconf`
- Logical device name – Used with most file system commands to refer to devices. For a list of file commands that use logical device names, see Table 30–1. Logical device files in the `/dev` directory are symbolically linked to physical device files in the `/devices` directory.

Logical Disk Device Names

Logical device names are used to access disk devices when you:

- Add a new disk to the system
- Move a disk from one system to another system
- Access or mount a file system residing on a local disk
- Back up a local file system

Many administration commands take arguments that refer to a disk slice or file system.

Refer to a disk device by specifying the subdirectory to which it is symbolically linked, either `/dev/dsk` or `/dev/rdsk`, followed by a string identifying the particular controller, disk, and slice.

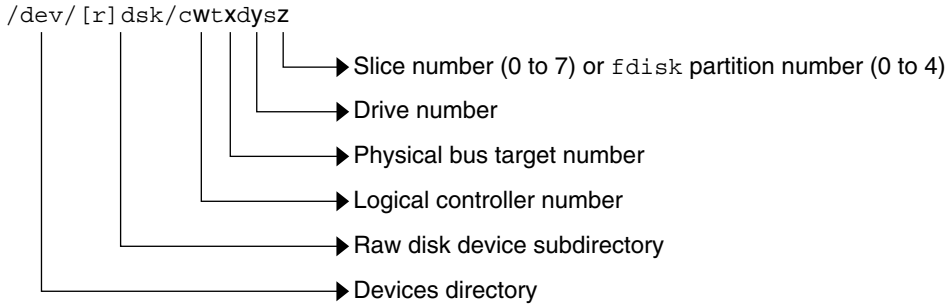


FIGURE 30–1 Logical Device Names

Specifying the Disk Subdirectory

Disk and file administration commands require the use of either a *raw* (or *character*) device interface, or a *block* device interface. The distinction is made by how data is read from the device.

Raw device interfaces transfer only small amounts of data at a time. Block device interfaces include a buffer from which large blocks of data are read at once.

Different commands require different interfaces.

- When a command requires the raw device interface, specify the `/dev/rdsk` subdirectory. (The "r" in `rdsk` stands for "raw.")
- When a command requires the block device interface, specify the `/dev/dsk` subdirectory.

- When you are not sure whether a command requires use of `/dev/dsk` or `/dev/rdsk`, check the man page for that command.

The following table shows which interface is required for some commonly used disk and file system commands.

TABLE 30–1 Device Interface Type Required by Some Frequently Used Commands

Command	Interface Type	Example of Use
`df`(1M)	Block	`df /dev/dsk/c0t3d0s6`
`fsck`(1M)	Raw	`fsck -p /dev/rdsk/c0t0d0s0`
`mount`(1M)	Block	`mount /dev/dsk/c1t0d0s7 /export/home`
`newfs`(1M)	Raw	`newfs /dev/rdsk/c0t0d1s1`
`prtvtoc`(1M)	Raw	`prtvtoc /dev/rdsk/c0t0d0s2`

Specifying the Slice

The string that you use to identify a specific slice on a specific disk depends on the controller type, either direct or bus-oriented. The following table describes the different types of direct or bus-oriented controllers on different platforms.

TABLE 30–2 Controller Types

Direct controllers	Bus-Oriented Controllers
IDE (x86)	SCSI (SPARC/x86)
	FCAL (SPARC)
	ATA (SPARC/x86)

The conventions for both types of controllers are explained in the following subsections.

Note – Controller numbers are assigned automatically at system initialization. The numbers are strictly logical and imply no direct mapping to physical controllers.

x86: Disks With Direct Controllers

To specify a slice on a disk with an IDE controller on an x86 based system, follow the naming convention shown in the following figure.

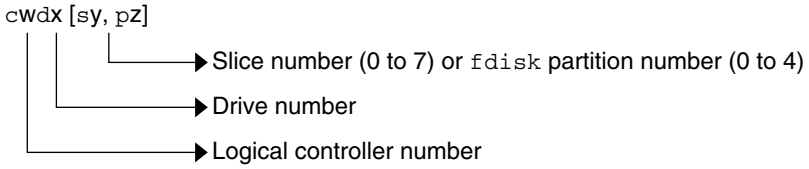


FIGURE 30–2 x86: Disks with Direct Controllers

To indicate the entire Solaris `fdisk` partition, specify slice 2 (`s2`).

If you have only one controller on your system, *w* is usually 0.

SPARC: Disks With Bus-Oriented Controllers

To specify a slice on a disk with a bus-oriented controller, SCSI for instance, on a SPARC based system, follow the naming convention shown in the following figure.

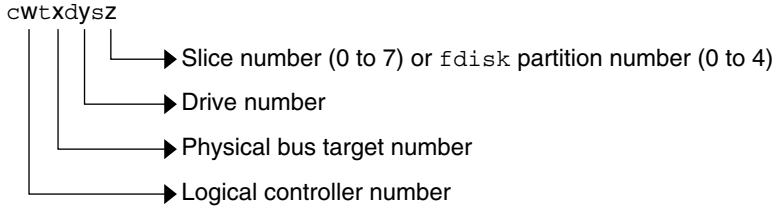


FIGURE 30–3 SPARC: Disks With Bus-Oriented Controllers

On a SPARC based system with directly connected disks such as the IDE disks on a Ultra10, the naming convention is the same as that for systems with bus-oriented controllers.

If you have only one controller on your system, *w* is usually 0.

For SCSI controllers, *x* is the target address set by the switch on the back of the unit, and *y* is the logical unit number (LUN) of the drive attached to the target. If the disk has an embedded controller, *y* is usually 0.

To indicate the whole disk, specify slice 2 (`s2`).

x86: Disks With SCSI Controllers

To specify a slice on a disk with a SCSI controller on an x86 based system, follow the naming convention shown in the following figure.

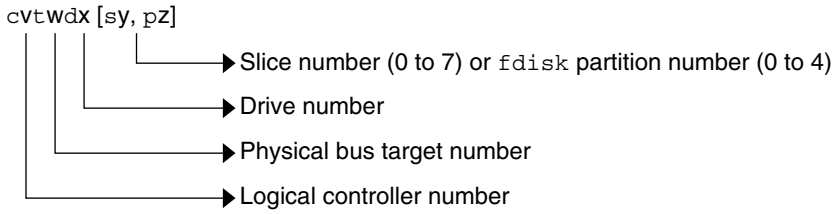


FIGURE 30–4 x86: Disks with SCSI Controllers

If you have only one controller on your system, *v* is usually 0.

For SCSI controllers, *w* is the target address set by the switch on the back of the unit, and *x* is the logical unit number (LUN) of the drive attached to the target. If the disk has an embedded controller, *x* is usually 0.

To indicate the entire Solaris `fdisk` partition, specify slice 2 (`s2`).

Logical Tape Device Names

Logical tape device files are found in the `/dev/rmt/*` directory as symbolic links from the `/devices` directory.

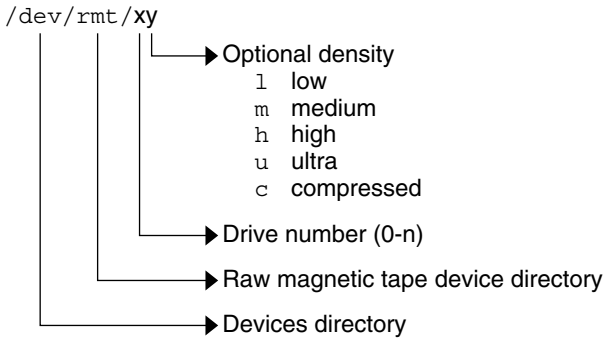


FIGURE 30–5 Logical Tape Device Names

The first tape device connected to the system is 0 (`/dev/rmt/0`). Tape density values (`l`, `m`, `h`, `c`, and `u`) are described in Chapter 52.

Logical Removable Media Device Names

Since removable media is managed by volume management (`vold`), the logical device name is usually not used unless you want to mount the media manually.

The logical device name that represents the removable media devices on a system are described in Chapter 18.

CHAPTER 31

Managing Disks Topics

This topic map lists the chapters that provide information on managing disks.

Chapter 32	Provides an overview of Solaris disk slices and an introduction to the `format` utility.
Chapter 33	Provides step-by-step instructions for formatting a disk, examining disk labels, and repairing a defective disk sector.
Chapter 34	Provides step-by-step instructions for adding a disk to a SPARC based system.
Chapter 35	Provides step-by-step instructions for adding a disk to an x86 based system.
Chapter 36	Provides a description of the `format` utility's menu and commands. This chapter also includes information about the `format.dat` file, rules for providing input to `format` commands, and instructions on using the help facility.

CHAPTER 32

Managing Disks (Overview)

This chapter provides overview information about Solaris disk slices and introduces the `format` utility.

This is a list of overview information in this chapter.

- "What's New in Disk Management in the Solaris 9 4/03 Release?" on page 415
- "Where to Find Disk Management Tasks" on page 420
- "Overview of Disk Management" on page 420
- "Disk Terminology" on page 420
- "About Disk Slices" on page 421
- "The `format` Utility" on page 426
- "About Disk Labels" on page 430
- "Dividing a Disk Into Slices" on page 432

For instructions on how to add a disk to your system, see Chapter 34 or Chapter 35.

What's New in Disk Management in the Solaris 9 4/03 Release?

This section describes new disk management features in the Solaris 9 4/03 release.

Overview of the EFI Disk Label

This Solaris release provides support for disks that are larger than 1 Tbyte on systems running a 64-bit Solaris kernel. The Extensible Firmware Interface (EFI) disk label is not available for disks connected to a system running a 32-bit Solaris kernel, such as a system running the Solaris x86 Platform Edition.

You can download the EFI specification at `http://www.intel.com/technology/efi/main_specification.htm`.

The EFI label provides support for physical disks and virtual disk volumes. This release also includes updated disk utilities for managing disks greater than 1 Tbyte. The UFS file system is compatible with the EFI disk label, but you cannot create a UFS file system greater than 1 Tbyte. The Solaris Volume Manager software can be used to manage disks greater than 1 Tbyte in this Solaris release. For information on using Solaris Volume Manager, see *Solaris Volume Manager Administration Guide*.

The VTOC label is still available for disks less than 1 Tbyte in size. If you are only using disks smaller than 1 Tbyte on your systems, managing disks will be the same as in previous Solaris releases. In addition, you can use the `format -e` command to label a disk less than 1 Tbyte with an EFI label. For more information, see "Example—Labeling a Disk Less Than 1 Tbyte with an EFI Label" on page 445.

Comparison of the EFI Label and the VTOC Label

The EFI disk label differs from the VTOC disk label in the following ways:

- Provides support for disks greater than 1 Tbyte in size.
- Provides slices 0–6, where slice 2 is just another slice.
- Partitions (or slices) cannot overlap with the primary or backup label, nor with any other partitions. The size of the EFI label is usually 34 sectors, so partitions start at sector 34. This feature means no partition can start at sector zero (0).
- There is no cylinder, head, or sector information stored in the label. Sizes are reported in blocks.
- Information that was stored in the alternate cylinders area, the last two cylinders of the disk, is now stored in slice 8.
- If you use the `format` utility to change partition sizes, the `unassigned` partition tag is assigned to partitions with sizes equal to zero. By default, the `format` utility assigns the `usr` partition tag to any partition with a size greater than zero. You can use the partition change menu to reassign partition tags after the partitions are changed. However, you cannot change a partition with a non-zero size to the `unassigned` partition tag.

Restrictions of the EFI Disk Label

Keep the following restrictions in mind when determining whether using disks greater than 1 Tbyte is appropriate for your environment:

- The layered software products intended for systems with EFI-labeled disks might be incapable of accessing a disk with an EFI disk label.
- A disk with an EFI disk label is not recognized on systems running previous Solaris releases.

- You cannot boot from a disk with an EFI disk label.
- You cannot use the Solaris Management Console's Disk Manager Tool to manage disks with EFI labels. Use the `format` utility or the Solaris Management Console's Enhanced Storage Tool to manage disks with EFI labels, after you use the `format` utility to partition the disk.
- The EFI specification prohibits overlapping slices. The whole disk is represented by *cxtydz*.
- UFS file systems are compatible with the EFI disk label. However, you cannot create a UFS file system that is greater than 1 Tbyte. The unbundled Sun QFS file system is available if you need to create file systems greater than 1 Tbyte. For information on the Sun QFS file system, see `http://docs.sun.com/db/doc/816-2542-10`.
- Provides information about disk or partition sizes in sectors and blocks, but not in cylinders and heads.
- The following `format` options are either not supported or are not applicable on disks with EFI labels:
  - The `save` option is not supported because disks with EFI labels do not need an entry in the `format.dat` file.
  - The `backup` option is not applicable because the disk driver finds the primary label and writes it back to the disk.

Installing a System With an EFI-Labeled Disk

The Solaris installation utilities automatically recognize disks with EFI labels, but cannot use the Solaris installation utilities to repartition these disks. You must use the `format` utility to repartition this disk before or after installation. The Solaris Upgrade and Live Upgrade utilities also recognize a disk with an EFI label. However, you cannot boot a system from an EFI-labeled disk.

After the Solaris release is installed on a system with an EFI-labeled disk, the partition table looks similar to the following:

```
Current partition table (original):
Total disk sectors available: 2479267806 blocks

Part      Tag    Flag     First Sector          Size          Last Sector
  0       home    wm                34         1.15TB           2479251421
  1 unassigned    wm                 0            0                    0
  2 unassigned    wm                 0            0                    0
  3 unassigned    wm                 0            0                    0
  4 unassigned    wm                 0            0                    0
  5 unassigned    wm                 0            0                    0
  6 unassigned    wm                 0            0
```

Managing Disks With EFI Disks Labels

Use the following table to locate information on managing disks with EFI disk labels.

Task	For More Information
If the system is already installed, connect the disk to the system and perform a reconfiguration boot.	"SPARC: Adding a System Disk or a Secondary Disk (Task Map)" on page 457
Repartition the disk with the `format` utility, if necessary.	"SPARC: How to Create Disk Slices and Label a Disk" on page 460
Create disk volumes, and if needed, create soft partitions with Solaris Volume Manager.	"Storage Management Concepts" in *Solaris Volume Manager Administration Guide*
Create UFS file systems for the new disk with the `newfs` command if the file systems are less than 1 Tbyte.	"SPARC: How to Create File Systems" on page 465
Or, create a QFS file system.	`http://docs.sun.com/db/coll/20445.2`

Cloning a Disk with an EFI Label

In previous Solaris releases, slice 2 (`s2`) was used to represent the whole disk. You could use the `dd` command to clone or copy disks by using syntax similar to the following:

```
dd if=/dev/rdsk/c0t0d0s2 of=/dev/rdsk/c0t2d0s2 bs=128k
```

Now, you must use a slightly different procedure to clone or copy disks larger than 1 Tbyte so that the UUID of cloned disks are unique. For example:

1. Use the `dd` command to clone the disk with an EFI label:

   ```
   # dd if=/dev/rdsk/c0t0d0s2 of=/dev/rdsk/c0t2d0s2 bs=128k
   ```

2. Pipe the `prtvtoc` output of the disk to be copied to the `fmthard` command to create a new label for the cloned disk.

   ```
   # prtvtoc /dev/rdsk/c0t0d0 | fmthard -s - /dev/rdsk/c0t2d0
   ```

Caution – If you do not create a new label for the cloned disk, other software products might corrupt data on EFI-labeled disks if they encounter duplicate UUIDs.

Troubleshooting Problems With EFI Disk Labels

Use the following error messages and solutions to troubleshooting problems with EFI-labeled disks.

Error Message

```
Dec  3 09:26:48 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/
sf@1,0/ssd@w50020f23000002a4,0 (ssd1):
Dec  3 09:26:48 holoship disk has 2576941056 blocks, which is too large
for a 32-bit kernel
```

Cause

You attempted to boot a system running a 32-bit SPARC kernel with a disk greater than 1 Tbyte.

Solution

Boot a system running a 64-bit SPARC kernel with a disk greater than 1 Tbyte.

Error Message

```
Dec  3 09:12:17 holoship scsi: WARNING: /sbus@a,0/SUNW,socal@d,10000/
sf@1,0/ssd@w50020f23000002a4,0 (ssd1):
Dec  3 09:12:17 holoship corrupt label - wrong magic number
```

Cause

You attempted to add this disk to a system running a Solaris release older than the Solaris 9 4/03 release.

Solution

Add this disk to a system running the Solaris 9 4/03 release.

What's New in Disk Management in the Solaris 9 Release?

This section describes new disk management features in the Solaris 9 release.

Solaris Volume Manager and Soft Partitioning

The previously unbundled Solstice DiskSuite™ product is now part of the Solaris 9 release and is called Solaris Volume Manager. Solaris Volume Manager's new partitioning feature, *soft partitioning*, enables more than eight partitions per disk.

For general information about Solaris Volume Manager, see "Storage Management Concepts" in *Solaris Volume Manager Administration Guide*. For information on soft partitioning, see "Soft Partitions (Overview)" in *Solaris Volume Manager Administration Guide*.

Where to Find Disk Management Tasks

Use these references to find step-by-step instructions for managing disks.

Disk Management Task	For More Information
Format a disk and examine a disk label	Chapter 33
Add a new disk to a SPARC system	Chapter 34
Add a new disk to an x86 system	Chapter 35
Hot-Plug a SCSI or PCI disk	Chapter 28

Overview of Disk Management

The management of disks in the Solaris environment usually involves setting up the system and running the Solaris installation program to create the appropriate disk slices and file systems and to install the operating system. Occasionally, you might need to use the `format` utility to add a new disk drive or replace a defective one.

Note – The Solaris operating environment runs on two types of hardware, or platforms—SPARC and x86. The Solaris operating environment runs on both 64–bit and 32–bit address spaces. The information in this document pertains to both platforms and address spaces unless called out in a special chapter, section, note, bullet, figure, table, example, or code example.

Disk Terminology

Before you can effectively use the information in this section, you should be familiar with basic disk architecture. In particular, you should be familiar with the following terms:

Disk Term	Description
Track	A concentric ring on a disk that passes under a single stationary disk head as the disk rotates.
Cylinder	The set of tracks with the same nominal distance from the axis about which the disk rotates.
Sector	A data storage area on a disk, or sections of each disk platter. A sector holds 512 bytes.
Disk controller	A chip and its associated circuitry that controls the disk drive.
Disk label	The first sector of a disk that contains disk geometry and partition information.
Device driver	A device driver is a kernel module that controls a hardware or virtual device.

For additional information, see the product information from your disk's manufacturer.

About Disk Slices

Files stored on a disk are contained in file systems. Each file system on a disk is assigned to a *slice*, which is a group of sectors set aside for use by that file system. Each disk slice appears to the operating system (and to the system administrator) as though it were a separate disk drive.

For information about file systems, see Chapter 38.

Note – Slices are sometimes referred to as partitions. This book uses *slice* but certain interfaces, such as the `format` utility, refer to slices as partitions.

When setting up slices, remember these rules:

- Each disk slice holds only one file system.
- No file system can span multiple slices.

Slices are set up slightly differently on SPARC and x86 platforms. The following table summarizes the differences.

TABLE 32–1 Slice Differences on Platforms

SPARC Platform	x86 Platform
Whole disk is devoted to Solaris environment.	Disk is divided into `fdisk` partitions, one `fdisk` partition per operating environment.
VTOC – Disk is divided into 8 slices, numbered 0–7.	**VTOC** – The Solaris `fdisk` partition is divided into 10 slices, numbered 0–9.
EFI – Disk is divided into 7 slices, numbered 0–6.	

SPARC: Disk Slices

The following table describes the slices on a SPARC based system.

TABLE 32–2 SPARC: Customary Disk Slices

Slice	File System	Usually Found on Client or Server Systems?	Comments
0	root (`/`)	Both	Holds files and directories that make up the operating system. **EFI** – You cannot boot from a disk with an EFI label.
1	swap	Both	Provides virtual memory, or *swap space*.
2	—	Both	**VTOC** – Refers to the entire disk, by convention. The size of this slice should not be changed. **EFI** – Optional slice to be defined based on your site's needs.
3	`/export`	Both	Optional slice that can be defined based on your site's needs. Can be used on a server to hold alternative versions of operating systems that are required by client systems.
4		Both	Optional slice to be defined based on your site's needs.

TABLE 32–2 SPARC: Customary Disk Slices *(Continued)*

Slice	File System	Usually Found on Client or Server Systems?	Comments
5		Both	Optional slice to be defined based on your site's needs. Can be used to hold application software added to a system. If a slice is not allocated for the `/opt` file system during installation, the `/opt` directory is put in slice 0.
6	`/usr`	Both	Holds operating system commands (also known as *executables*). This slice also holds documentation, system programs (`init` and `syslogd`, for example) and library routines.
7	`/home` or `/export/home`	Both	**VTOC** – Holds files that are created by users. **EFI** – Not applicable.
8	N/A	N/A	**VTOC** – Not applicable **EFI** – A reserved slice created by default.This area is similar to the VTOC's alternate cylinders. Do not modify nor delete this slice.

x86: Disk Slices

On x86 based systems, disks are divided into `fdisk` partitions. An `fdisk` partition is a section of the disk that reserved for a particular operating environment, such as the Solaris release.

The Solaris release places ten slices, numbered 0–9, on a Solaris `fdisk` partition as shown in the following table.

TABLE 32–3 x86: Customary Disk Slices

Slice	File System	Usually Found on Client or Server Systems?	Purpose
0	root (/)	Both	Holds the files and directories that make up the operating system.
1	swap	Both	Provides virtual memory, or *swap space*.

TABLE 32–3 x86: Customary Disk Slices *(Continued)*

Slice	File System	Usually Found on Client or Server Systems?	Purpose
2	—	Both	Refers to the entire disk, by convention. The size of this slice should not be changed.
3	`/export`	Both	Optional slice to be defined based on your site's needs. Can be used on a server to hold alternative versions of operating systems that are required by client systems.
4			Optional slice to be defined based on your site's needs.
5		Both	Optional slice to be defined based on your site's needs. Can be used to hold application software added to a system. If a slice is not allocated for the `/opt` file system during installation, the `/opt` directory is put in slice 0.
6	`/usr`	Both	Holds operating system commands (also known as *executables*). This slice also holds documentation, system programs (`init` and `syslogd`, for example) and library routines.
7	`/home` or `/export/home`	Both	Holds files that are created by users.
8	—	Both	Contains information necessary for to boot the Solaris environment from the hard disk. The slice resides at the beginning of the Solaris `fdisk` partition (although the slice number itself does not indicate this fact), and is known as the boot slice.
9	—	Both	Provides an area that is reserved for alternate disk blocks. Slice 9 is known as the alternate sector slice.

Using Raw Data Slices

The SunOS operating system stores the disk label in block 0 of each disk. So, third-party database applications that create raw data slices must not start at block 0, or the disk label will be overwritten and the data on the disk will be inaccessible.

Do not use the following areas of the disk for raw data slices, which are sometimes created by third-party database applications:

- Block 0 where the disk label is stored
- Slice 2, which represents the entire disk with a VTOC label

Slice Arrangements on Multiple Disks

Although a single large disk can hold all slices and their corresponding file systems, two or more disks are often used to hold a system's slices and file systems.

Note – A slice cannot be split between two or more disks. However, multiple swap slices on separate disks are allowed.

For instance, a single disk might hold the root (`/`) file system, a swap area, and the `/usr` file system, while another disk holds the `/export/home` file system and other file systems that contain user data.

In a multiple disk arrangement, the disk that contains the operating system software and swap space (that is, the disk that holds the root (`/`) and `/usr` file systems and the slice for swap space) is called the *system disk*. Other disks are called *secondary disks* or *non-system disks*.

When you arrange a system's file systems on multiple disks, you can modify file systems and slices on the secondary disks without having to shut down the system or reload operating system software.

When you have more than one disk, you also increase input-output (I/O) volume. By distributing disk load across multiple disks, you can avoid I/O bottlenecks.

Determining Which Slices to Use

When you set up a disk's file systems, you choose not only the size of each slice, but also which slices to use. Your decisions about these matters depend on the configuration of the system to which the disk is attached and the software you want to install on the disk.

System configurations that need disk space are as follows:

- Servers
- Standalone systems

Each system configuration can use slices in a different way. The following table lists some examples.

TABLE 32–4 System Configurations and Slices

Slice	Servers	Standalone Systems
0	root	root
1	swap	swap
2	—	—
3	`/export`	—
6	`/usr`	`/usr`
7	`/export/home`	`/home`

For more information about system configurations, see "Overview of System Types" on page 116.

Note – The Solaris installation program provides default slice sizes based on the software you select for installation.

The `format` Utility

Read the following overview of the `format` utility and its uses before proceeding to the "how-to" or reference sections.

The `format` utility is a system administration tool that is used to prepare hard disk drives for use on your Solaris system.

The following table shows the features and associated benefits that the `format` utility provides.

TABLE 32–5 Features and Benefits of the `format` Utility

Feature	Benefit
Searches your system for all attached disk drives	Reports on the following: ■ Target location ■ Disk geometry ■ Whether the disk is formatted ■ If the disk has mounted partitions
Retrieves disk labels	Convenient for repair operations
Repairs defective sectors	Allows administrators to repair disk drives with recoverable errors instead of sending the drive back to the manufacturer
Formats and analyzes a disk	Creates sectors on the disk and verifies each sector
Partitions a disk	Divides a disk into slices so individual file systems can be created on separate slices
Labels a disk	Writes disk name and configuration information to the disk for future retrieval (usually for repair operations)

The `format` utility options are fully described in Chapter 36.

When to Use the `format` Utility

Disk drives are partitioned and labeled by the Solaris installation program when you install the Solaris release. You can use the `format` utility to do the following:

- Display slice information
- Divide a disk into slices
- Add a disk drive to an existing system
- Format a disk drive
- Label a disk
- Repair a disk drive
- Analyze a disk for errors

The main reason a system administrator uses the `format` utility is to divide a disk into disk slices. These steps are covered in Chapter 34 and Chapter 35.

See the following section for guidelines on using the `format` utility.

Guidelines for Using the `format` Utility

TABLE 32–6 The `format` Utility Guidelines

Task	Guidelines	For More Information
Format a disk	■ Any existing data is destroyed when you reformat a disk. ■ The need for formatting a disk drive has dropped as more and more manufacturers ship their disk drives formatted and partitioned. You might not need to use the `format` utility when you add a disk drive to an existing system. ■ If a disk has been relocated and is displaying a lot of disk errors, you can attempt to reformat it, which will automatically remap any bad sectors.	"How to Format a Disk" on page 439
Replace a system disk	■ Data from the damaged system disk must be restored from a backup medium. Otherwise, the system will have to be reinstalled by using the installation program.	"SPARC: How to Connect a System Disk and Boot" on page 459 or "x86: How to Connect a System Disk and Boot" on page 468 or if the system must be reinstalled, *Solaris 9 Installation Guide*
Divide a disk into slices	■ Any existing data is destroyed when you repartition and relabel a disk with existing slices. ■ Existing data must be copied to backup media before the disk is repartitioned and restored.	"SPARC: How to Create Disk Slices and Label a Disk" on page 460 or "x86: How to Create Disk Slices and Label a Disk" on page 476
Add a secondary disk to an existing system	■ Any existing data must be restored from backup media if the secondary disk is reformatted or repartitioned.	"SPARC: How to Connect a Secondary Disk and Boot" on page 459 or "x86: How to Connect a Secondary Disk and Boot" on page 469

TABLE 32–6 The `format` Utility Guidelines *(Continued)*

Task	Guidelines	For More Information
Repair a disk drive	■ Some customer sites prefer to replace rather than repair defective drives. If your site has a repair contract with the disk drive manufacturer, you might not need to use the `format` utility to repair disk drives. ■ The repair of a disk drive usually means that a bad sector is added to a defect list. New controllers remap bad sectors automatically with no system interruption. ■ If the system has an older controller, you might need to remap a bad sector and restore any lost data.	"Repairing a Defective Sector" on page 452

Formatting a Disk

In most cases, disks are formatted by the manufacturer or reseller. So, they do not need to be reformatted when you install the drive. To determine if a disk is formatted, use the `format` utility. For more information, see "How to Determine if a Disk is Formatted" on page 439.

If you determine that a disk is not formatted, use the `format` utility to format the disk.

When you format a disk, you accomplishes two steps:

- The disk media is prepared for use
- A list of disk defects based on a surface analysis is compiled

Caution – Formatting a disk is a destructive process because it overwrites data on the disk. For this reason, disks are usually formatted only by the manufacturer or reseller. If you think disk defects are the cause of recurring problems, you can use the `format` utility to do a surface analysis. However, be careful to use only the commands that do not destroy data. For details, see "How to Format a Disk" on page 439.

A small percentage of total disk space that is available for data is used to store defect and formatting information. This percentage varies according to disk geometry, and decreases as the disk ages and develops more defects.

Formatting a disk might take anywhere from a few minutes to several hours, depending on the type and size of the disk.

About Disk Labels

A special area of every disk is set aside for storing information about the disk's controller, geometry, and slices. That information is called the disk's *label*. Another term that is used to described the disk label is the VTOC (Volume Table of Contents) on a disk with a VTOC label. To *label* a disk means to write slice information onto the disk. You usually label a disk after you change its slices.

If you fail to label a disk after you create slices, the slices will be unavailable because the operating system has no way of "knowing" about the slices.

Partition Table

An important part of the disk label is the *partition table*, which identifies a disk's slices, the slice boundaries (in cylinders), and the total size of the slices. You can display a disk's partition table by using the `format` utility. The following table describes partition table terminology.

TABLE 32–7 Partition Table Terminology

Partition Term	Value	Description
Number	`0-7`	**VTOC** – Partitions or slices, numbered 0–7. **EFI** – Partitions or slices, numbered 0–6.
Tag	`0=UNASSIGNED 1=BOOT 2=ROOT 3=SWAP 4=USR 5=BACKUP 7=VAR 8=HOME 11=RESERVED`	A numeric value that usually describes the file system mounted on this partition.
Flags	`wm`	The partition is writable and mountable.
	`wu rm`	The partition is writable and unmountable. This is the default state of partitions that are dedicated for swap areas. (However, the `mount` command does not check the "not mountable" flag.)
	`rm`	The partition is read only and mountable.

Partition flags and tags are assigned by convention and require no maintenance.

For more information on displaying the partition table, see "How to Display Disk Slice Information" on page 441 or "How to Examine a Disk Label" on page 445.

Displaying Partition Table Information

The following is an example of a partition table from a 4.0-Gbyte disk with a VTOC label displayed from the `format` utility:

```
Total disk cylinders available: 8892 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm    1110 - 4687        1.61GB    (0/3578/0) 3381210
  1       swap    wu       0 - 1109      512.00MB    (0/1110/0) 1048950
  2     backup    wm       0 - 8891        4.01GB    (0/8892/0) 8402940
  3 unassigned    wm       0                0         (0/0/0)          0
  4 unassigned    wm       0                0         (0/0/0)          0
  5 unassigned    wm       0                0         (0/0/0)          0
  6 unassigned    wm       0                0         (0/0/0)          0
  7       home    wm    4688 - 8891        1.89GB    (0/4204/0) 3972780
```

The partition table displayed by the `format` utility contains the following information:

Column Name	Description
`Part`	Partition (or slice number). See Table 32–7 for a description of this column.
`Tag`	Partition tag. See Table 32–7 for a description of this column.
`Flags`	Partition flag. See Table 32–7 for a description of this column.
`Cylinders`	The starting and ending cylinder number for the slice.
`Size`	The slice size in Mbytes.
`Blocks`	The total number of cylinders and the total number of sectors per slice in the far right column.
First Sector	**EFI** – The starting block number.
Last Sector	**EFI** – The ending block number.

The following is an example of a EFI disk label displayed by using the `prtvtoc` command.

```
# prtvtoc /dev/rdsk/c3t1d0s0
* /dev/rdsk/c3t1d0s0 partition map
*
* Dimensions:
*     512 bytes/sector
* 2479267840 sectors
* 2479267773 accessible sectors
*
* Flags:
*   1: unmountable
*  10: read-only
```

```
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00          34    262144    262177
       1      3    01      262178    262144    524321
       6      4    00      524322 2478727100 2479251421
       8     11    00  2479251422     16384 2479267805
```

The `prtvtoc` command provides the following information:

Column Name	Description
`Dimensions`	This section describes the physical dimensions of the disk drive.
`Flags`	This section describes the flags listed in the partition table section. For a description of partition flags, see Table 32–7.
`Partition` (or Slice) Table	This section contains the following information:
`Partition`	Partition (or slice number). For a description of this column, see Table 32–7.
`Tag`	Partition tag. For a description of this column, see Table 32–7.
`Flags`	Partition flag. For a description of this column, see Table 32–7.
`First Sector`	The first sector of the slice.
`Sector Count`	The total number of sectors in the slice.
`Last Sector`	The last sector of the slice.
`Mount Directory`	The last mount point directory for the file system.

Dividing a Disk Into Slices

The `format` utility is most often used by system administrators to divide a disk into slices. The steps are as follows:

- Determining which slices are needed
- Determining the size of each slice
- Using the `format` utility to divide the disk into slices
- Labeling the disk with new slice information
- Creating the file system for each slice

The easiest way to divide a disk into slices is to use the `modify` command from the `partition` menu of the `format` utility. The `modify` command allows you to create slices by specifying the size of each slice without having to keep track of the starting cylinder boundaries. The `modify` command also keeps tracks of any disk space that remains in the "free hog" slice.

Using the Free Hog Slice

When you use the `format` utility to change the size of one or more disk slices, you designate a temporary slice that will expand and shrink to accommodate the resizing operations.

This temporary slice donates, or "frees," space when you expand a slice, and receives, or "hogs," the discarded space when you shrink a slice. For this reason, the donor slice is sometimes called the *free hog*.

The free hog slice exists only during installation or when you run the `format` utility. There is no permanent free hog slice during day-to-day operations.

For information on using the free hog slice, see "SPARC: How to Create Disk Slices and Label a Disk" on page 460 or "x86: How to Create Disk Slices and Label a Disk" on page 476.

CHAPTER 33

Administering Disks (Tasks)

This chapter contains disk administration procedures. Many procedures described in this chapter are optional if you are already familiar with how disks are managed on systems running the Solaris release.

For information on the procedures associated with administering disks, see "Administering Disks (Task Map)" on page 435.

For overview information about disk management, see Chapter 32.

Administering Disks (Task Map)

Task	Description	For Instructions
Identify the disks on a system	If you are not sure of the types of disks on a system, use the `format` utility to identify the disk types.	"How to Identify the Disks on a System" on page 436
Format the disk	Determine whether a disk is already formatted by using the `format` utility. In most cases, disks are already formatted. Use the `format` utility if you need to format a disk.	"How to Determine if a Disk is Formatted" on page 439 "How to Format a Disk" on page 439
Display slice information	Display slice information by using the `format` utility.	"How to Display Disk Slice Information" on page 441

Task	Description	For Instructions
Label the disk	Create the disk label by using the `format` utility.	"How to Label a Disk" on page 443
Examine the disk label	Examine the disk label by using the `prtvtoc` command.	"How to Examine a Disk Label" on page 445
Recover a corrupted disk label	You can attempt to recover a disk label that was damaged due to a system or power failure.	"How to Recover a Corrupted Disk Label" on page 447
Create a `format.dat` entry	Create a `format.dat` entry to support a third-party disk.	"How to Create a `format.dat` Entry" on page 450
Automatically configure a SCSI disk	You can automatically configure a SCSI disk with the SCSI-2 specification for disk device mode sense pages even if the specific drive type is not listed in the `/etc/format.dat` file.	"How to Automatically Configure a SCSI Drive" on page 451
Repair a defective disk sector	Identify a defective disk sector by using the `format` utility.	"How to Identify a Defective Sector by Using Surface Analysis" on page 453
If necessary, fix a defective disk sector	Fix a defective disk sector by using the `format` utility.	"How to Repair a Defective Sector" on page 454

Identifying Disks on a System

Use the `format` utility to discover the types of disks that are connected to a system. You can also use the `format` utility to verify that a disk is known to the system. For information on using the `format` utility, see Chapter 36.

▼ How to Identify the Disks on a System

1. **Become superuser or assume an equivalent role.**
2. **Identify the disks that are recognized on the system with the `format` utility.**

   ```
   # format
   ```

   The format utility displays a list of disks that it recognizes under `AVAILABLE DISK SELECTIONS`.

Examples—Identifying the Disks on a System

The following `format` output is from a system with one disk.

```
# format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <ST34321A cyl 8892 alt 2 hd 15 sec 63>
          /pci@1f,0/pci@1,1/ide@3/dad@0,0
Specify disk (enter its number):
```

The `format` output associates a disk's physical and logical device name to the disk's marketing name, which appears in angle brackets <>. See the example below. This method is an easy way to identify which logical device names represent the disks that are connected to your system. For a description of logical and physical device names, see Chapter 30.

The following example uses a wildcard to display the disks that are connected to a second controller.

```
# format /dev/rdsk/c2*
AVAILABLE DISK SELECTIONS:
  0. /dev/rdsk/c2t0d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
     /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@0,0
  1. /dev/rdsk/c2t1d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
     /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@1,0
  2. /dev/rdsk/c2t2d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
     /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@2,0
  3. /dev/rdsk/c2t3d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
     /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@3,0
  4. /dev/rdsk/c2t5d0s0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
     /io-unit@f,e0200000/sbi@0,0/QLGC,isp@2,10000/sd@5,0
Specify disk (enter its number):
```

The following example identifies the disks on a SPARC based system.

```
# format
AVAILABLE DISK SELECTIONS:
  0. c0t3d0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
     /iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,8800000/sd@3,0
Specify disk (enter its number):
```

The `format` output identifies that disk 0 (target 3) is connected to the first SCSI host adapter (`espdma@...`), which is connected to the first SBus device (`sbus@0...`). The output also associates both the physical and logical device name to the disk's marketing name, `SUN2.1G`.

The following example shows how to identify the disks on an x86 based system.

```
# format
AVAILABLE DISK SELECTIONS:
  0. c0d0 <DEFAULT cyl 615 alt 2 hd 64 sec 63>
```

```
       /pci@0,0/pci-ide@7,1/ata@0/cmdk@0,0
  1. c0d1 <DEFAULT cyl 522 alt 2 hd 32 sec 63>
       /pci@0,0/pci-ide@7,1/ata@0/cmdk@1,0
  2. c1d0 <DEFAULT cyl 817 alt 2 hd 256 sec 63>
       /pci@0,0/pci-ide@7,1/ata@1/cmdk@0,0
Specify disk (enter its number):
```

The `format` output identifies that disk 0 is connected to the first PCI host adapter (`pci-ide@7...`), which is connected to the ATA device (`ata...`). The `format` output on an x86 based system does not identify disks by their marketing names.

Where to Go From Here

Check the following table if the `format` utility did not recognize a disk.

Disk Problem	To Solve the Problem
Disk is newly added and you didn't perform a reconfiguration boot	Go to Chapter 34 or Chapter 35.
Disk is a third-party disk	Go to "Creating a `format.dat` Entry" on page 449.
Label was corrupted by a system problem, such as a power failure	Go to "How to Label a Disk" on page 443.
Disk is not properly connected to the system	Connect the disk to the system by using your disk hardware documentation.

Formatting a Disk

Disks are formatted by the manufacturer or reseller. They usually do not need to be reformatted when you install the drive.

A disk must be formatted before you can do the following:

- Write data to it. However, most disks are already formatted.
- Use the Solaris installation program to install the system.

Caution – Formatting a disk is a destructive process because it overwrites data on the disk. For this reason, disks are usually formatted only by the manufacturer or reseller. If you think disk defects are the cause of recurring problems, you can use the `format` utility to do a surface analysis. However, be careful to use only the commands that do not destroy data.

▼ How to Determine if a Disk is Formatted

1. **Become superuser or assume an equivalent role.**
2. **Invoke the `format` utility.**

   ```
   # format
   ```

3. **Type the number of the disk that you want to check from the list displayed on your screen.**

   ```
   Specify disk (enter its number): 0
   ```

4. **Verify that the disk you chose is formatted by noting the following message.**

   ```
   [disk formatted]
   ```

Example—Determining if a Disk Is Formatted

The following example shows that disk `c0t3d0` is formatted.

```
# format
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 0
selecting c0t1d0
[disk formatted]
```

▼ How to Format a Disk

1. **Become superuser or assume an equivalent role.**
2. **Invoke the `format` utility.**

   ```
   # format
   ```

3. **Type the number of the disk that you want to format from the list displayed on your screen.**

```
Specify disk (enter its number): 0
```

Caution – Do not select the system disk. If you format your system disk, you delete the operating system and any data on this disk.

4. **To begin formatting the disk, type `format` at the `format>` prompt. Confirm the command by typing `y`.**

```
format> format
Ready to format.  Formatting cannot be interrupted
and takes 23 minutes (estimated). Continue? yes
```

5. **Verify that the disk format is successful by noting the following messages.**

```
Beginning format. The current time Tue ABC xx xx:xx:xx xxxx

Formatting...
done

Verifying media...
        pass 0 - pattern = 0xc6dec6de
   2035/12/18

        pass 1 - pattern = 0x6db6db6d
   2035/12/18

Total of 0 defective blocks repaired.
```

Example—Formatting a Disk

The following example shows how to format the disk `c0t3d0`.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number):1
Selecting c0t3d0
[disk formatted]
format> format
Ready to format. Formatting cannot be interrupted
and takes 23 minutes (estimated). Continue? yes
Beginning format. The current time is Thu Dec  6 09:54:40 2001
Formatting ...
done
Verifying media...
        pass 0 - pattern = 0xc6dec6de
```

```
   2035/12/18

        pass 1 - pattern = 0x6db6db6d
   2035/12/18

Total of 0 defective blocks repaired.
format>
```

Displaying Disk Slices

You can use the `format` utility to check whether a disk has the appropriate disk slices. If you determine that a disk does not contain the slices you want to use, use the `format` utility to re-create them and label the disk. For information on creating disk slices, see "SPARC: How to Create Disk Slices and Label a Disk" on page 460 or "x86: How to Create Disk Slices and Label a Disk" on page 476.

Note – The `format` utility uses the term *partition* instead of *slice*.

▼ How to Display Disk Slice Information

1. **Become superuser or assume an equivalent role.**
2. **Invoke the `format` utility.**

   ```
   # format
   ```

3. **Type the number of the disk for which you want to display slice information from the list displayed on your screen.**

   ```
   Specify disk (enter its number):1
   ```

4. **Select the partition menu.**

   ```
   format> partition
   ```

5. **Display the slice information for the current disk drive.**

   ```
   partition> print
   ```

6. **Exit the `format` utility.**

   ```
   partition> q
   format> q
   #
   ```

7. **Verify the displayed slice information by identifying specific slice tags and slices.**
   If the screen output shows that no slice sizes are assigned, the disk probably does not have slices.

Examples—Displaying Disk Slice Information

The following example displays slice information for disk with a VTOC label.

```
# format
Searching for disks...done
Specify disk (enter its number):1
Selecting c0t0d0
format> partition
partition> print
Current partition table (original):
Total disk cylinders available: 8892 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm    1110 - 4687        1.61GB    (0/3578/0) 3381210
  1       swap    wu       0 - 1109      512.00MB    (0/1110/0) 1048950
  2     backup    wm       0 - 8891        4.01GB    (0/8892/0) 8402940
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6 unassigned    wm       0               0         (0/0/0)           0
  7       home    wm    4688 - 8891        1.89GB    (0/4204/0) 3972780
partition> q
format> q
#
```

For a detailed description of the slice information in these examples, see Chapter 32.

The following example shows the slice information on a disk with an EFI label.

```
# format
Searching for disks...done
Specify disk (enter its number): 9
selecting c3t1d0
[disk formatted]
format> partition
partition> print
Current partition table (original):
Total disk sectors available: 2479267806 blocks

Part      Tag    Flag     First Sector          Size          Last Sector
  0       root    wm                34       128.00MB           262177
  1       swap    wu            262178       128.00MB           524321
  2 unassigned    wm                 0            0                0
  3 unassigned    wm                 0            0                0
  4 unassigned    wm                 0            0                0
  5 unassigned    wm                 0            0                0
  6        usr    wm            524322         1.15TB           2479251421
partition> q
```

```
format> q
```

Creating and Examining a Disk Label

The labeling of a disk is usually done during system installation or when you are creating new disk slices. You might need to relabel a disk if the disk label becomes corrupted (for example, from a power failure).

The `format` utility attempts to automatically configure any unlabeled SCSI disk. If the `format` utility is able to automatically configure an unlabeled disk, it displays a message like the following:

```
c0t0d1: configured with capacity of 4.00GB
```

Tip – For information on labeling multiple disks with the same disk label, see "Label Multiple Disks by Using the `prtvtoc` and `fmthard` Commands" on page 455.

▼ How to Label a Disk

You can use the following procedure to label at disk with a VTOC label or a disk greater than 1 Tbyte with an EFI label. If you want to put an EFI label on disk smaller than 1 Tbyte, see "Example—Labeling a Disk Less Than 1 Tbyte with an EFI Label" on page 445.

1. **Become superuser or assume an equivalent role.**
2. **Invoke the `format` utility.**

   ```
   # format
   ```

3. **Type the number of the disk that you want to label from the list displayed on your screen.**

   ```
   Specify disk (enter its number):1
   ```

4. **Select one of the following.**

   a. **If the disk is unlabeled and was successfully configured, go to step 5 to label the disk.**

      The `format` utility will ask if you want to label the disk.

   b. **If the disk is labeled and you want to change the disk type, or if the `format` utility was not able to automatically configure the disk, follow steps 6-7 to set the disk type and label the disk.**

5. **Label the disk by typing `y` at the `Label it now?` prompt.**

   ```
   Disk not labeled. Label it now? y
   ```

   The disk is now labeled. Go to step 10 to exit the `format` utility.

6. **Enter `type` at the `format>` prompt.**

   ```
   format> type
   ```

   The Available Drive Types menu is displayed.

7. **Select a disk type from the list of possible disk types.**

   ```
   Specify disk type (enter its number)[12]: 12
   ```

   Or, select `0` to automatically configure a SCSI-2 disk. For more information, see "How to Automatically Configure a SCSI Drive" on page 451.

8. **Label the disk. If the disk is not labeled, the following message is displayed.**

   ```
   Disk not labeled. Label it now? y
   ```

   Otherwise, you are prompted with this message:

   ```
   Ready to label disk, continue? y
   ```

9. **Verify the disk label.**

   ```
   format> verify
   ```

10. **Exit the `format` utility.**

    ```
    partition> q
    format> q
    #
    ```

Example—Labeling a Disk

The following example shows how to automatically configure and label a 1.05-Gbyte disk.

```
# format
    c1t0d0: configured with capacity of 1002.09MB

AVAILABLE DISK SELECTIONS:
       0. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
       1. c1t0d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
Specify disk (enter its number): 1
Disk not labeled.  Label it now?  yes
format> verify
#
```

Example—Labeling a Disk Less Than 1 Tbyte with an EFI Label

The following example shows how to use the `format -e` command to label a disk less than 1 Tbyte with an EFI label. Remember to verify that your layered software products will continue to work on systems with EFI-labeled disks. For general information on EFI label restrictions, see "Restrictions of the EFI Disk Label" on page 416.

```
# format -e
Searching for disks...done
AVAILABLE DISK SELECTIONS:
       1. c1t0d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@0,0
       2. c1t1d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@1,0
       3. c1t8d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@8,0
       4. c1t9d0 <SUNW18g cyl 7506 alt 2 hd 19 sec 248>
          /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 4
selecting c1t9d0
[disk formatted]
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[0]: 1
Ready to label disk, continue? yes
format> quit
```

▼ How to Examine a Disk Label

Examine disk label information by using the `prtvtoc` command. For a detailed description of the disk label and the information that is displayed by the `prtvtoc` command, see Chapter 32.

1. **Become superuser or assume an equivalent role.**
2. **Display the disk label information.**

   # **prtvtoc /dev/rdsk/***device-name*

   device-name is the raw disk device you want to examine.

Examples—Examining a Disk Label

The following example shows the disk label information for disk with a VTOC label.

```
# prtvtoc /dev/rdsk/c0t0d0s0
* /dev/rdsk/c0t0d0s0 partition map
*
```

```
* Dimensions:
*     512 bytes/sector
*      63 sectors/track
*      15 tracks/cylinder
*     945 sectors/cylinder
*    8894 cylinders
*    8892 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00    1048950   3381210   4430159   /
       1      3    01          0   1048950   1048949
       2      5    00          0   8402940   8402939
       7      8    00    4430160   3972780   8402939   /export/home
```

The following example shows the disk label information for disk with an EFI label.

```
# prtvtoc /dev/rdsk/c3t1d0s0
* /dev/rdsk/c3t1d0s0 partition map
*
* Dimensions:
*     512 bytes/sector
* 2479267840 sectors
* 2479267773 accessible sectors
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                          First     Sector    Last
* Partition  Tag  Flags    Sector     Count    Sector  Mount Directory
       0      2    00         34    262144    262177
       1      3    01     262178    262144    524321
       6      4    00     524322 2478727100 2479251421
       8     11    00  2479251422     16384 2479267805
```

Recovering a Corrupted Disk Label

Sometimes, a power or system failure causes a disk's label to become unrecognizable. A corrupted disk label doesn't always mean that the slice information or the disk's data must be recreated or restored.

The first step to recovering a corrupted disk label is to label the disk with the correct geometry and disk type information. You can complete this step through the normal disk labeling method, by using either automatic configuration or manual disk type specification.

If the `format` utility recognizes the disk type, the next step is to search for a backup label to label the disk. Labeling the disk with the backup label labels the disk with the correct partitioning information, the disk type, and disk geometry.

▼ How to Recover a Corrupted Disk Label

1. **Boot the system to single-user mode.**

   If necessary, boot the system from a local CD-ROM or the network in single-user mode to access the disk.

   See Chapter 13 or Chapter 14 for information on booting the system.

2. **Relabel the disk.**

   ```
   # format
   ```

   At this point, the `format` utility attempts to automatically configure any unlabeled SCSI disk. If the `format` utility is able to configure the unlabeled and corrupted disk, it will display:

   c`w`*t*`x`*d*`y`: `configured with capacity of` *abc*`MB`

   The `format` utility then displays the list of disks on the system.

3. **Type the number of the disk that you need to recover from the list displayed on your screen.**

   ```
   Specify disk (enter its number): 1
   ```

4. **Select one of the following to determine how to label the disk.**

   a. **If the disk was configured successfully, follow steps 5 and 6. Then go to step 12.**

   b. **If the disk was not configured successfully, follow steps 7-11. Then go to step 12.**

5. **Search for the backup label.**

   ```
   format> verify
   Warning: Could not read primary label.
   Warning: Check the current partitioning and 'label' the disk or
   use the 'backup' command.
   Backup label contents:
   Volume name = <        >
   ascii name  = <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
   pcyl        = 2038
   ncyl        = 2036
   acyl        =    2
   nhead       =   14
   nsect       =   72
   Part      Tag    Flag     Cylinders        Size            Blocks
     0       root    wm       0 -  300      148.15MB    (301/0/0)   303408
     1       swap    wu     301 -  524      110.25MB    (224/0/0)   225792
     2     backup    wm       0 - 2035     1002.09MB    (2036/0/0) 2052288
   ```

```
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6        usr    wm     525 - 2035      743.70MB    (1511/0/0) 1523088
  7 unassigned    wm       0               0         (0/0/0)           0
```

6. **If the `format` utility was able to find a backup label and the backup label contents appear satisfactory, use the `backup` command to label the disk with the backup label.**

```
format> backup
Disk has a primary label, still continue? y

Searching for backup labels...found.
Restoring primary label
```

The disk label has been recovered. Go to step 12.

7. **If the `format` utility was not able to automatically configure the disk, specify the disk type by using the `type` command.**

```
format> type
```

The Available Drives Type menu is displayed.

8. **Select `0` to automatically configure the disk, or select a disk type from the list of possible disk types.**

```
Specify disk type (enter its number)[12]: 12
```

9. **If the disk was successfully configured, reply with `no` when the `format` utility asks if you want to label the disk.**

```
Disk not labeled.  Label it now?  no
```

10. **Use the `verify` command to search for backup labels.**

```
format> verify
Warning: Could not read primary label.
Warning: Check the current partitioning and 'label' the disk
or use the 'backup' command.
.
.
.
```

11. **If the `format` utility was able to find a backup label and the backup label contents appear satisfactory, use the `backup` command to label the disk with the backup label.**

```
format> backup
Disk has a primary label, still continue? y
Searching for backup labels...found.
Restoring primary label
```

The disk label has been recovered.

12. **Exit the `format` utility.**

```
format> q
```

13. **Verify the file systems on the recovered disk by using the `fsck` command.**
    For information on using the `fsck` command, see Chapter 43.

Adding a Third-Party Disk

The Solaris environment supports many third-party disks. However, you might need to supply either a device driver, a `format.dat` entry, or both for the disk to be recognized. Other options for adding disks are as follows:

- If you are adding a SCSI disk, you might to try the `format` utility's automatic configuration feature. For more information, see "Automatically Configuring SCSI Disk Drives" on page 450.
- You might try hot-plugging a PCI, SCSI, or USB disk. For more information, see Chapter 27.

If the third-party disk is designed to work with standard SunOS-compatible device drivers, then creation of an appropriate `format.dat` entry should be enough to allow the disk to be recognized by the `format` utility. In other cases, you need to load a third-party device driver to support the disk.

Note – Sun cannot guarantee that its `format` utility will work properly with all third-party disk drivers. If the disk driver is not compatible with the Solaris `format` utility, the disk drive vendor should supply you with a custom format program.

This section discusses what to do if some of this software support is missing. Typically, you discover that software support is missing when you invoke the `format` utility and find that the disk type is not recognized.

Supply the missing software as described in this section, and then refer to the appropriate configuration procedure for adding system disks or secondary disks in Chapter 34 or Chapter 35.

Creating a `format.dat` Entry

Unrecognized disks cannot be formatted without precise information about the disk's geometry and operating parameters. This information is supplied in the `/etc/format.dat` file.

Note – SCSI-2 drives do not require a `format.dat` entry. The `format` utility automatically configures the SCSI-2 drivers if the drives are powered on during a reconfiguration boot. For step-by-step instructions on configuring a SCSI disk drive automatically, see "How to Automatically Configure a SCSI Drive" on page 451.

If your disk is unrecognized, use a text editor to create an entry in `format.dat` for the disk. You need to gather all the pertinent technical specifications about the disk and its controller before you start. This information should have been provided with the disk. If not, contact the disk manufacturer or your supplier.

▼ How to Create a `format.dat` Entry

1. **Become superuser or assume an equivalent role.**
2. **Make a copy of the `/etc/format.dat` file.**

   ```
   # cp /etc/format.dat /etc/format.dat.gen
   ```

3. **Modify the `/etc/format.dat` file to include an entry for the third-party disk by using the `format.dat` information that is described in Chapter 36.**

   Use the disk's hardware product documentation to gather the required information.

Automatically Configuring SCSI Disk Drives

The `format` utility automatically configures SCSI disk drives even if that specific type of drive is not listed in the `/etc/format.dat` file. This feature enables you to format, create slices for, and label any disk driver that is compliant with the SCSI-2 specification for disk device mode sense pages.

Other options for adding disks are:

- If you are adding a SCSI disk, you might to try the `format` utility's automatic configuration feature. For more information, see "Automatically Configuring SCSI Disk Drives" on page 450.
- You might try hot-plugging a PCI, SCSI, or USB disk. For more information, see Chapter 27.

The following steps are involved in configuring a SCSI drive by using automatic configuration:

- Shutting down the system
- Attaching the SCSI disk drive to the system
- Turning on the disk drive
- Performing a reconfiguration boot
- Using the `format` utility to automatically configure the SCSI disk drive

After the reconfiguration boot, invoke the `format` utility. The `format` utility will attempt to configure the disk and, if successful, alert the user that the disk was configured. For step-by-step instructions on configuring a SCSI disk drive automatically, see "How to Automatically Configure a SCSI Drive" on page 451.

Here's an example of a partition table for a 1.3-Gbyte SCSI disk drive that was displayed by the `format` utility.

```
Part    Tag    Flag     Cylinders     Size        Blocks
   0     root    wm       0 -   96    64.41MB      (97/0/0)
   1     swap    wu      97 -  289   128.16MB     (193/0/0)
   2   backup    wu       0 - 1964     1.27GB    (1965/0/0)
   6      usr    wm     290 - 1964     1.09GB    (1675/0/0)
```

For more information on using SCSI automatic configuration, see Chapter 36.

▼ How to Automatically Configure a SCSI Drive

1. **Become superuser or equivalent role.**
2. **Create the `/reconfigure` file that will be read when the system is booted.**

   ```
   # touch /reconfigure
   ```

3. **Shut down the system.**

   ```
   # shutdown -i0 -gn -y
   ```

`-i`*n*	Brings the system down to init level 0, the power-down state.
`-g30`	Notifies logged-in users that they have *n* seconds before the system begins to shut down.
`-y`	Specifies that the command should run without user intervention.

   The `ok` prompt is displayed after the system is shut down.

4. **Turn off the power to the system and all external peripheral devices.**
5. **Make sure that the disk you are adding has a different target number than the other devices on the system.**

   You will often find a small switch located at the back of the disk for this purpose.

6. **Connect the disk to the system and check the physical connections.**

Refer to the disk's hardware installation guide for installation details.

7. **Turn on the power to all external peripherals.**
8. **Turn on the power to the system.**
   The system boots and displays the login prompt.
9. **Log back in as superuser or assume an equivalent role.**
10. **Invoke the `format` utility and select the disk that you want to configure automatically.**

```
# format
Searching for disks...done
c1t0d0: configured with capacity of 1002.09MB
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 1
```

11. **Type `yes` to the prompt to label the disk.**
    Typing `y` causes the disk label to be generated and written to the disk by SCSI automatic configuration.

```
Disk not labeled. Label it now? y
```

12. **Verify the disk label.**

```
format> verify
```

13. **Exit the `format` utility.**

```
format> q
```

Repairing a Defective Sector

If a disk on your system has a defective sector, you can repair it by following procedures in this section. You might become aware of defective sectors when you do the following:

- Run surface analysis on a disk

  For more information on the analysis feature of the `format` utility, see "The `analyze` Menu" on page 486.

  The defective area reported while your system is running might not be accurate. Since the system does disk operations many sectors at a time, it is often hard to pinpoint exactly which sector caused a given error. To find the exact sector(s), use

"How to Identify a Defective Sector by Using Surface Analysis" on page 453.

- Get multiple error messages from the disk driver concerning a particular portion of the disk while your system is running.

  Messages that are related to disk errors look like the following:

```
WARNING: /io-unit@f,e0200000/sbi@0,0/QLGC,isp@1,10000/sd@3,0 (sd33):
    Error for command 'read' Error Level: Retryable
    Requested Block 126, Error Block: 179
    Sense Key: Media Error
    Vendor 'name':
    ASC = 0x11 (unrecovered read error), ASCQ = 0x0, FRU = 0x0
```

  The preceding console message indicates that block 179 might be defective. Relocate the bad block by using the `format` utility's `repair` command or use the `analyze` command with the repair option enabled.

▼ How to Identify a Defective Sector by Using Surface Analysis

1. **Become superuser or assume an equivalent role.**

2. **Unmount the file system in the slice that contains the defective sector.**

```
# umount /dev/dsk/device-name
```

   For more information, see `mount`(1M).

3. **Invoke the `format` utility.**

```
# format
```

4. **Select the affected disk.**

```
Specify disk (enter its number):1
selecting c0t2d0:
[disk formatted]
Warning: Current Disk has mounted partitions.
```

5. **Select the analyze menu.**

```
format> analyze
```

6. **Set up the analysis parameters by typing `setup` at the `analyze>` prompt.**

   Use the parameters shown here:

```
analyze> setup
Analyze entire disk [yes]? n
Enter starting block number [0, 0/0/0]: 12330
Enter ending block number [2052287, 2035/13/71]: 12360
Loop continuously [no]? y
Repair defective blocks [yes]? n
```

```
Stop after first error [no]? n
Use random bit patterns [no]? n
Enter number of blocks per transfer [126, 0/1/54]: 1
Verify media after formatting [yes]? y
Enable extended messages [no]? n
Restore defect list [yes]? y
Create defect label [yes]? y
```

7. **Use the `read` command to find the defect.**

```
analyze> read
Ready to analyze (won't harm SunOS). This takes a long time,
but is interruptible with Control-C. Continue? y
        pass 0
   2035/12/1825/7/24
        pass 1
Block 12354  (18/4/18), Corrected media error (hard data ecc)
   25/7/24
^C
Total of 1 defective blocks repaired.
```

▼ How to Repair a Defective Sector

1. **Become superuser or assume an equivalent role.**
2. **Invoke the `format` utility.**

```
# format
```

3. **Select the disk that contains the defective sector.**

```
Specify disk (enter its number): 1
selecting c0t3d0
[disk formatted]
format>
```

4. **Select the `repair` command.**

```
format> repair
```

5. **Type the defective block number.**

```
Enter absolute block number of defect: 12354
   Ready to repair defect, continue? y
   Repairing block 12354 (18/4/18)...ok.
format>
```

If you are unsure of the format that is used to identify the defective sector, see "How to Identify a Defective Sector by Using Surface Analysis" on page 453 for more information.

Tips and Tricks for Managing Disks

Use the following tips to help you manage disks more efficiently.

Debugging `format` Sessions

Invoke `format -M` to enable extended and diagnostic messages for ATA and SCSI devices.

In this example, the series of numbers under `Inquiry:` represent the hexadecimal value of the `inquiry` data that is displayed to the right of the numbers.

```
# format -M
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
  1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
     /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0

Specify disk (enter its number): 0
selecting c0t3d0
[disk formatted]
format> inquiry
Inquiry:
00 00 02 02 8f 00 00 12 53 45 41 47 41 54 45 20     ........NAME....
53 54 31 31 32 30 30 4e 20 53 55 4e 31 2e 30 35     ST11200N SUN1.05
38 33 35 38 30 30 30 33 30 32 30 39 00 00 00 00     835800030209....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00     ................
00 43 6f 70 79 72 69 67 68 74 20 28 63 29 20 31     .Copyright (c) 1
39 39 32 20 53 65 61 67 61 74 65 20 41 6c 6c 20     992 NAME    All
72 69 67 68 74 73 20 72 65 73 65 72 76 65 64 20     rights reserved
30 30 30                                            000
Vendor:   name
Product:  ST11200N SUN1.05
Revision: 8358
format>
```

Label Multiple Disks by Using the `prtvtoc` and `fmthard` Commands

Use the `prtvtoc` and `fmthard` commands to label multiple disks with the same disk geometry.

Use the following `for loop` in a script to copy a disk label from one disk and replicate it on multiple disks.

```
# for i in xyz
> do
> prtvtoc /dev/rdsk/cwtxdysz | fmthard -s - /dev/rdsk/cwt${i}d0s2
> done
```

Example—Labeling Multiple Disks

In this example, the disk label from `c2t0d0s0` is copied to four other disks.

```
# for i in 1 2 3 5
> do
> prtvtoc /dev/rdsk/c2t0d0s0 | fmthard -s - /dev/rdsk/c2t${i}d0s2
> done
fmthard:  New volume table of contents now in place.
fmthard:  New volume table of contents now in place.
fmthard:  New volume table of contents now in place.
fmthard:  New volume table of contents now in place.
#
```

CHAPTER 34

SPARC: Adding a Disk (Tasks)

This chapter describes how to add a disk to a SPARC based system.

For information on the procedures associated with adding a disk to a SPARC based system, see "SPARC: Adding a System Disk or a Secondary Disk (Task Map)" on page 457.

For overview information about disk management, see Chapter 32. For step-by-step instructions on adding a disk to an x86 based system, see Chapter 35.

SPARC: Adding a System Disk or a Secondary Disk (Task Map)

The following task map identifies the procedures for adding a disk to a SPARC based system.

Task	Description	For Instructions
1. Connect the disk and boot	*System Disk* Connect the new disk and boot from a local or remote Solaris CD or DVD.	"SPARC: How to Connect a System Disk and Boot" on page 459

Task	Description	For Instructions
	Secondary Disk Connect the new disk and perform a reconfiguration boot so that the system will recognize the disk.	"SPARC: How to Connect a Secondary Disk and Boot" on page 459
2. Create slices and label the disk	Create disk slices and label the disk if the disk manufacturer has not already done so.	"SPARC: How to Create Disk Slices and Label a Disk" on page 460
3. Create file systems	Create UFS file systems on the disk slices with the `newfs` command. You must create the root (`/`) or `/usr` file system, or both, for a system disk.	"SPARC: How to Create File Systems" on page 465
4. Restore file systems	Restore the root (`/`) or `/usr` file system, or both, on the system disk. If necessary, restore file systems on the secondary disk.	Chapter 49
5. Install boot block	*System Disk Only.* Install the boot block on the root (`/`) file system, so that the system can boot.	"SPARC: How to Install a Boot Block on a System Disk" on page 466

SPARC: Adding a System Disk or a Secondary Disk

A system disk contains the root (`/`) or `/usr` file systems, or both. If the disk that contains either of these file systems becomes damaged, you have two ways to recover:

- You can reinstall the entire Solaris environment.
- Or, you can replace the system disk and restore your file systems from a backup medium.

A secondary disk does not contain the root (`/`) and `/usr` file systems. A secondary disk usually contains space for user files. You can add a secondary disk to a system for more disk space, or you can replace a damaged secondary disk. If you replace a secondary disk on a system, you can restore the old disk's data on the new disk.

▼ SPARC: How to Connect a System Disk and Boot

This procedure assumes that the system is shut down.

1. **Disconnect the damaged system disk from the system.**
2. **Make sure that the disk you are adding has a different target number than the other devices on the system.**

   You will often find a small switch located at the back of the disk for this purpose.
3. **Connect the replacement system disk to the system and check the physical connections.**

   Refer to the disk's hardware installation guide for installation details.
4. **Follow the instructions in the following table, depending on whether you are booting from a local Solaris CD or DVD or a remote Solaris CD or DVD from the network.**

Boot Type	Action
From a Solaris CD or DVD in a local drive	1. Make sure the CD or DVD is in the drive. 2. Boot from the media to single-user mode: `ok boot cdrom -s`
From the network	Boot from the network to single-user mode: `ok boot net -s`

After a few minutes, the root prompt (#) is displayed.

Where to Go From Here

After you boot the system, you can create slices and a disk label on the disk. Go to "SPARC: How to Create Disk Slices and Label a Disk" on page 460.

▼ SPARC: How to Connect a Secondary Disk and Boot

1. **Become superuser or assume an equivalent role.**
2. **If the disk type is unsupported by the Solaris software, add the device driver for the disk by following the instructions included with the hardware.**

   For information on creating a `format.dat` entry for the disk, see "How to Create a `format.dat` Entry" on page 450, if necessary.

3. **Create the `/reconfigure` file that will be read when the system is booted.**

   ```
   # touch /reconfigure
   ```

   The `/reconfigure` file causes the SunOS software to check for the presence of any newly installed peripheral devices when you power on or boot your system later.

4. **Shut down the system.**

   ```
   # shutdown -i0 -gn -y
   ```

`-i0`	Changes to run level 0, the power-down state.
`-gn`	Notifies logged-in users that they have *n* seconds before the system begins to shut down.
`-y`	Specifies that the command should run without user intervention.

   The `ok` prompt is displayed after the operating environment is shut down.

5. **Turn off the power to the system and all external peripheral devices.**

6. **Make sure that the disk you are adding has a different target number than the other devices on the system.**

   You will often find a small switch located at the back of the disk for this purpose.

7. **Connect the disk to the system and check the physical connections.**

   Refer to the disk's hardware installation guide for installation details.

8. **Turn on the power to all external peripherals.**

9. **Turn on the power to the system.**

   The system boots and displays the login prompt.

Where to Go From Here

After you boot the system, you can create slices and a disk label on the disk. Go to "SPARC: How to Create Disk Slices and Label a Disk" on page 460.

▼ SPARC: How to Create Disk Slices and Label a Disk

1. **Become superuser or assume an equivalent role.**

2. **Invoke the `format` utility.**

   ```
   # format
   ```

A list of available disks is displayed. For more information, see `format`(1M).

3. **Type the number of the disk that you want to repartition from the list displayed on your screen.**

   ```
   Specify disk (enter its number): disk-number
   ```

   disk-number is the number of the disk that you want to repartition.

4. **Select the `partition` menu.**

   ```
   format> partition
   ```

5. **Display the current partition (slice) table.**

   ```
   partition> print
   ```

6. **Start the modification process.**

   ```
   partition> modify
   ```

7. **Set the disk to all free hog.**

   ```
   Choose base (enter number) [0]? 1
   ```

   For more information about the free hog slice, see "Using the Free Hog Slice" on page 433.

8. **Create a new partition table by answering `y` when prompted to continue.**

   ```
   Do you wish to continue creating a new partition table based on
   above table[yes]? y
   ```

9. **Identify the free hog partition (slice) and the sizes of the slices when prompted.**

   When adding a system disk, you must set up slices for:

   - root (slice 0) and swap (slice 1)
   - `/usr` (slice 6)

   After you identify the slices, the new partition table is displayed.

   For an example of creating disk slices, see "SPARC: Example—Creating Disk Slices and Labeling a System Disk" on page 462.

10. **Make the displayed partition table the current partition table by answering `y` when asked.**

    ```
    Okay to make this the current partition table[yes]? y
    ```

    If you do not want the current partition table and you want to change it, answer `no` and go to Step 6.

11. **Name the partition table.**

    ```
    Enter table name (remember quotes): "partition-name"
    ```

    partition-name is the name for the new partition table.

12. **Label the disk with the new partition table after you have finished allocating slices on the new disk.**

```
Ready to label disk, continue? yes
```

13. **Quit the `partition` menu.**

```
partition> q
```

14. **Verify the disk label.**

```
format> verify
```

15. **Exit the `format` menu.**

```
format> q
```

SPARC: Example—Creating Disk Slices and Labeling a System Disk

The following example shows the `format` utility being used to divide a 1-Gbyte disk into three slices: one slice for the root (`/`) file system, one slice for the swap area, and one slice for the `/usr` file system.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
   0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
   1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 0
selecting c0t1d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
    0. Current partition table (original)
    1. All Free Hog
Choose base (enter number) [0]? 1

Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0               0         (0/0/0)           0
  1       swap    wu       0               0         (0/0/0)           0
  2     backup    wu       0 - 2035     1002.09MB    (2036/0/0) 2052288
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6        usr    wm       0               0         (0/0/0)           0
  7 unassigned    wm       0               0         (0/0/0)           0
Do you wish to continue creating a new partition
table based on above table[yes]? yes
```

```
Free Hog partition[6]? 6
Enter size of partition '0' [0b, 0c, 0.00mb]: 200mb
Enter size of partition '1' [0b, 0c, 0.00mb]: 200mb
Enter size of partition '3' [0b, 0c, 0.00mb]:
Enter size of partition '4' [0b, 0c, 0.00mb]:
Enter size of partition '6' [0b, 0c, 0.00mb]:
Enter size of partition '7' [0b, 0c, 0.00mb]:

 Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0 -  406      200.32MB    (407/0/0)   410256
  1       swap    wu     407 -  813      200.32MB    (407/0/0)   410256
  2     backup    wu       0 - 2035     1002.09MB    (2036/0/0) 2052288
  3 unassigned    wm       0               0         (0/0/0)          0
  4 unassigned    wm       0               0         (0/0/0)          0
  5 unassigned    wm       0               0         (0/0/0)          0
  6        usr    wm     814 - 2035      601.45MB    (1222/0/0) 1231776
  7 unassigned    wm       0               0         (0/0/0)          0

Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "disk0"
Ready to label disk, continue? yes
partition> quit
format> verify
format> quit
```

SPARC: Examples—Creating Disk Slices and Labeling a Secondary Disk

The following example shows the `format` utility being used to divide a 1-Gbyte disk into one slice for the `/export/home` file system.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
   0. c0t1d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@1,0
   1. c0t3d0 <SUN1.05 cyl 2036 alt 2 hd 14 sec 72>
      /iommu@f,e0000000/sbus@f,e0001000/espdma@f,400000/esp@f,800000/sd@3,0
Specify disk (enter its number): 0
selecting c0t1d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
    0. Current partition table (original)
    1. All Free Hog
Choose base (enter number) [0]? 1

Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0               0         (0/0/0)          0
  1       swap    wu       0               0         (0/0/0)          0
```

```
  2     backup    wu       0 - 2035     1002.09MB    (2036/0/0) 2052288
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6        usr    wm       0               0         (0/0/0)           0
  7 unassigned    wm       0               0         (0/0/0)           0
Do you wish to continue creating a new partition
table based on above table[yes]? y
Free Hog partition[6]? 7
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:
 Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0               0         (0/0/0)           0
  1       swap    wu       0               0         (0/0/0)           0
  2     backup    wu       0 - 2035     1002.09MB    (2036/0/0) 2052288
  3 unassigned    wm       0               0         (0/0/0)           0
  4 unassigned    wm       0               0         (0/0/0)           0
  5 unassigned    wm       0               0         (0/0/0)           0
  6        usr    wm       0               0         (0/0/0)           0
  7 unassigned    wm       0 - 2035     1002.09MB    (2036/0/0) 2052288
Okay to make this the current partition table[yes]? yes
Enter table name (remember quotes): "home"
Ready to label disk, continue? y
partition> q
format> verify
format> q
#
```

The following example shows how to use the format utility to divide a 1.15 Tbyte disk with an EFI label into 3 slices.

```
# format
.
.
.
partition> modify
Select partitioning base:
        0. Current partition table (original)
        1. All Free Hog
Choose base (enter number) [0]? 1

Part      Tag    Flag     First Sector          Size          Last Sector
  0 unassigned    wm                 0             0                0
  1 unassigned    wm                 0             0                0
  2 unassigned    wm                 0             0                0
  3 unassigned    wm                 0             0                0
  4 unassigned    wm                 0             0                0
  5 unassigned    wm                 0             0                0
  6        usr    wm                 0             0                0

Do you wish to continue creating a new partition
```

```
table based on above table[yes]? y
Free Hog partition[6]? 4
Enter size of partition 0 [0b, 34e, 0mb, 0gb, 0tb]:
Enter size of partition 1 [0b, 34e, 0mb, 0gb, 0tb]:
Enter size of partition 2 [0b, 34e, 0mb, 0gb, 0tb]: 400gb
Enter size of partition 3 [0b, 838860834e, 0mb, 0gb, 0tb]: 400gb
Enter size of partition 5 [0b, 1677721634e, 0mb, 0gb, 0tb]:
Enter size of partition 6 [0b, 1677721634e, 0mb, 0gb, 0tb]:
Part      Tag    Flag     First Sector          Size          Last Sector
  0 unassigned    wm                 0            0               0
  1 unassigned    wm                 0            0               0
  2        usr    wm                34       400.00GB          838860833
  3        usr    wm         838860834       400.00GB         1677721633
  4        usr    wm        1677721634       382.20GB         2479251421
  5 unassigned    wm                 0            0               0
  6 unassigned    wm                 0            0               0

Ready to label disk, continue? yes

partition> q
```

Where to Go From Here

After you create disk slices and label the disk, you can create file systems on the disk. Go to "SPARC: How to Create File Systems" on page 465.

▼ SPARC: How to Create File Systems

1. **Become superuser or assume an equivalent role.**
2. **Create a file system for each slice.**

   ```
   # newfs /dev/rdsk/cwtxdysz
   ```

   /dev/rdsk/*cwtxdysx* is the raw device for the file system to be created.

   For more information about the `newfs` command, see Chapter 39 or `newfs`(1M).

3. **Verify the new file system by mounting.**

   ```
   # mount /dev/dsk/cwtxdysz /mnt
   # ls lost+found
   ```

SPARC: Where to Go From Here

Add Disk Task	Action
System Disk	You need to restore the root (/) and /usr file systems on the disk. Go to Chapter 49.
	After the root (/) and /usr file systems are restored, install the boot block. Go to "SPARC: How to Install a Boot Block on a System Disk" on page 466.
Secondary Disk	You might need to restore file systems on the new disk. Go to Chapter 49.
	If you are not restoring file systems on the new disk, you are finished adding a secondary disk. See Chapter 40 for information on making the file systems available to users.

▼ SPARC: How to Install a Boot Block on a System Disk

1. **Become superuser or assume an equivalent role.**
2. **Install a boot block on the system disk.**

```
# installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk /dev/rdsk/cwtxdys0
```

/usr/platform/`uname -i`/lib/fs/ufs/bootblk	Is the boot block code.
/dev/rdsk/*cwtxdy*s0	Is the raw device of the root (/) file system.

For more information, see installboot(1M).

3. **Verify that the boot blocks are installed by rebooting the system to run level 3.**

```
# init 6
```

SPARC: Example—Installing a Boot Block on a System Disk

The following example shows how to install the boot block on a SPARCstation 10.

```
# installboot /usr/platform/sun4m/lib/fs/ufs/bootblk /dev/rdsk/c0t0d0s0
```

CHAPTER 35

x86: Adding a Disk (Tasks)

This chapter describes how to add a disk to an x86 based system.

For information on the procedures associated with adding a disk to an x86 based system, see "x86: Adding a System Disk or a Secondary Disk (Task Map)" on page 467.

For overview information about disk management, see Chapter 32. For step-by-step instructions on adding a disk to a SPARC based system, see Chapter 34.

x86: Adding a System Disk or a Secondary Disk (Task Map)

Task	Description	For Instructions
1. Connect the disk and boot	*System Disk* Connect the new disk and boot from a local or remote Solaris CD or DVD. *Secondary Disk* Connect the new disk and perform a reconfiguration boot, so that the system will recognize the disk.	"x86: How to Connect a System Disk and Boot" on page 468 "x86: How to Connect a Secondary Disk and Boot" on page 469

Task	Description	For Instructions
2. Create slices and label the disk	Create disk slices and label the disk if the disk manufacturer has not already done so.	"x86: How to Create a Solaris `fdisk` Partition" on page 471 and "x86: How to Create Disk Slices and Label a Disk" on page 476
3. Create File Systems	Create UFS file systems on the disk slices with the `newfs` command. You must create the root (`/`) or `/usr` file system (or both) for a system disk.	"x86: How to Create File Systems" on page 478
4. Restore File Systems	Restore the root (`/`) or `/usr` file system (or both) on the system disk. If necessary, restore file systems on the secondary disk.	Chapter 49
5. Install Boot Block	*System Disk Only.* Install the boot block on the root (`/`) file system so that the system can boot.	"x86: How to Install a Boot Block on a System Disk" on page 479

x86: Adding a System or Secondary Disk

A system disk contains the root (`/`) or `/usr` file systems, or both. If the disk that contains either of these file systems becomes damaged, you have two ways to recover:

- You can reinstall the entire Solaris environment.
- Or, you can replace the system disk and restore your file systems from a backup medium.

A secondary disk doesn't contain the root (`/`) and `/usr` file systems. A secondary disk usually contains space for user files. You can add a secondary disk to a system for more disk space, or you can replace a damaged secondary disk. If you replace a secondary disk on a system, you can restore the old disk's data on the new disk.

▼ x86: How to Connect a System Disk and Boot

This procedure assumes that the system is down.

1. **Disconnect the damaged system disk from the system.**

2. **Make sure that the disk you are adding has a different target number than the other devices on the system.**

   You will often find a small switch located at the back of the disk for this purpose.

3. **Connect the replacement system disk to the system and check the physical connections.**

   Refer to the disk's hardware installation guide for installation details.

4. **Follow steps a-e if you are booting from a local Solaris CD or DVD or a remote Solaris CD or DVD from the network.**

   If you are booting from the network, skip step a.

   a. **If you are booting from a local Solaris CD or DVD, insert the Solaris installation CD or DVD into the drive.**

   b. **Insert the Solaris boot diskette into the primary diskette drive (DOS drive A).**

   c. **Press any key to reboot the system if the system displays the `Type any key to continue` prompt. Or, use the reset button to restart the system if the system is shut down.**

      The Boot Solaris screen is displayed after a few minutes.

   d. **Select the CD-ROM drive or net(work) as the boot device from the Boot Solaris screen.**

      The Current Boot Parameters screen is displayed.

   e. **Boot the system in single-user mode.**

      ```
      Select the type of installation: b -s
      ```

      After a few minutes, the root prompt (#) is displayed.

x86: Where to Go From Here

After you boot the system, you can create an `fdisk` partition. Go to "x86: How to Create a Solaris `fdisk` Partition" on page 471.

▼ x86: How to Connect a Secondary Disk and Boot

1. **Become superuser or assume an equivalent role.**

2. **If the disk is unsupported by the Solaris software, add the device driver for the disk by following the instructions included with the hardware.**

3. **Create the `/reconfigure` file that will be read when the system is booted.**

   ```
   # touch /reconfigure
   ```

The `/reconfigure` file causes the SunOS software to check for the presence of any newly installed peripheral devices when you power on or boot your system later.

4. **Shut down the system.**

   ```
   # shutdown -i0 -gn -y
   ```

`-i0`	Brings the system down to run level 0, the power-down state.
`-g`*n*	Notifies logged-in users that they have *n* seconds before the system begins to shut down.
`-y`	Specifies that the command should run without user intervention.

   The `Type any key to continue` prompt is displayed.

5. **Turn off the power to the system and all external peripheral devices.**
6. **Make sure that the disk you are adding has a different target number than the other devices on the system.**

   You will often find a small switch located at the back of the disk for this purpose.

7. **Connect the disk to the system and check the physical connections.**

   Refer to the disk's hardware installation guide for installation details.

8. **Turn on the power to all external peripherals.**
9. **Turn on the power to the system.**

   The system boots and displays the login prompt.

x86: Where to Go From Here

After you boot the system, you can create an `fdisk` partition. Go to "x86: How to Create a Solaris `fdisk` Partition" on page 471.

x86: Guidelines for Creating an `fdisk` Partition

Follow these guidelines when you set up the `fdisk` partition.

- The disk can be divided into a maximum of four `fdisk` partitions. One of partitions must be a Solaris partition.
- The Solaris partition must be made the active partition on the disk. The active partition is partition whose operating system will be booted by default at system startup.
- Solaris `fdisk` partitions must begin on cylinder boundaries.

- Solaris `fdisk` partitions must begin at cylinder 1, not cylinder 0, on the first disk because additional boot information, including the master boot record, is written in sector 0.
- The Solaris `fdisk` partition can be the entire disk or you might want to make it smaller to allow room for a DOS partition. You can also make a new `fdisk` partition on a disk without disturbing existing partitions (if there is enough room to create a new one).

---

x86 only – Solaris slices are sometimes called partitions. This book uses the term slice, but some Solaris documentation and programs might refer to a *slice* as a *partition*.

To avoid confusion, Solaris documentation tries to distinguish between `fdisk` partitions (which are supported only on Solaris (x86 Platform Edition)) and the divisions within the Solaris `fdisk` partition, which might be called slices or partitions.

▼ x86: How to Create a Solaris `fdisk` Partition

1. **Read "x86: Guidelines for Creating an `fdisk` Partition" on page 470.**
2. **Become superuser or assume an equivalent role.**
3. **Invoke the `format` utility.**

   ```
   # format
   ```

   For more information, see `format`(1M).
4. **Type the number of the disk on which to create a Solaris `fdisk` partition from the list displayed on your screen.**

   ```
   Specify disk (enter its number): disk-number
   ```

   disk-number is the number of the disk on which you want to create a Solaris `fdisk` partition.
5. **Select the `fdisk` menu.**

   ```
   format> fdisk
   ```

   The `fdisk` menu that is displayed depends upon whether the disk has existing `fdisk` partitions. Determine the next step using the following table.

Task	Go To	For More Information
Create a Solaris `fdisk` partition to span the entire disk.	Step 6	"x86: Example—Creating a Solaris `fdisk` Partition That Spans the Entire Drive" on page 474

Task	Go To	For More Information
Create a Solaris `fdisk` partition and preserve one or more existing non-Solaris `fdisk` partition.	Step 7	"x86: Example—Creating a Solaris `fdisk` Partition While Preserving an Existing `fdisk` Partition" on page 474
Create a Solaris `fdisk` partition and one or more additional non-Solaris `fdisk` partition.	Step 7	"x86: Example—Creating a Solaris `fdisk` Partition and an Additional `fdisk` Partition" on page 475

6. **Create and activate a Solaris `fdisk` partition that spans the entire disk by specifying `y` at the prompt. Then, go to step 14.**

```
The recommended default partitioning for your disk is:

  a 100% "SOLARIS System" partition.

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

7. **Specify `n` at the prompt if you do not want the Solaris `fdisk` partition to span the entire disk.**

```
To select this, please type "y".  To partition your disk
differently, type "n" and the "fdisk" program will let you
select other partitions. n
Total disk size is 2694 cylinders
             Cylinder size is 765 (512 byte) blocks
                                           Cylinders
    Partition   Status    Type      Start   End   Length    %
    =========   ======    ========  =====   ===   ======   ===
THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE
FOLLOWING:

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

8. **Select option 1, `Create a partition`, to create an `fdisk` partition.**

```
             Total disk size is 2694 cylinders
             Cylinder size is 765 (512 byte) blocks
                                           Cylinders
    Partition   Status    Type      Start   End   Length    %
    =========   ======    ========  =====   ===   ======   ===
THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE
FOLLOWING:
```

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

9. **Create a Solaris `fdisk` partition by selecting `1(=Solaris).`**

```
Indicate the type of partition you want to create
  (1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
  (5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ? 1
```

10. **Identify the percentage of the disk to be reserved for the Solaris `fdisk` partition. Keep in mind the size of any existing `fdisk` partitions when you calculate this percentage.**

```
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). nn
```

11. **Activate the Solaris `fdisk` partition by typing `y` at the prompt.**

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

    The `Enter Selection:` prompt is displayed after the `fdisk` partition is activated.

12. **Select option 1, `Create a partition`, to create another `fdisk` partition.**

    See steps 9-11 for instructions on creating an `fdisk` partition.

13. **Update the disk configuration and exit the `fdisk` menu from the selection menu.**

```
Selection: 4
```

14. **Relabel the disk by using the `label` command.**

```
WARNING: Solaris fdisk partition changed - Please relabel the disk
format> label
Ready to label disk, continue? yes
format>
```

15. **Quit the `format` menu.**

```
format> quit
```

x86: Where to Go From Here

After you create a Solaris `fdisk` partition on the disk, you can create slices on the disk. Go to “x86: How to Create Disk Slices and Label a Disk” on page 476.

x86: Example—Creating a Solaris `fdisk` Partition That Spans the Entire Drive

The following example uses the `format`'s utility's `fdisk` option to create a Solaris `fdisk` partition that spans the entire drive.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
       0. c0d0 <DEFAULT cyl 2466 alt 2 hd 16 sec 63>
          /pci@0,0/pci-ide@7,1/ide@0/cmdk@0,0
       1. c0d1 <DEFAULT cyl 522 alt 2 hd 32 sec 63>
          /pci@0,0/pci-ide@7,1/ide@0/cmdk@1,0
       2. c1d0 <DEFAULT cyl 13102 alt 2 hd 16 sec 63>
          /pci@0,0/pci-ide@7,1/ide@1/cmdk@0,0
Specify disk (enter its number): 0
selecting c0d0
Controller working list found
[disk formatted]
format> fdisk
The recommended default partitioning for your disk is:

  a 100% "SOLARIS System" partition.

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

WARNING: Solaris fdisk partition changed - Please relabel the disk
format> label
Ready to label disk, continue? yes
format> quit
```

x86: Example—Creating a Solaris `fdisk` Partition While Preserving an Existing `fdisk` Partition

The following example shows how to create a Solaris `fdisk` partition on a disk that has an existing `DOS-BIG` `fdisk` partition.

```
format> fdisk
             Total disk size is 2694 cylinders
             Cylinder size is 765 (512 byte) blocks
                                          Cylinders
     Partition   Status    Type     Start  End   Length    %
     =========   ======   ========  =====   ===   ======  ===
        1                 DOS-BIG       1   538     538   20
SELECT ONE OF THE FOLLOWING:
     1.   Create a partition
     2.   Change Active (Boot from) partition
     3.   Delete a partition
     4.   Exit (Update disk configuration and exit)
     5.   Cancel (Exit without updating disk configuration)
```

```
Enter Selection: 1
Indicate the type of partition you want to create
  (1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
  (5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?1
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". y
Partition 2 is now the Active partition Total disk size is 2694
cylinders
             Cylinder size is 765 (512 byte) blocks
                                          Cylinders
    Partition   Status    Type      Start   End   Length    %
    =========   ======    ========  =====   ===   ======   ===
        1                 DOS-BIG       1   538      538     20
        2       Active    SOLARIS     539  2693     2155     80
SELECT ONE OF THE FOLLOWING:
     1.   Create a partition
     2.   Change Active (Boot from) partition
     3.   Delete a partition
     4.   Exit (Update disk configuration and exit)
     5.   Cancel (Exit without updating disk configuration)
Enter Selection: Selection: 4
WARNING: Solaris fdisk partition changed - Please relabel the disk
format> label
Ready to label disk, continue? yes
format> q
```

x86: Example—Creating a Solaris `fdisk` Partition and an Additional `fdisk` Partition

This following example shows how to create a Solaris `fdisk` partition and a `DOSBIG` `fdisk` partition.

```
format> fdisk
The recommended default partitioning for your disk is:
   a 100% "SOLARIS System" partition.
To select this, please type "y".  To partition your disk
differently, type "n" and the "fdisk" program will let you
select other partitions. n
             Total disk size is 2694 cylinders
             Cylinder size is 765 (512 byte) blocks
                                       Cylinders
    Partition   Status    Type      Start   End   Length    %
    =========   ======    ========  =====   ===   ======   ===
THERE ARE NO PARTITIONS CURRENTLY DEFINED SELECT ONE OF THE FOLLOWING:
     1.   Create a partition
     2.   Change Active (Boot from) partition
     3.   Delete a partition
     4.   Exit (Update disk configuration and exit)
     5.   Cancel (Exit without updating disk configuration)
```

```
Enter Selection: 1
Indicate the type of partition you want to create
  (1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
  (5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?8
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 20
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". n
             Total disk size is 2694 cylinders
             Cylinder size is 765 (512 byte) blocks
                                          Cylinders
    Partition   Status    Type      Start   End   Length    %
    =========   ======    ========  =====   ===   ======   ===
        1                 DOS-BIG       1   538     538     20
SELECT ONE OF THE FOLLOWING:
     1.   Create a partition
     2.   Change Active (Boot from) partition
     3.   Delete a partition
     4.   Exit (Update disk configuration and exit)
     5.   Cancel (Exit without updating disk configuration)Enter
Selection: 1
Indicate the type of partition you want to create
  (1=SOLARIS, 2=UNIX, 3=PCIXOS, 4=Other, 8=DOSBIG)
  (5=DOS12, 6=DOS16, 7=DOSEXT, 0=Exit) ?1
Indicate the percentage of the disk you want this partition
to use (or enter "c" to specify in cylinders). 80
Do you want this to become the Active partition? If so, it will be
activated each time you reset your computer or when you turn it on
again. Please type "y" or "n". y
Partition 2 is now the Active partition Total disk size is 2694
cylinders
            Cylinder size is 765 (512 byte) blocks
                                           Cylinders
    Partition   Status    Type      Start   End   Length    %
    =========   ======    ========  =====   ===   ======   ===
        1                 DOS-BIG       1   538     538     20
        2       Active    SOLARIS     539  2693    2155     80

SELECT ONE OF THE FOLLOWING:
     1.   Create a partition
     2.   Change Active (Boot from) partition
     3.   Delete a partition
     4.   Exit (Update disk configuration and exit)
     5.   Cancel (Exit without updating disk configuration)
Enter Selection: 4
format> q
```

▼ x86: How to Create Disk Slices and Label a Disk

1. **Become superuser or assume an equivalent role.**
2. **Start the `format` utility.**

```
# format
```

3. **Type the number of the disk that you want to repartition from the list displayed on your screen.**

   ```
   Specify disk (enter its number): disk-number
   ```

   disk-number is the number of the disk that you want to repartition.

4. **Select the `partition` menu.**

   ```
   format> partition
   ```

5. **Display the current partition (slice) table.**

   ```
   partition> print
   ```

6. **Start the modification process.**

   ```
   partition> modify
   ```

7. **Set the disk to all free hog.**

   ```
   Choose base (enter number) [0]? 1
   ```

   For more information about the free hog slice, see "Using the Free Hog Slice" on page 433.

8. **Create a new partition table by answering `yes` when prompted to continue.**

   ```
   Do you wish to continue creating a new partition
   table based on above table[yes]? yes
   ```

9. **Identify the free hog partition (slice) and the sizes of the slices when prompted.**

   When adding a system disk, you must set up slices for:

   - root (slice 0) and swap (slice 1) and/or
   - `/usr` (slice 6)

   After you identify the slices, the new partition table is displayed.

10. **Make the displayed partition table the current partition table by answering `yes` when asked.**

    ```
    Okay to make this the current partition table[yes]? yes
    ```

    If you don't want the current partition table and you want to change it, answer `no` and go to Step 6.

11. **Name the partition table.**

    ```
    Enter table name (remember quotes): "partition-name"
    ```

    partition-name is the name for the new partition table.

12. **Label the disk with the new partition table after you have finished allocating slices on the new disk.**

```
Ready to label disk, continue? yes
```

13. **Quit the `partition` menu.**

```
partition> quit
```

14. **Verify the new disk label.**

```
format> verify
```

15. **Exit the `format` menu.**

```
format> quit
```

x86: Where to Go From Here

After you create disk slices and label the disk, you can create file systems on the disk. Go to "x86: How to Create File Systems" on page 478.

▼ x86: How to Create File Systems

1. **Become superuser or assume an equivalent role.**
2. **Create a file system for each slice.**

```
# newfs /dev/rdsk/cwtxdysz
```

`/dev/rdsk/cwtxdysz` is the raw device for the file system to be created.

For more information about the `newfs` command, see Chapter 39 or `newfs`(1M).

3. **Verify the new file system by mounting.**

```
# mount /dev/dsk/cwtxdysz /mnt
# ls /mnt
lost+found
```

x86: Where to Go From Here

Add Disk Task	Action
System Disk	You need to restore the root (/) and `/usr` file systems on the disk. Go to Chapter 49. After the root (/) and `/usr` file systems are restored, install the boot block. Go to "x86: How to Install a Boot Block on a System Disk" on page 479.

Add Disk Task	Action
Secondary Disk	You might need to restore file systems on the new disk. Go to Chapter 49. If you are not restoring file systems on the new disk, you are finished adding a secondary disk. See Chapter 40 for information on making the file systems available to users.

▼ x86: How to Install a Boot Block on a System Disk

1. **Become superuser or assume an equivalent role.**
2. **Install the boot block on the system disk.**

```
# installboot /usr/platform/‘uname -i‘/lib/fs/ufs/pboot /usr/platform/‘uname -i‘
/lib/fs/ufs/bootblk /dev/rdsk/cwtxdys2
```

`/usr/platform/‘uname -i‘/lib/fs/ufs/pboot`	Is the partition boot file.
`/usr/platform/‘uname -i‘/lib/fs/ufs/bootblk`	Is the boot block code.
`/dev/rdsk/`*cwtxdy*`s2`	Is the raw device name that represents the whole disk.

3. **Verify that the boot blocks are installed by rebooting the system to run level 3.**

```
# init 6
```

x86: Example—Installing a Boot Block on a System Disk

```
# installboot /usr/platform/i86pc/lib/fs/ufs/pboot
 /usr/platform/i86pc/lib/fs/ufs/bootblk /dev/rdsk/c0t6d0s2
```

CHAPTER 36

The format Utility (Reference)

This chapter describes the format utility's menu and commands.

This is a list of the reference information in this chapter.

- "Recommendations and Requirements for Using The format Utility" on page 481
- "Format Menu and Command Descriptions" on page 482
- "The format.dat File" on page 488
- "Rules for Input to format Commands" on page 493
- "Getting Help on the format Utility" on page 495

For a overview of when to use the format utility, see Chapter 32.

Recommendations and Requirements for Using The format Utility

You must be superuser or a member of an equivalent role to use the format utility. If you are not superuser or have assumed an equivalent role, you will see the following error message when trying to use the format utility:

```
$ format
Searching for disks...done
No permission (or no disks found)!
```

Keep the following guidelines in mind when using the format utility and you want to preserve the existing data:

- Back up all files on the disk drive.
- Save all your defect lists in files by using the format utility's dump command. The file name should include the drive type, model number, and serial number.

- Save the paper copies of the manufacturer's defect list that was shipped with your drive.

Format Menu and Command Descriptions

The `format` main menu looks like the following:

```
FORMAT MENU:
        disk       - select a disk
        type       - select (define) a disk type
        partition  - select (define) a partition table
        current    - describe the current disk
        format     - format and analyze the disk
        repair     - repair a defective sector
        label      - write label to the disk
        analyze    - surface analysis
        defect     - defect list management
        backup     - search for backup labels
        verify     - read and display labels
        save       - save new disk/partition definitions
        inquiry    - show vendor, product and revision
        volname    - set 8-character volume name
        quit
format>
```

The following table describes the `format` main menu items.

TABLE 36–1 The `format` Main Menu Item Descriptions

Item	Command or Menu?	Description
`disk`	Command	Lists all of the system's drives. Also lets you choose the disk you want to use in subsequent operations. This disk is referred to as the current disk.
`type`	Command	Identifies the manufacturer and model of the current disk. Also displays a list of known drive types. Choose the `Auto configure` option for all SCSI-2 disk drives.
`partition`	Menu	Creates and modifies slices. For more information, see "The `partition` Menu" on page 484.

TABLE 36–1 The `format` Main Menu Item Descriptions *(Continued)*

Item	Command or Menu?	Description
`current`	Command	Displays the following information about the current disk: ■ Device name and device type ■ Number of cylinders, alternate cylinders, heads and sectors ■ Physical device name
`format`	Command	Formats the current disk by using one of these sources of information in this order: 1. Information that is found in the `format.dat` file 2. Information from the automatic configuration process 3. Information that you enter at the prompt if there is no `format.dat` entry This command does not apply to IDE disks. IDE disks are pre–formatted by the manufacturer.
`fdisk`	Menu	x86 platform only: Runs the `fdisk` program to create a Solaris `fdisk` partition.
`repair`	Command	Repairs a specific block on the current disk.
`label`	Command	Writes a new label to the current disk.
`analyze`	Menu	Runs read, write, compare tests. For more information, see "The `analyze` Menu" on page 486.
`defect`	Menu	Retrieves and prints defect lists. For more information, see "The `defect` Menu" on page 487. This feature does not apply to IDE disks. IDE disks perform automatic defect management.
`backup`	Command	**VTOC** – Searches for backup labels. **EFI** – Not supported.
`verify`	Command	Prints the following information about the current disk: ■ Device name and device type ■ Number of cylinders, alternate cylinders, heads and sectors ■ Partition table
`save`	Command	**VTOC** –Saves new disk and partition information. **EFI** – Not applicable.
`inquiry`	Command	Prints the vendor, product name, and revision level of the current drive (SCSI disks only).

TABLE 36–1 The `format` Main Menu Item Descriptions *(Continued)*

Item	Command or Menu?	Description
`volname`	Command	Labels the disk with a new eight-character volume name.
`quit`	Command	Exits the `format` menu.

The `partition` Menu

The `partition` menu looks similar to the following:

```
format> partition
PARTITION MENU:
        0      - change '0' partition
        1      - change '1' partition
        2      - change '2' partition
        3      - change '3' partition
        4      - change '4' partition
        5      - change '5' partition
        6      - change '6' partition
        7      - change '7' partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        quit
partition>
```

The following table describes the `partition` menu items.

TABLE 36–2 Descriptions for `partition` Menu Items

Sub-Command	Description
`change '`*n*`' partition`	Lets you specify the following information for the new slice: ■ Identification tag ■ Permission flags ■ Starting cylinder ■ Size
`select`	Lets you choose a predefined slice table.
`modify`	Lets you change all the slices in the slice table. This command is preferred over the individual `change '`*x*`' partition` commands.
`name`	Lets you specify a name for the current slice table.

TABLE 36–2 Descriptions for `partition` Menu Items *(Continued)*

Sub-Command	Description
`print`	Displays the current slice table.
`label`	Writes the slice map and the label to the current disk.
`quit`	Exits the partition menu.

x86: The `fdisk` Menu

The `fdisk` menu appears on x86 based systems only and looks similar to the following.

```
format> fdisk
             Total disk size is 1855 cylinders
             Cylinder size is 553 (512 byte) blocks
                                              Cylinders
      Partition   Status    Type          Start   End   Length    %
      =========   ======    ========      =====   ===   ======   ===
          1                 DOS-BIG           0   370      371    20
          2       Active    SOLARIS         370  1851     1482    80

SELECT ONE OF THE FOLLOWING:
     1.   Create a partition
     2.   Change Active (Boot from) partition
     3.   Delete a partition
     4.   Exit (Update disk configuration and exit)
     5.   Cancel (Exit without updating disk configuration)
Enter Selection:
```

The following table describes the `fdisk` menu items.

TABLE 36–3 x86: Descriptions for `fdisk` Menu Items

Menu Item	Description
`Create a partition`	Creates an `fdisk` partition. You must create a separate partition for each operating environment such as Solaris or DOS. There is a maximum of 4 partitions per disk. You are prompted for the size of the `fdisk` partition as a percentage of the disk.
`Change Active partition`	Lets you specify the partition to be used for booting. This menu item identifies where the first stage boot program looks for the second stage boot program.
`Delete a partition`	Deletes a previously created partition. This command destroys all the data in the partition.

TABLE 36–3 x86: Descriptions for `fdisk` Menu Items *(Continued)*

Menu Item	Description
`Exit`	Writes a new version of the partition table and exits the `fdisk` menu.
`Cancel`	Exits the `fdisk` menu without modifying the partition table.

The `analyze` Menu

The `analyze` menu looks similar to the following.

```
format> analyze

ANALYZE MENU:
    read     - read only test   (doesn't harm SunOS)
    refresh  - read then write  (doesn't harm data)
    test     - pattern testing  (doesn't harm data)
    write    - write then read      (corrupts data)
    compare  - write, read, compare (corrupts data)
    purge    - write, read, write   (corrupts data)
    verify   - write entire disk, then verify (corrupts data)
    print    - display data buffer
    setup    - set analysis parameters
    config   - show analysis parameters
    quit
analyze>
```

The following table describes the `analyze` menu items.

TABLE 36–4 Descriptions for `analyze` Menu Item

Sub-Command	Description
`read`	Reads each sector on the current disk. Repairs defective blocks as a default.
`refresh`	Reads then writes data on the current disk without harming the data. Repairs defective blocks as a default.
`test`	Writes a set of patterns to the disk without harming the data. Repairs defective blocks as a default.
`write`	Writes a set of patterns to the disk then reads the data on the disk back. Destroys existing data on the disk. Repairs defective blocks as a default.
`compare`	Writes a set of patterns to the disk, reads the data back, and then compares it to the data in the write buffer. Destroys existing data on the disk. Repairs defective blocks as a default.

TABLE 36–4 Descriptions for `analyze` Menu Item *(Continued)*

Sub-Command	Description
`purge`	Removes all data from the disk so that the data can't be retrieved by any means. Data is removed by writing three distinct patterns over the entire disk (or a section of the disk). If the verification passes, a hex-bit pattern is written over the entire disk (or a section of the disk). Repairs defective blocks as a default.
`verify`	Writes unique data to each block on the entire disk in the first pass. Reads and verifies the data in the next pass. Destroys existing data on the disk. Repairs defective blocks as a default.
`print`	Displays the data in the read/write buffer.
`setup`	Lets you specify the following analysis parameters: `Analyze entire disk?` **`yes`** Starting block number: *depends on drive* Ending block number: *depends on drive* Loop continuously? **`no`** Number of passes: **`2`** Repair defective blocks? **`yes`** Stop after first error? **`no`** Use random bit patterns? **`no`** Number of blocks per transfer: **`126 (0/`***`n`***`/`***`nn`***`)`** Verify media after formatting? **`yes`** Enable extended messages? **`no`** Restore defect list? **`yes`** Restore disk label? **`yes`** Defaults are shown in bold.
`config`	Displays the current analysis parameters.
`quit`	Exits the `analyze` menu.

The `defect` Menu

The `defect` menu looks similar to the following:

```
format> defect

DEFECT MENU:
        primary  - extract manufacturer's defect list
        grown    - extract manufacturer's and repaired defects lists
        both     - extract both primary and grown defects lists
        print    - display working list
        dump     - dump working list to file
        quit
defect>
```

The following table describes the `defect` menu items.

TABLE 36–5 The `defect` Menu Item Descriptions

Sub-Command	Description
`primary`	Reads the manufacturer's defect list from the disk drive and updates the in-memory defect list.
`grown`	Reads the grown defect list, which are defects that have been detected during analysis, and then updates the in-memory defect list.
`both`	Reads both the manufacturer's defect list and the grown defect list, and then updates the in-memory defect list.
`print`	Displays the in-memory defect list.
`dump`	Saves the in-memory defect list to a file.
`quit`	Exits the `defect` menu.

The `format.dat` File

The `format.dat` file that is shipped with the Solaris operating environment supports many standard disks. If your disk drive is not listed in the `format.dat` file, you can choose to add an entry for it or adding entries with the `format` utility by selecting the `type` command and choosing the `other` option.

Adding an entry to the `format.dat` file can save time if the disk drive will be used throughout your site. To use the `format.dat` file on other systems, copy the file to each system that will use the specific disk drive that you added to the `format.dat` file.

You should modify the `/etc/format.dat` file for your system if you have one of the following:

- A disk that is not supported by the Solaris operating environment
- A disk with a slice table that is different from the Solaris operating environment default configuration

Note – Do not alter default entries in the `/etc/format.dat` file. If you want to alter the default entries, copy the entry, give it a different name, and make the appropriate changes to avoid confusion.

The `/etc/format.dat` is not applicable for disks with EFI labels.

Contents of the `format.dat` File

The `format.dat` contains specific disk drive information that is used by the `format` utility. Three items are defined in the `format.dat` file:

- Search paths
- Disk types
- Slice tables

Syntax of the `format.dat` File

The following syntax rules apply to the `/etc/format.dat` file:

- The pound sign (`#`) is the comment character. Any text on a line after a pound sign is not interpreted by the `format` utility.
- Each definition in the `format.dat` file appears on a single logical line. If the definition is longer than one line long, all but the last line of the definition must end with a backslash (`\`).
- A definition consists of a series of assignments that have an identifier on the left side and one or more values on the right side. The assignment operator is the equal sign (`=`). The assignments within a definition must be separated by a colon (`:`).
- White space is ignored by the `format` utility. If you want an assigned value to contain white space, enclose the entire value in double quotation marks (`"`). This syntax will cause the white space within the quotes to be preserved as part of the assignment value.
- Some assignments can have multiple values on the right hand side. Separate values by a comma.

Keywords in the `format.dat` File

The `format.dat` file contains disk definitions that are read by the `format` utility when it is started. Each definition starts with one of the following keywords: `disk_type` or `partition`. These keywords are described in the following table.

TABLE 36–6 Keyword Descriptions for the `format.dat` File

Keyword	Use
`disk_type`	Defines the controller and disk model. Each `disk_type` definition contains information that concerns the physical geometry of the disk. The default data file contains definitions for the controllers and disks that the Solaris operating environment supports. You need to add a new `disk_type` only if you have an unsupported disk. You can add as many `disk_type` definitions to the data file as you want.
`partition`	Defines a slice table for a specific disk type. The slice table contains the slice information, plus a name that lets you refer to it in the `format` utility. The default `format.dat` file contains default slice definitions for several kinds of disk drives. Add a slice definition if you recreated slices on any of the disks on your system. Add as many slice definitions to the data file as you need.

Disk Type (`format.dat`)

The `disk_type` keyword in the `format.dat` file defines the controller and disk model. Each `disk_type` definition contains information about the physical geometry of the disk. The default `format.dat` file contains definitions for the controllers and disks that the Solaris operating environment supports. You need to add a new `disk_type` only if you have an unsupported disk. You can add as many `disk_type` definitions to the data file as you want.

The keyword itself is assigned the name of the disk type. This name appears in the disk's label, and is used to identify the disk type whenever the `format` utility is run. Enclose the name in double quotation marks to preserve any white space in the name. The following table describes the identifiers that must also be assigned values in all `disk_type` definitions.

TABLE 36–7 Required `disk_type` Identifiers

Identifier	Description
`ctlr`	Identifies the controller type for the disk type. Currently, the supported values are SCSI and ATA.
`ncyl`	Specifies the number of data cylinders in the disk type. This determines how many logical cylinders of the disk the system will be allowed to access.
`acyl`	Specifies the number of alternate cylinders in the disk type. These cylinders are used by the `format` utility to store information such as the defect list for the drive. You should always leave at least two cylinders for alternates.

TABLE 36–7 Required `disk_type` Identifiers *(Continued)*

Identifier	Description
`pcyl`	Specifies the number of physical cylinders in the disk type. This number is used to calculate the boundaries of the disk media. This number is usually equal to `ncyl` plus `acyl`.
`nhead`	Specifies the number of heads in the disk type. This number is used to calculate the boundaries of the disk media.
`nsect`	Specifies the number of data sectors per track in the disk type. This number is used to calculate the boundaries of the disk media. Note that this is only the data sectors. Any spares are not reflected in the number of data sections per track.
`rpm`	The rotations per minute of the disk type. This information is put in the label and later used by the file system to calculate the optimal placement of file data.

Other identifiers might be necessary, depending on the controller. The following table describes the identifiers that are required for SCSI controllers.

TABLE 36–8 `disk_type` Identifiers for SCSI Controllers

Identifier	Description
`fmt_time`	A number that Indicates how long it takes to format a given drive. See the controller manual for more information.
`cache`	A number that controls the operation of the on-board cache while the `format` utility is operating. See the controller manual for more information.
`trks_zone`	A number that specifies how many tracks you have per defect zone, to be used in alternate sector mapping. See the controller manual for more information.
`asect`	A number that specifies how many sectors are available for alternate mapping within a given defect zone. See the controller manual for more information.

The following are examples of `disk_type` definitions:

```
disk_type = "SUN1.3G" \
        : ctlr = SCSI : fmt_time = 4 \
        : trks_zone = 17 : asect = 6 : atrks = 17 \
        : ncyl = 1965 : acyl = 2 : pcyl = 3500 : nhead = 17 : nsect = 80 \
        : rpm = 5400 : bpt = 44823

disk_type = "SUN2.1G" \
        : ctlr = SCSI : fmt_time = 4 \
        : ncyl = 2733 : acyl = 2 : pcyl = 3500 : nhead = 19 : nsect = 80 \
        : rpm = 5400 : bpt = 44823

disk_type = "SUN2.9G" \
        : ctlr = SCSI : fmt_time = 4 \
```

```
        : ncyl = 2734 : acyl = 2 : pcyl = 3500 : nhead = 21 : nsect = 99 \
        : rpm = 5400
```

Partition or Slice Tables (`format.dat`)

A partition table in the `format.dat` file defines a slice table for a specific disk type.

The `partition` keyword in the `format.dat` file is assigned the name of the slice table. Enclose the name in double quotation marks to preserve any white space in the name. The following table describes the identifiers that must be assigned values in all slice tables.

TABLE 36–9 Required Identifiers for Slice Tables

Identifier	Description
`disk`	The name of the `disk_type` that this slice table is defined for. This name must appear exactly as it does in the `disk_type` definition.
`ctlr`	The disk controller type that this slice table can be attached to. Currently, the supported values are ATA for ATA controllers and SCSI for SCSI controllers. The controller type that is specified here must also be defined for the `disk_type` that you specified in the `disk_type` definition.

The other identifiers in a slice definition describe the actual slice information. The identifiers are the numbers 0 through 7. These identifiers are optional. Any slice that is not explicitly assigned is set to 0 length. The value of each of these identifiers is a pair of numbers separated by a comma. The first number is the starting cylinder for the slice, and the second is the number of sectors in the slice. The following are some examples of slice definitions:

```
partition = "SUN1.3G" \
        : disk = "SUN1.3G" : ctlr = SCSI \
        : 0 = 0, 34000 : 1 = 25, 133280 : 2 = 0, 2672400 : 6 = 123, 2505120

partition = "SUN2.1G" \
        : disk = "SUN2.1G" : ctlr = SCSI \
        : 0 = 0, 62320 : 1 = 41, 197600 : 2 = 0, 4154160 : 6 = 171, 3894240

partition = "SUN2.9G" \
        : disk = "SUN2.9G" : ctlr = SCSI \
        : 0 = 0, 195426 : 1 = 94, 390852 : 2 = 0, 5683986 : 6 = 282, 5097708
```

Specifying an Alternate Data File for the `format` utility

The `format` utility learns of the location of an alternate file by the following methods.

1. If a file name is given with the `format -x` option, that file is always used as the data file.
2. If the `-x` option is not specified, then the `format` utility looks in the current directory for a file named `format.dat`. If the file exists, it is used as the data file.
3. If neither of these methods yields a data file, the `format` utility uses the `/etc/format.dat` file as the data file. This file is shipped with the Solaris operating environment and should always be present.

Rules for Input to `format` Commands

When you use the `format` utility, you need to provide various kinds of information. This section describes the rules for this information. For information on using `format`'s help facility when you enter data, see "Getting Help on the `format` Utility" on page 495.

Specifying Numbers to `format` Commands

Several places in the `format` utility require an number as input. You must either specify the data or select a number from a list of choices. In either case, the `help` facility causes `format` to print the upper and lower limits of the number expected. Simply enter the number desired. The number is assumed to be in decimal format unless a base is explicitly specified as part of the number (for example, 0x for hexadecimal).

The following are examples of integer input:

```
Enter number of passes [2]: 34
Enter number of passes [34] 0xf
```

Specifying Block Numbers to `format` Commands

Whenever you are required to specify a disk block number, there are two ways to enter the information:

- Block number as an integer
- Block number in the cylinder/head/sector format

You can specify the information as an integer that represents the logical block number. You can specify the number in any base, but the default is decimal. The maximum operator (a dollar sign, `$`) can also be used here to let the `format` utility select the appropriate value. Logical block format is used by the SunOS disk drivers in error messages.

The other way to specify a block number is by the cylinder/head/sector designation. In this method, you must specify explicitly the three logical components of the block number: the cylinder, head, and sector values. These values are still logical, but they allow you to define regions of the disk that are related to the layout of the media.

If any of the cylinder/head/sector numbers are not specified, the value is assumed to be zero. You can also use the maximum operator in place of any of the numbers and let the `format` utility select the appropriate value. The following are some examples of cylinder, head, and sector entries:

```
Enter defective block number: 34/2/3
Enter defective block number: 23/1/
Enter defective block number: 457//
Enter defective block number: 12345
Enter defective block number: Oxabcd
Enter defective block number: 334/$/2
Enter defective block number: 892//$
```

The `format` utility always prints block numbers, in both formats. Also, the `help` facility shows you the upper and lower bounds of the block number expected, in both formats.

Specifying `format` Command Names

Command names are needed as input whenever the `format` utility displays a menu prompt. You can *abbreviate* the command names, as long as what you enter is sufficient to uniquely identify the command desired.

For example, use `p` to enter the `partition` menu from the `format` menu. Then, enter `p` to display the current slice table.

```
format> p
PARTITION MENU:
        0      - change '0' partition
        1      - change '1' partition
        2      - change '2' partition
        3      - change '3' partition
        4      - change '4' partition
        5      - change '5' partition
        6      - change '6' partition
        7      - change '7' partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        quit
partition> p
```

Specifying Disk Names to `format` Commands

There are certain times in the `format` utility when you must name something. In these cases, you are free to specify any string you want for the name. If the name has white space in it, the entire name must be enclosed in double quotation marks (`"`). Otherwise, only the first word of the name is used.

For example, if you want to identify a specific partition table for a disk, you can use the name sub-command available from the partition menu:

```
partition> name
Enter table name (remember quotes): "new disk3"
```

Getting Help on the `format` Utility

The `format` utility provides a help facility that you can use whenever the `format` utility is expecting input. You can request help about what input is expected by entering a question mark (`?`). The `format` utility displays a brief description of what type of input is needed.

If you enter a `?` at a menu prompt, a list of available commands is displayed.

The man pages associated with the `format` utility include the following:

- `format`(1M) - Describes the basic `format` utility capabilities and provides descriptions of all command-line variables.
- `format.dat`(4) - Describes disk drive configuration information for the `format` utility.

CHAPTER 37

Managing File Systems Topics

This topic map lists the chapters that provide information on managing file systems.

Chapter 38	Provides a high-level overview of file system topics, including descriptions of file system types, commonly used administration commands, and the mounting and unmounting of file systems. Also includes the procedure for determining a file system's type.
Chapter 39	Provides step-by-step instructions for creating a UFS file system, a temporary file system (TMPFS), and a loopback file system (LOFS).
Chapter 40	Provides step-by-step instructions for determining which file systems are mounted, how to mount files listed in the `/etc/vfstab` file, and how to mount UFS, NFS, and PCFS (DOS) file systems.
Chapter 41	Provides overview information and step-by-step instructions for using the CacheFS file system.
Chapter 42	Provides step-by-step instructions for monitoring swap resources, creating swap files and making them available, and removing unneeded swap space.
Chapter 43	Provides information on how the file system state is recorded, what `fsck` program checks, how to modify automatic boot checking, and how to use the `fsck` program.
Chapter 44	Provides reference information for file systems, including default directories for the root (`/`) and `/usr` file systems, default directories that are contained in the `/kernel` directory, and specifics for the `mkfs` and `newfs` commands.

CHAPTER 38

Managing File Systems (Overview)

The management of file systems is one of your most important system administration tasks.

This is a list of the overview information in this chapter.

- "What's New in File Systems?" on page 499
- "Where to Find File System Management Tasks" on page 502
- "Overview of File Systems" on page 502
- "Types of File Systems" on page 503
- "Commands for File System Administration" on page 507
- "The Default Solaris File Systems" on page 509
- "Swap Space" on page 510
- "The UFS File System" on page 510
- "Mounting and Unmounting File Systems" on page 513
- "Determining a File System's Type" on page 518

What's New in File Systems?

This section describes new file system features in the Solaris 9 release.

Extended File Attributes

The UFS, NFS, and TMPFS file systems have been enhanced to include extended file attributes, which enable application developers to associate specific attributes to a file. For example, a developer of a windowing system file management application might choose to associate a display icon with a file. Extended file attributes are logically represented as files within a hidden directory that is associated with the target file.

You can use the `runat` command to add attributes and execute shell commands in the extended attribute name space, which is a hidden attribute directory that is associated with the specified file.

To use the `runat` command to add attributes to a file, you first have to create the attributes file.

```
$ runat filea cp /tmp/attrdata attr.1
```

Then, use the `runat` command to list the attributes of a file.

```
$ runat filea ls -l
```

For more information, see the `runat`(1) man page.

Many Solaris file system commands have been modified to support file system attributes by providing an attribute-aware option that you can use to query, copy, or find file attributes. For more information, see the specific man page for each file system command.

UFS Snapshots

You can use the `fssnap` command to create a read-only snapshot of a file system. A snapshot is a file system's temporary image that is intended for backup operations.

See Chapter 48 for more information.

Improved UFS Direct I/O Concurrency

The performance of direct I/O, which is used by database applications to access unbuffered file system data, has been improved by allowing concurrent read and write access to regular UFS files. Previously, an operation that updated file data would lock out all other read or write accesses until the update operation was completed.

Concurrent writes are restricted to the special case of file rewrites. If the file is being extended, writing is single threaded as before. Generally, databases pre-allocate files and seldomly extend them thereafter. Therefore, the effects of this enhancement are evident during normal database operations.

The direct I/O improvements brings I/O bound database performance on a UFS file system to about 90% of raw partition access speeds. If the database is CPU bound or bus bandwidth bound, there might be no improvement.

Consider running your I/O database applications with direct I/O enabled if you are already using UFS to store database tables. Use your database administrative procedures to enable direct I/O, if possible. If there is no way to enable direct I/O through your database product, use the `mount -forcedirectio` option to enable direct I/O for each file system. Or, use the `directio(3C)` library call to enable direct I/O.

See `mount_ufs`(1M) or `directio`(3C) for more information.

Improved `mkfs` Performance

The `mkfs` command now has improve performance when you create file systems. Improved `mkfs` performance is often 10 times faster than in previous Solaris releases. Performance improvements occur on systems when you create both large and small file systems. However, the biggest performance improvements occur when creating file systems on systems with high-capacity disks or high-speed disks.

New `labelit` Options for UDF File Systems

The `labelit` command provides new options for use with Universal Disk Format (UDF) file systems. You can use the new `labelit` command options to identify the author name, organization, and contact information for a UDF volume.

There was no mechanism to update this information, which is part of general UDF file systems, in previous Solaris releases.

The new UDF specific options for the `labelit` command, specified with the `-o` option, are the following:

- `lvinfo1` - Identifies the person who is creating the file system
- `lvinfo2` - Identifies the organization that is responsible for creating the file system
- `lvinfo3` - Identifies the contact information for media that contains the UDF file system

The maximum length for each option is 35 bytes.

For more information, see `labelit_udfs`(1M).

Where to Find File System Management Tasks

Use these references to find step-by-step instructions for the management of file systems.

File System Management Task	For More Information
Create new file systems	Chapter 39 and Chapter 41
Make local and remote files available to users	Chapter 40
Connect and configure new disk devices	Chapter 32
Design and implement a backup schedule and restoring files and file systems, as needed	Chapter 46
Check for and correct file system inconsistencies	Chapter 43

Overview of File Systems

A file system is a structure of directories that is used to organize and store files. The term *file system* is used to describe the following:

- A particular type of file system: disk-based, network-based, or virtual
- The entire file tree, begining with the root directory
- The data structure of a disk slice or other media storage device
- A portion of a file tree structure that is attached to a mount point on the main file tree so that the files are accessible

Usually, you can tell from the context which meaning is intended.

The Solaris operating environment uses the *virtual file system* (VFS) architecture, which provides a standard interface for different file system types. The VFS architecture enables the kernel to handle basic operations, such as reading, writing, and listing files, and makes it easier to add new file systems.

Types of File Systems

The Solaris operating environment supports three types of file systems:

- Disk-based
- Network-based
- Virtual

To identify the file system type, see "Determining a File System's Type" on page 518.

Disk-Based File Systems

Disk-based file systems are stored on physical media such as hard disks, CD-ROMs, and diskettes. Disk-based file systems can be written in different formats. The available formats are the following:

Disk-Based File System	Format Description
UFS	UNIX file system (based on the BSD Fast File system that was provided in the 4.3 Tahoe release). UFS is the default disk-based file system for the Solaris operating environment. Before you can create a UFS file system on a disk, you must format the disk and divide it into slices. For information on formatting disks and dividing disks into slices, see Chapter 32.
HSFS	High Sierra, Rock Ridge, and ISO 9660 file system. High Sierra is the first CD-ROM file system. ISO 9660 is the official standard version of the High Sierra File System. The HSFS file system is used on CD-ROMs, and is a read-only file system. Solaris HSFS supports Rock Ridge extensions to ISO 9660, which, when present on a CD-ROM, provide all UFS file system features and file types, except for writability and hard links.
PCFS	PC file system, which allows read and write access to data and programs on DOS-formatted disks that are written for DOS-based personal computers.
UDF	The Universal Disk Format (UDF) file system, the industry-standard format for storing information on the optical media technology called DVD (Digital Versatile Disc or Digital Video Disc).

Each type of disk-based file system is customarily associated with a particular media device, as follows:

- UFS with hard disk

- HSFS with CD-ROM
- PCFS with diskette
- UDF with DVD

These associations are not, however, restrictive. For example, CD-ROMs and diskettes can have UFS file systems created on them.

Network-Based File Systems

Network-based file systems can be accessed from the network. Typically, network-based file systems reside on one system, typically a server, and are accessed by other systems across the network.

With NFS, you can administer distributed *resources* (files or directories) by exporting them from a server and mounting them on individual clients. For more information, see "The NFS Environment" on page 517.

Virtual File Systems

Virtual file systems are memory-based file systems that provide access to special kernel information and facilities. Most virtual file systems do not use file system disk space. However, the CacheFS file system uses a file system on the disk to contain the cache. Also, some virtual file systems, such as the temporary file system (TMPFS), use the swap space on a disk.

The CacheFS File System

The CacheFS™ file system can be used to improve performance of remote file systems or slow devices such as CD-ROM drives. When a file system is cached, the data that is read from the remote file system or CD-ROM is stored in a cache on the local system.

If you want to improve the performance and scalability of an NFS or CD-ROM file system, you should use the CacheFS file system. The CacheFS software is a general purpose caching mechanism for file systems that improves NFS server performance and scalability by reducing server and network load.

Designed as a layered file system, the CacheFS software provides the ability to cache one file system on another. In an NFS environment, CacheFS software increases the client per server ratio, reduces server and network loads, and improves performance for clients on slow links, such as Point-to-Point Protocol (PPP). You can also combine a CacheFS file system with the AutoFS service to help boost performance and scalability.

For detailed information about the CacheFS file system, see Chapter 41.

The Universal Disk Format (UDF) File System

The UDF file system is the industry-standard format for storing information on the *DVD* (Digital Versatile Disc or Digital Video Disc) optical media.

The UDF file system is provided as dynamically loadable, 32–bit and 64–bit modules, with system administration utilities for creating, mounting, and checking the file system on both SPARC and x86 platforms. The Solaris UDF file system works with supported ATAPI and SCSI DVD drives, CD-ROM devices, and disk and diskette drives. In addition, the Solaris UDF file system is fully compliant with the UDF 1.50 specification.

The UDF file system provides the following features:

- Ability to access the industry standard CD-ROM and DVD-ROM media when they contain a UDF file system
- Flexibility in exchanging information across platforms and operating systems
- A mechanism for implementing new applications rich in broadcast-quality video, high-quality sound along with the richness in interactivity using the DVD video specification based on UDF format

The following features are not included in the UDF file system:

- Support for write-once media, CD-RW, and DVD-RAM, with either the sequential disk-at-once and incremental recording
- UFS components such as quotas, ACLs, transaction logging, file system locking, and file system threads, which are not part of the UDF 1.50 specification

The UDF file system requires the following:

- The Solaris 7 11/99, Solaris 8, or Solaris 9 releases
- Supported SPARC or x86 platforms
- Supported CD-ROM or DVD-ROM device

The Solaris UDF file system implementation provides:

- Support for industry-standard read/write UDF version 1.50
- Fully internationalized file system utilities

Temporary File System

The temporary file system (TMPFS) uses local memory for file system reads and writes, which is typically much faster than a UFS file system. Using TMPFS can improve system performance by saving the cost of reading and writing temporary files to a local disk or across the network. For example, temporary files are created when you compile a program, and the operating system generates a lot of disk activity or network activity while manipulating these files. Using TMPFS to hold these temporary files can significantly speed up their creation, manipulation, and deletion.

Files in TMPFS file systems are not permanent. They are deleted when the file system is unmounted and when the system is shut down or rebooted.

TMPFS is the default file system type for the `/tmp` directory in the Solaris operating environment. You can copy or move files into or out of the `/tmp` directory, just as you would in a UFS file system.

The TMPFS file system uses swap space as a temporary backing store. If a system with a TMPFS file system does not have adequate swap space, two problems can occur:

- The TMPFS file system can run out of space, just as regular file systems do.
- Because TMPFS allocates swap space to save file data (if necessary), some programs might not execute because of insufficient swap space.

For information about creating TMPFS file systems, see Chapter 39. For information about increasing swap space, see Chapter 42.

The Loopback File System

The loopback file system (LOFS) lets you create a new virtual file system so that you can access files by using an alternative path name. For example, you can create a loopback mount of root `(/)` on `/tmp/newroot`, which will make the entire file system hierarchy look like it is duplicated under `/tmp/newroot`, including any file systems mounted from NFS servers. All files will be accessible either with a path name starting from root (`/`), or with a path name that starts from `/tmp/newroot`.

For information on how to create LOFS file systems, see Chapter 39.

Process File System

The process file system (PROCFS) resides in memory and contains a list of active processes, by process number, in the `/proc` directory. Information in the `/proc` directory is used by commands like `ps`. Debuggers and other development tools can also access the address space of the processes by using file system calls.

Caution – Do not delete the files in the `/proc` directory. The deletion of processes from the `/proc` directory does not kill them. Remember, `/proc` files do not use disk space, so there is little reason to delete files from this directory.

The `/proc` directory does not require administration.

Additional Virtual File Systems

These additional types of virtual file systems are listed for your information. They do not require administration.

Virtual File System	Description
FIFOFS (first-in first-out)	Named pipe files that give processes common access to data
FDFS (file descriptors)	Provides explicit names for opening files using file descriptors
MNTFS	Provides read-only access to the table of mounted file systems for the local system
NAMEFS	Used mostly by STREAMS for dynamic mounts of file descriptors on top of files
SPECFS (special)	Provides access to character special devices and block devices
SWAPFS	Used by the kernel for swapping

Commands for File System Administration

Most commands for file system administration have both a generic component and a file system–specific component. Whenever possible, you should use the generic commands, which call the file system–specific component. The following table lists the generic commands for file system administration, which are located in the `/usr/sbin` directory.

TABLE 38–1 Generic Commands for File System Administration

Command	Man Page	Description
`clri`	`clri`(1M)	Clears inodes
`df`	`df`(1M)	Reports the number of free disk blocks and files
`ff`	`ff`(1M)	Lists file names and statistics for a file system
`fsck`	`fsck`(1M)	Checks the integrity of a file system and repairs any damage found

TABLE 38–1 Generic Commands for File System Administration *(Continued)*

Command	Man Page	Description
`fsdb`	`fsdb`(1M)	Debugs the file system
`fstyp`	`fstyp`(1M)	Determines the file system type
`labelit`	`labelit`(1M)	Lists or provides labels for file systems when they are copied to tape (for use by the `volcopy` command only)
`mkfs`	`mkfs`(1M)	Creates a new file system
`mount`	`mount`(1M)	Mounts local and remote file systems
`mountall`	`mountall`(1M)	Mounts all file systems that are specified in the virtual file system table (`/etc/vfstab`)
`ncheck`	`ncheck`(1M)	Generates a list of path names with their inode numbers
`umount`	`mount`(1M)	Unmounts local and remote file systems
`umountall`	`mountall`(1M)	Unmounts all file systems that are specified in a virtual file system table (`/etc/vfstab`)
`volcopy`	`volcopy`(1M)	Creates an image copy of a file system

How File System Commands Determine the File System Type

The generic file system commands determine the file system type by following this sequence:

1. From the `-F` option, if supplied.
2. By matching a special device with an entry in the `/etc/vfstab` file (if *special* is supplied). For example, `fsck` first looks for a match against the `fsck device` field. If no match is found, it then checks the *special* device field.
3. By using the default specified in the `/etc/default/fs` file for local file systems and in the `/etc/dfs/fstypes` file for remote file systems.

Manual Pages for Generic and Specific Commands

Both the generic commands and specific commands have manual pages in the *man Pages(1M): System Administration Commands*. The manual page for the generic file system commands provide information about generic command options only. The manual page for a specific file system command has specific information about

options for that file system. To look at a specific manual page, append an underscore and the abbreviation for the file system type to the generic command name. For example, to see the specific manual page for mounting a UFS file system, type the following:

```
$ man mount_ufs
```

The Default Solaris File Systems

The Solaris UFS file system is hierarchical, starting with the root directory (/) and continuing downwards through a number of directories. The Solaris installation process enables you to install a default set of directories and uses a set of conventions to group similar types of files together. The following table provides a summary of the default Solaris file systems.

TABLE 38–2 The Default Solaris File Systems

File System or Directory	File System Type	Description
root (/)	UFS	The top of the hierarchical file tree. The root directory contains the directories and files that are critical for system operation, such as the kernel, the device drivers, and the programs used to boot the system. The root directory also contains the mount point directories where local and remote file systems can be attached to the file tree.
/usr	UFS	System files and directories that can be shared with other users. Files that run only on certain types of systems are in the /usr file system (for example, SPARC executables). Files that can be used on all types of systems, such as the man pages, are in the /usr/share directory.
/export/home or /home	NFS, UFS	The mount point for users' home directories, which store user work files. By default the /home directory is an automounted file system. On standalone systems, the /home directory might be a UFS file system on a local disk slice.
/var	UFS	System files and directories that are likely to change or grow over the life of the local system. These include system logs, vi and ex backup files, and uucp files.

TABLE 38–2 The Default Solaris File Systems *(Continued)*

File System or Directory	File System Type	Description
`/opt`	NFS, UFS	Optional mount point for third-party software. On some systems, the `/opt` directory might be a UFS file system on a local disk slice.
`/tmp`	TMPFS	Temporary files, which are cleared each time the system is booted or the `/tmp` file system is unmounted.
`/proc`	PROCFS	A list of active processes, by number.
`/etc/mnttab`	MNTFS	A file system that provides read-only access to the table of mounted file systems for the local system.
`/var/run`	TMPFS	A file system for storing temporary files that are not needed after the system is booted.

The root (`/`) and `/usr` file systems are needed to run a system. Some of the most basic commands in the `/usr` file system (like `mount`) are included in the root (`/`) file system so that they are available when the system boots or is in single-user mode and `/usr` is not mounted. For more detailed information on the default directories for the root (`/`) and `/usr` file systems, see Chapter 44.

Swap Space

The Solaris operating environment uses some disk slices for temporary storage rather than for file systems. These slices are called *swap* slices, or *swap space*. Swap space is used as virtual memory storage areas when the system does not have enough physical memory to handle current processes.

Since many applications rely on swap space, you should know how to plan for, monitor, and add more swap space when needed. For an overview about swap space and instructions for adding swap space, see Chapter 42.

The UFS File System

UFS is the default disk-based file system in Solaris operating environment. Most often, when you administer a disk-based file system, you will be administering UFS file systems. UFS provides the following features:

UFS Feature	Description
State flags	Show the state of the file system: clean, stable, active, logging, or unknown. These flags eliminate unnecessary file system checks. If the file system is “clean,” “stable,” or “logging,” file system checks are not run.
Extended fundamental types (EFT)	Provides 32-bit user ID (UID), group ID (GID), and device numbers.
Large file systems	Allows 1 Tbyte (terabyte) files in a UFS file system. The Solaris operating environment does not provide striping, which is required to make a logical slice large enough for a 1-Tbyte file system. However, Solaris Volume Manager provides this capability.
Large files	Allows 2 Gbytes (gigabytes) files in a UFS file system by default. You must explicitly use the `nolargefiles` mount option to enforce a 2–Gbyte maximum file size limit.

For detailed information about the UFS file system structure, see Chapter 44.

UFS Logging

UFS logging is the process of storing transactions (changes that make up a complete UFS operation) in a log before the transactions are applied to the UFS file system. Once a transaction is stored, the transaction can be applied to the file system later.

At reboot, the system discards incomplete transactions, but applies the transactions for completed operations. The file system remains consistent because only completed transactions are ever applied. This consistency remains even when a system crashes, which normally interrupts system calls and introduces inconsistencies into a UFS file system.

UFS logging provides two advantages:

- Prevents file systems from becoming inconsistent, therefore eliminating the need to run the `fsck` command. And, because `fsck` checking can be bypassed, UFS logging reduces the time required to reboot a system if it crashes, or after an unclean halt (see “What the `fsck` Command Checks and Tries to Repair” on page 588 for details on unclean halts).
- UFS logging can significantly reduce the boot time on systems with large file systems, which usually take a long time to read and verify with the `fsck` command.

The log created by UFS logging is continually flushed as it fills up. The log is flushed when the file system is unmounted or as a result of the `lockfs -f` command.

UFS logging is not enabled by default. To enable UFS logging, you must specify the `-o logging` option with the `mount` command in the `/etc/vfstab` file or when you mount the file system manually. The log is allocated from free blocks on the file system, and it is sized at approximately 1 Mbyte per 1 Gbyte of file system, up to a maximum of 64 Mbytes. Logging can be enabled on any UFS file system, including the root (`/`) file system. Also, the `fsdb` command now has new debugging commands to support UFS logging.

Planning UFS File Systems

When laying out file systems, you need to consider possible conflicting demands. Here are some suggestions:

- Distribute the work load as evenly as possible among different I/O systems and disk drives. Distribute the `/export/home` file system and swap space evenly across disks.
- Keep pieces of projects or members of groups within the same file system.
- Use as few file systems per disk as possible. On the system (or boot) disk, you should have three file systems: root (`/`), `/usr`, and swap space. On other disks, create one or, at most, two file systems; one being additional swap space, preferably. Fewer, roomier file systems cause less file fragmentation than many small, over-crowded file systems. Higher-capacity tape drives and the ability of the `ufsdump` command to handle multiple volumes make it easier to back up larger file systems.
- If you have some users who consistently create very small files, consider creating a separate file system with more inodes. However, most sites do not need to keep similar types of user files in the same file system.

For information on default file system parameters as well as procedures for creating new UFS file systems, see Chapter 39.

UFS Direct Input/Output (I/O)

Direct I/O is intended to boost bulk I/O operations. Bulk I/O operations use large buffer sizes to transfer large files (larger than 256 Kbytes).

Using UFS direct I/O might benefit applications, such as database engines, that do their own internal buffering. Starting with the Solaris 8 1/01 release, UFS direct I/O has been enhanced to allow the same kind of I/O concurrency seen when accessing raw devices. Now you can get the benefit of file system naming and flexibility with very little performance penalty. Check with your database vendor to see if they can enable UFS direct I/O in their product configuration options.

Direct I/O can also be enabled on a file system by using the `forcedirectio` option to the `mount` command. Enabling direct I/O is a performance benefit only when a file system is transferring large amounts of sequential data.

When a file system is mounted with this option, data is transferred directly between a user's address space and the disk. When forced direct I/O is not enabled for a file system, data transferred between a user's address space and the disk is first buffered in the kernel address space.

The default behavior is no forced direct I/O on a UFS file system. For more information, see `mount_ufs`(1M).

Mounting and Unmounting File Systems

Before you can access the files on a file system, you need to mount the file system. When you mount a file system, you attach that file system to a directory (*mount point*) and make it available to the system. The root (/) file system is always mounted. Any other file system can be connected or disconnected from the root (/) file system.

When you mount a file system, any files or directories in the underlying mount point directory are unavailable as long as the file system is mounted. These files are not permanently affected by the mounting process, and they become available again when the file system is unmounted. However, mount directories are typically empty, because you usually do not want to obscure existing files.

For example, the following figure shows a local file system, starting with a root (/) file system and the `sbin`, `etc`, and `opt` subdirectories.

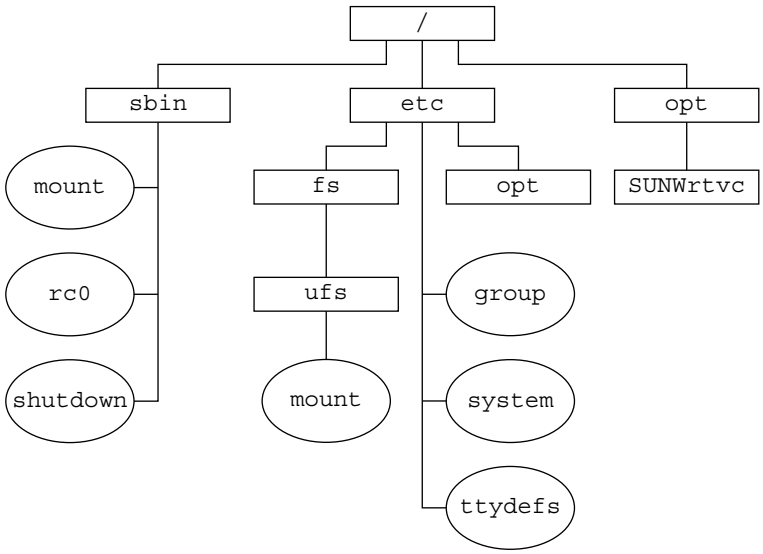


FIGURE 38–1 Sample root (/) File System

To access a local file system from the `/opt` file system that contains a set of unbundled products, you must do the following:

- First, you must create a directory to use as a mount point for the file system you want to mount, for example, `/opt/unbundled`.
- Once the mount point is created, you can mount the file system (by using the `mount` command), which makes all of the files and directories in `/opt/unbundled` available, as shown in the following figure.

For step-by-step instructions on how to mount file systems, see Chapter 40.

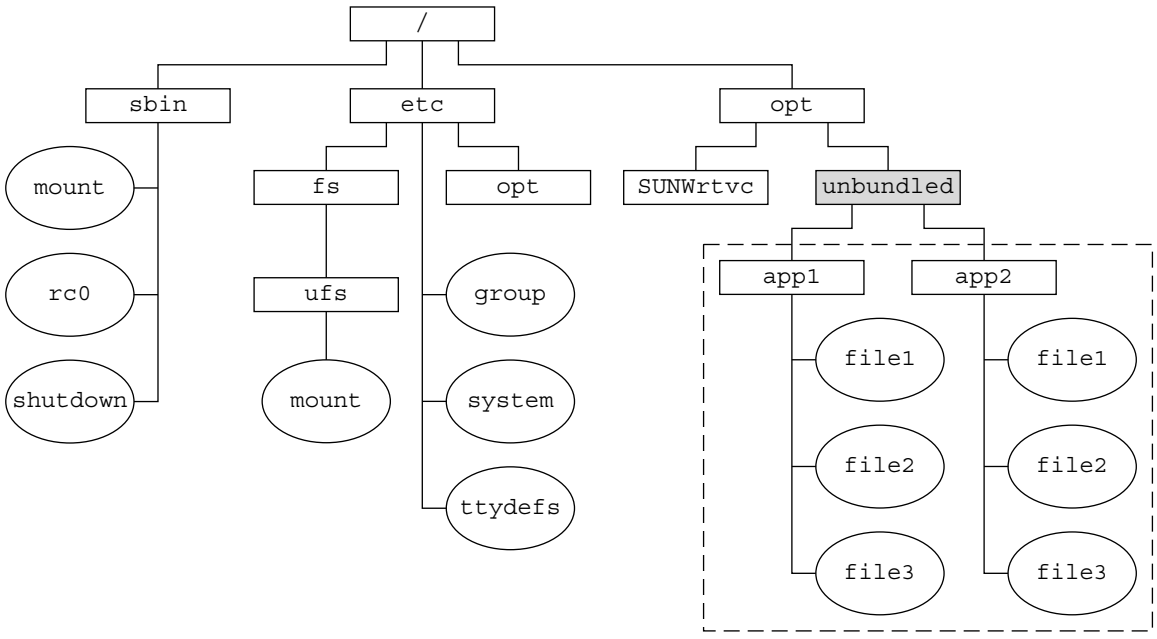


FIGURE 38–2 Mounting a File System

The Mounted File System Table

Whenever you mount or unmount a file system, the `/etc/mnttab` (mount table) file is modified with the list of currently mounted file systems. You can display the contents of this file with the `cat` or `more` commands, but you cannot edit it. Here is an example of an `/etc/mnttab` file:

```
$ more /etc/mnttab
/dev/dsk/c0t0d0s0  /  ufs rw,intr,largefiles,onerror=panic,suid,dev=2200000 938557523
/proc   /proc   proc    dev=3180000     938557522
fd      /dev/fd fd      rw,suid,dev=3240000     938557524
mnttab  /etc/mnttab     mntfs   dev=3340000     938557526
swap    /var/run        tmpfs   dev=1   938557526
swap    /tmp    tmpfs   dev=2   938557529
/dev/dsk/c0t0d0s7 /export/home ufs rw,intr,largefiles,onerror=panic,suid,dev=2200007 ...
$
```

The Virtual File System Table

It would be a very time-consuming and error-prone task to manually mount file systems every time you wanted to access them. To avoid this problem, the virtual file system table (the `/etc/vfstab` file) provides a list of file systems and how to mount them.

The `/etc/vfstab` file provides two important features:

- You can specify file systems to automatically mount when the system boots.
- You can mount file systems by using only the mount point name, because the `/etc/vfstab` file contains the mapping between the mount point and the actual device slice name.

A default `/etc/vfstab` file is created when you install a system, depending on the selections you make when installing system software. However, you can edit the `/etc/vfstab` file on a system whenever you want. To add an entry, the main information you need to specify is the device where the file system resides, the name of the mount point, the type of the file system, whether you want the file system to mount automatically when the system boots (by using the `mountall` command), and any mount options.

The following is an example of an `/etc/vfstab` file. Comment lines begin with #. This example shows an `/etc/vfstab` file for a system with two disks (`c0t0d0` and `c0t3d0`).

```
$ more /etc/vfstab
#device         device          mount           FS      fsck   mount   mount
#to mount       to fsck         point           type    pass   at boot options
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 /         ufs     1       no      -
/proc           -               /proc           proc    -       no      -
/dev/dsk/c0t0d0s1 -                -            swap    -       no      -
swap            -               /tmp            tmpfs   -       yes     -
/dev/dsk/c0t0d0s6 /dev/rdsk/c0t0d0s6 /usr       ufs     2       no      -
/dev/dsk/c0t3d0s7 /dev/rdsk/c0t3d0s7 /test      ufs     2       yes     -
$
```

In the preceding example, the last entry specifies that a UFS file system on the `/dev/dsk/c0t3d0s7` slice will be automatically mounted on the `/test` mount point when the system boots. Note that, for root (`/`) and `/usr`, the `mount at boot` field value is specified as `no`, because these file systems are mounted by the kernel as part of the boot sequence before the `mountall` command is run.

For descriptions of each of the `/etc/vfstab` fields and information on how to edit and use the file, see Chapter 40.

The NFS Environment

NFS is a distributed file system service that can be used to share *resources* (files or directories) from one system, typically a server, with other systems on the network. For example, you might want to share third-party applications or source files with users on other systems.

NFS makes the actual physical location of the resource irrelevant to the user. Instead of placing copies of commonly used files on every system, NFS allows you to place one copy on one system's disk and let all other systems access it from the network. Under NFS, remote files are virtually indistinguishable from local ones.

A system becomes an NFS server if it has resources to share on the network. A server keeps a list of currently shared resources and their access restrictions (such as read/write or read-only access).

When you share a resource, you make it available for mounting by remote systems.

You can share a resource in these ways:

- By using the `share` or `shareall` command
- By adding an entry to the `/etc/dfs/dfstab` (distributed file system table) file and rebooting the system

For information on how to share resources, see Chapter 40. For a complete description of NFS, see "Managing Network File Systems (Overview)" in *System Administration Guide: Resource Management and Network Services*.

Automounting or AutoFS

You can mount NFS file system resources by using a client-side service called automounting (or AutoFS), which enables a system to automatically mount and unmount NFS resources whenever you access them. The resource remains mounted as long as you remain in the directory and are using a file. If the resource is not accessed for a certain period of time, it is automatically unmounted.

AutoFS provides the following features:

- NFS resources don't need to be mounted when the system boots, which saves booting time.
- Users don't need to know the root password to mount and unmount NFS resources.
- Network traffic might be reduced, since NFS resources are only mounted when they are in use.

The AutoFS service is initialized by the `automount` utility, which runs automatically when a system is booted. The `automountd` daemon runs continuously and is responsible for the mounting and unmounting of the NFS file systems on an as-needed basis. By default, the `/home` file system is is mounted by the `automount` daemon.

With AutoFS, you can specify multiple servers to provide the same file system. This way, if one of the servers is down, AutoFS can try to mount from another machine.

For complete information on how to set up and administer AutoFS, see *System Administration Guide: IP Services*.

Determining a File System's Type

You can determine a file system's type by using the following:

- The `FS type` field in the virtual file system table (`/etc/vfstab` file)
- The `/etc/default/fs` file for local file systems
- The `/etc/dfs/fstypes` file for NFS file systems

▼ How to Determine a File System's Type

This procedure works whether the file system is mounted or not.

Determine a file system's type by using the `grep` command.

```
$ grep mount-point fs-table
```

mount-point	Specifies the mount point name of the file system for which you want to know the file system type. For example, the `/var` directory.
fs-table	Specifies the absolute path to the file system table in which to search for the file system's type. If the file system is mounted, *fs-table* should be `/etc/mnttab`. If the file system isn't mounted, *fs-table* should be `/etc/vfstab`.

Information for the mount point is displayed.

Note – If you have the raw device name of a disk slice, you can use the `fstyp` command to determine a file system's type (if the disk slice contains a file system). For more information, see `fstyp`(1M).

Example—Determining a File System's Type

The following example uses the `/etc/vfstab` file to determine the type of the `/export` file system.

```
$ grep /export /etc/vfstab
/dev/dsk/c0t3d0s6   /dev/rdsk/c0t3d0s6  /export ufs   2       yes    -
$
```

The following example uses the `/etc/mnttab` file to determine the file system type of the currently mounted diskette (which was mounted by `vold`).

```
$ grep /floppy /etc/mnttab
/vol/dev/diskette0/unnamed_floppy   /floppy/unnamed_floppy  pcfs rw,
nohidden,nofoldcase,dev=16c0009      89103376
$
```

CHAPTER 39

Creating File Systems (Tasks)

This chapter describes how to create UFS, temporary (TMPFS), and loopback (LOFS) file systems. For UFS file systems, this chapter shows you how to create a file system on a hard disk by using the `newfs` command. Because TMPFS and LOFS are virtual file systems, you actually "access" them by mounting them.

This is a list of the step-by-step instructions in this chapter.

- "How to Create a UFS File System" on page 522
- "How to Create a TMPFS File System" on page 524
- "How to Create an LOFS File System" on page 526

Note – For instructions on how to create UFS and DOS file systems on removable media, see Chapter 17.

Creating a UFS File System

Before you can create a UFS file system on a disk, the disk must be formatted and divided into slices. A disk slice is a physical subset of a disk that is composed of a single range of contiguous blocks. A slice can be used either as a raw device that provides, for example, swap space, or to hold a disk-based file system. See Chapter 32 for complete information on formatting disks and dividing disks into slices.

Volume management products, like Solaris Volume Manager, create more sophisticated *volumes*, that expand beyond single slice or single disk boundaries. For more information about using volumes, see *Solaris Volume Manager Administration Guide*.

Note – Solaris device names use the term slice (and the letter `s` in the device name) to refer to the slice number. Slices are also called "partitions."

You need to create UFS file systems only occasionally, because the Solaris operating environment automatically creates them as part of the installation process. You need to create (or re-create) a UFS file system when you want to do the following:

- Add or replace disks
- Change the existing partitioning structure
- Do a full restoration of a file system

The `newfs` command is the standard way to create UFS file systems. The `newfs` command is a convenient front-end to the `mkfs` command, which actually creates the new file system. The `newfs` command reads parameter defaults, such as tracks per cylinder and sectors per track, from the disk label that will contain the new file system. The options you choose are passed to the `mkfs` command to build the file system.

Default Parameters for the `newfs` Command

To make a new file system on a disk slice, you almost always use the `newfs` command. The following table shows the default parameters that are used by the `newfs` command.

Parameter	Default Value
Block size	8 Kbytes
Fragment size	1 Kbyte
Minimum free space	((64 Mbytes/partition size) * 100), rounded down to the nearest integer and limited to between 1% and 10%, inclusively
Rotational delay	Zero
Optimization type	Time
Number of inodes	1 inode for each 2 Kbytes of data space

▼ How to Create a UFS File System

1. **Make sure you have met the following prerequisites:**

   a. **The disk must be formatted and divided into slices.**

For information on formatting disks and dividing disks into slices, see Chapter 32.

b. You need to know the device name of the slice that will contain the file system.
For information on finding disks and disk slice numbers, see Chapter 33.

c. If you are re-creating an existing UFS file system, unmount it.

d. You must be superuser or assume an equivalent role.

2. Create the UFS file system.

```
# newfs [-N] [-b size] [-i bytes] /dev/rdsk/device-name
```

`-N`	Displays what parameters the `newfs` command would pass to the `mkfs` command without actually creating the file system. This option is a good way to test the `newfs` command.
`-b` *size*	Specifies the block size for the file system, either 4096 or 8192 bytes per block. The default is 8192.
`-i` *bytes*	Specifies the number of bytes per inode. The default varies depending on the disk size. For more information, see `newfs`(1M).
device-name	Specifies the disk device name on which to create the new file system.

The system asks for confirmation.

Caution – Be sure you have specified the correct device name for the slice before performing this step. If you specify the wrong slice, you will erase its contents when the new file system is created. This error might cause the system to panic.

3. To verify the creation of the UFS file system, check the new file system.

```
# fsck /dev/rdsk/device-name
```

The *device-name* argument specifies the name of the disk device that contains the new file system.

The `fsck` command checks the consistency of the new file system, reports any problems, and prompts you before it repairs the problems. For more information on the `fsck` command, see Chapter 43 or `fsck`(1M).

Example—Creating a UFS File System

The following example shows how to create a UFS file system on `/dev/rdsk/c0t1d0s7`.

```
# newfs /dev/rdsk/c0t1d0s7
/dev/rdsk/c0t1d0s7:  725760 sectors in 720 cylinders of 14 tracks, 72 sectors
        354.4MB in 45 cyl groups (16 c/g, 7.88MB/g, 3776 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 16240, 32448, 48656, 64864, 81072, 97280, 113488, 129696, 145904, 162112,
 178320, 194528, 210736, 226944, 243152, 258080, 274288, 290496, 306704,
 322912, 339120, 355328, 371536, 387744, 403952, 420160, 436368, 452576,
 468784, 484992, 501200, 516128, 532336, 548544, 564752, 580960, 597168,
 613376, 629584, 645792, 662000, 678208, 694416, 710624,
#
```

Where to Go From Here

To mount the UFS file system and make it available, go to Chapter 40.

Creating a Temporary File System (TMPFS)

A temporary file system (TMPFS) uses local memory for file system reads and writes, which is typically much faster than reads and writes in a UFS file system. TMPFS file systems can improve system performance by saving the cost of reading and writing temporary files to a local disk or across the network. Files in TMPFS file systems do not survive across reboots or unmounts.

If you create multiple TMPFS file systems, be aware that they all use the same system resources. Files created under one TMPFS file system use up the space available for any other TMPFS file system, unless you limit TMPFS sizes by using the `-o size` option of the `mount` command.

For more information, see the `tmpfs`(7FS).

▼ How to Create a TMPFS File System

1. **Become superuser or assume an equivalent role.**
2. **Create the directory that you want to mount as the TMPF file system, if necessary.**

   # **mkdir** */mount-point*

   mount-point is the directory on which the TMPFS file system is mounted.
3. **Mount the TMPFS file system.**

   # **mount -F tmpfs** [**-o size=***number*] **swap** *mount-point*

`-o size=`*number*	Specifies the size limit of the TMPFS file system in Mbytes.
mount-point	Specifies the directory on which the TMPFS file system is mounted.

To set up the system to automatically mount a TMPFS file system when it boots, see "Example—Mounting a TMPFS File System at Boot Time" on page 525.

4. **Verify that the TMPFS file system has been created.**

```
# mount -v
```

Example—Creating a TMPFS File System

The following example shows how to create, mount, and limit the size of the TMPFS file system, `/export/reports`, to 50 Mbytes.

```
# mkdir /export/reports
# chmod 777 /export/reports
# mount -F tmpfs -o size=50 swap /export/reports
```

Example—Mounting a TMPFS File System at Boot Time

You can set up the system to automatically mount a TMPFS file system when it boots by adding an `/etc/vfstab` entry. The following example shows an entry in the `/etc/vfstab` file that mounts `/export/test` as a TMPFS file system when the system boots. Since the `size=`*number* option is not specified, the size of the TMPFS file system on `/export/test` is limited only by the available system resources.

```
swap - /export/test  tmpfs   -  yes  -
```

For more information on the `/etc/vfstab` file, see "Field Descriptions for the `/etc/vfstab` File" on page 532.

Creating a Loopback File System (LOFS)

A LOFS file system is a virtual file system that provides an alternate path to an existing file system. When other file systems are mounted onto an LOFS file system, the original file system does not change.

For more information, see the `lofs`(7FS).

Caution – Be careful when creating LOFS file systems. Because LOFS file systems are virtual file systems, the potential for confusing both users and applications is enormous.

▼ How to Create an LOFS File System

1. **Become superuser or assume an equivalent role.**
2. **Create the directory you want to mount as an LOFS file system, if necessary.**

   # **mkdir** *loopback-directory*
3. **Grant the appropriate permissions and ownership on the newly created directory.**
4. **Create the mount point where you want to mount the LOFS file system, if necessary.**

   # **mkdir** */mount-point*
5. **Mount the LOFS file system.**

   # **mount -F lofs** *loopback-directory /mount-point*

loopback-directory	Specifies the file system to be mounted on the loopback mount point.
/mount-point	Specifies the directory on which to mount the LOFS file system.

6. **Verify that the LOFS file system has been mounted.**

   # **mount -v**

Example—Creating and Mounting an LOFS File System

The following example illustrates how to create, mount, and test new software in the `/new/dist` directory as a loopback file system without actually having to install it.

```
# mkdir /tmp/newroot
# mount -F lofs /new/dist /tmp/newroot
# chroot /tmp/newroot newcommand
```

Example—Mounting an LOFS File System at Boot Time

You can set up the system to automatically mount an LOFS file system when it boots by adding an entry to the end of the `/etc/vfstab` file. The following example shows an entry in the `/etc/vfstab` file that mounts an LOFS file system for the root (`/`) file system on `/tmp/newroot`.

```
/ - /tmp/newroot  lofs   -  yes  -
```

Caution – Make sure the loopback entries are the last entries in the `/etc/vfstab` file. Otherwise, if the `/etc/vfstab` entry for a loopback file system precedes the file systems to be included in it, the loopback file system cannot be mounted.

For more information on the `/etc/vfstab` file, see "Field Descriptions for the `/etc/vfstab` File" on page 532.

CHAPTER 40

Mounting and Unmounting File Systems (Tasks)

This chapter describes how to mount and unmount file systems.

This is a list of the step-by-step instructions in this chapter.

- "How to Determine Which File Systems Are Mounted" on page 534
- "How to Add an Entry to the `/etc/vfstab` File" on page 535
- "How to Mount a File System (`/etc/vfstab` File)" on page 536
- "How to Mount a UFS File System (`mount` Command)" on page 537
- "How to Mount a UFS File System Without Large Files (`mount` Command)" on page 538
- "How to Mount an NFS File System (`mount` Command)" on page 539
- "x86: How to Mount a PCFS (DOS) File System From a Hard Disk (`mount` Command)" on page 540
- "How to Verify a File System is Unmounted" on page 542
- "How to Stop All Processes Accessing a File System" on page 542
- "How to Unmount a File System" on page 543

Overview of Mounting File Systems

After you create a file system, you need to make it available to the system so you can use it. You make a file system available by mounting it, which attaches the file system to the system directory tree at the specified mount point. The root (`/`) file system is always mounted.

The following table provides guidelines on mounting file systems based on how you use them.

Mount Type Needed	Suggested Mount Method
Local or remote file systems that need to be mounted infrequently	The `mount` command that you enter manually from the command line.
Local file systems that need to be mounted frequently	The `/etc/vfstab` file, which mounts the file system automatically when the system is booted in multi-user state.
Remote file systems that need to be mounted frequently, such as home directories	■ The `/etc/vfstab` file, which automatically mounts the file system when the system is booted in multi-user state. ■ AutoFS, which automatically mounts or unmounts the file system when you access or change out of the directory. To enhance performance, you can also cache the remote file systems by using the CacheFS file system.

You can mount media that contains a file system by inserting the media into the drive and running the `volcheck` command if necessary. For more information on mounting removable media, see Chapter 17.

Commands for Mounting and Unmounting File Systems

The following table lists the commands in the `/usr/sbin` directory that you use to mount and unmount file systems.

TABLE 40–1 Commands for Mounting and Unmounting File Systems

Command	Man Page	Description
`mount`	`mount`(1M)	Mounts file systems and remote resources.
`mountall`	`mountall`(1M)	Mounts all file systems that are specified in the `/etc/vfstab` file. The `mountall` command runs automatically when the system enters multiuser mode.
`umount`	`mount`(1M)	Unmounts file systems and remote resources.
`umountall`	`mountall`(1M)	Unmounts all file systems that are specified in the `/etc/vfstab` file.

The `mount` and `mountall` commands will not mount a read/write file system that has known inconsistencies. If you receive an error message from the `mount` or `mountall` command, you might need to check the file system. See Chapter 43 for information on how to check the file system.

The `umount` and `umountall` commands will not unmount a file system that is busy. A file system is considered busy if one of the following is true:

- A user is accessing a file or directory in the file system.
- If a program has a file open in that file system.
- If the file system is shared.

Commonly Used Mount Options

The following table describes the commonly used options that you can specify with the `mount -o` option. If you specify multiple options, separate them with commas (no spaces). For example, `-o ro,nosuid`.

For a complete list of mount options for each file system type, refer to the specific mount man pages (for example, `mount_ufs`(1M)).

TABLE 40–2 Commonly Used `-o` Mount Options

Option	File System	Description
`bg	fg`	NFS	If the first mount attempt fails, retries in the background (`bg`) or in the foreground (`fg`). This option is safe for non-critical `vfstab` entries. The default is `fg`.
`hard	soft`	NFS	Specifies the procedure if the server does not respond. The `soft` option indicates that an error is returned. The `hard` option indicates that the retry request is continued until the server responds. The default is `hard`.
`intr	nointr`	NFS	Specifies whether keyboard interrupts are delivered to a process that is hung while waiting for a response on a hard-mounted file system. The default is `intr` (interrupts allowed).
`largefiles	nolargefiles`	UFS	Enables you to create files larger than 2 Gbytes. The `largefiles` option means that a file system mounted with this option *might* contain files larger than 2 Gbytes, but it is not required. If the `nolargefiles` option is specified, the file system cannot be mounted on a system that is running Solaris 2.6 or compatible versions. The default is `largefiles`.

TABLE 40–2 Commonly Used `-o` Mount Options *(Continued)*

Option	File System	Description
`logging | nologging`	UFS	Enables or disables logging for the file system. UFS logging is the process of storing transactions (changes that make up a complete UFS operation) into a log before the transactions are applied to the UFS file system. Logging helps prevent UFS file systems from becoming inconsistent, which means `fsck` can be bypassed. Bypassing `fsck` reduces the time to reboot a system if it crashes, or after a system is shutdown uncleanly. The log is allocated from free blocks on the file system, and is sized at approximately 1 Mbyte per 1 Gbyte of file system, up to a maximum of 64 Mbytes. The default is `nologging`.
`atime | noatime`	UFS	Suppresses access time updates on files, except when they coincide with updates to the time of the last file status change or the time of the last file modification. For more information, see `stat`(2). This option reduces disk activity on file systems where access times are unimportant (for example, a Usenet news spool). The default is normal access time (`atime`) recording.
`remount`	All	Changes the mount options associated with an already-mounted file system. This option can generally be used with any option except `ro`, but what can be changed with this option is dependent on the file system type.
`retry=`*n*	NFS	Retries the mount operation when it fails. *n* is the number of times to retry.
`ro | rw`	CacheFS, NFS, PCFS, UFS, HSFS	Specifies read/write (rw) or read-only (ro). If you do not specify this option, the default is rw. The default option for HSFS is `ro`.
`suid | nosuid`	CacheFS, HSFS, NFS, UFS	Allows or disallows `setuid` execution. The default is to allow `setuid` execution.

Field Descriptions for the `/etc/vfstab` File

An entry in the `/etc/vfstab` file has seven fields, which are described in the following table.

TABLE 40–3 Field Descriptions for the /etc/vfstab File

Field Name	Description
device to mount	This field identifies one of the following: ■ The block device name for a local UFS file system (for example, /dev/dsk/c0t0d0s0). ■ The resource name for a remote file system (for example, myserver:/export/home). For more information about NFS, see *System Administration Guide: IP Services*. ■ The block device name of the slice on which to swap (for example, /dev/dsk/c0t3d0s1). ■ A directory for a virtual file system type.
device to fsck	The raw (character) device name that corresponds to the UFS file system identified by the device to mount field (for example, /dev/rdsk/c0t0d0s0). This field determines the raw interface that is used by the fsck command. Use a dash (-) when there is no applicable device, such as for a read-only file system or a remote file system.
mount point	Identifies where to mount the file system (for example, /usr).
FS type	Identifies the type of file system.
fsck pass	The pass number used by the fsck command to decide whether to check a file system. When the field contains a dash (-), the file system is not checked. When the field contains a zero, UFS file systems are not checked but non-UFS file systems are checked. When the field contains a value greater than zero, the file system is always checked. All file systems with a value of 1 in this field are checked one at a time in the order they appear in the vfstab file. When the fsck command is run on multiple UFS file systems that have fsck pass values greater than one and the preen option (-o p) is used, the fsck command automatically checks the file systems on different disks in parallel to maximize efficiency. Otherwise, the value of the pass number does not have any effect.
mount at boot	Set to yes or no for whether the file system should be automatically mounted by the mountall command when the system is booted. Note that this field has nothing to do with AutoFS. The root (/), /usr and /var file systems are not mounted from the vfstab file initially. This field should always be set to no for these file systems and for virtual file systems such as /proc and /dev/fd.
mount options	A list of comma-separated options (with no spaces) that are used for mounting the file system. Use a dash (-) to indicate no options. For a list of commonly used mount options, see Table 40–2.

Note – You must have an entry in each field in the `/etc/vfstab` file. If there is no value for the field, be sure to enter a dash (`-`). Otherwise, the system might not boot successfully. Similarly, white space should not be used in a field value.

Mounting File Systems

The following sections describe how to mount a file system by adding an entry in the `/etc/vfstab` file or by using the `mount` command from the command line.

▼ How to Determine Which File Systems Are Mounted

You can determine which file systems are already mounted by using the `mount` command.

```
$ mount [ -v ]
```

`-v` Displays the list of mounted file systems in verbose mode.

Example—Determining Which File Systems Are Mounted

This example shows how to use the `mount` command to display information about the file systems that are currently mounted.

```
$ mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror= ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/onerror= ...
/proc on /proc read/write/setuid/dev=4300000 on Fri Nov 30 11:25:13 2001
/etc/mnttab on mnttab read/write/setuid/dev=43c0000 on Fri Nov 30 11:25:13 ...
/dev/fd on fd read/write/setuid/dev=4400000 on Fri Nov 30 11:25:17 2001
/var/run on swap read/write/setuid/xattr/dev=1 on Fri Nov 30 11:25:20 2001
/tmp on swap read/write/setuid/xattr/dev=2 on Fri Nov 30 11:25:24 2001
/export on /dev/dsk/c0t0d0s4 read/write/setuid/intr/largefiles/xattr/ ...
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/ ...
$
```

▼ How to Add an Entry to the /etc/vfstab File

1. **Become superuser or assume an equivalent role.**
2. **Create a mount point for the file system to be mounted, if necessary.**

   # **mkdir** */mount-point*

   There must be a mount point on the local system to mount a file system. A mount point is a directory to which the mounted file system is attached.
3. **Edit the /etc/vfstab file and add an entry. Make sure that you do the following:**

   a. **Separate each field with white space (a space or a tab).**

   b. **Enter a dash (-) if a field has no contents.**

   c. **Save the changes.**

   For detailed information about the /etc/vfstab field entries, see Table 40–3.

Note – Since the root (/) file system is mounted read-only by the kernel during the boot process, only the remount option (and options that can be used in conjunction with remount) affect the root (/) entry in the /etc/vfstab file.

Examples—Adding an Entry to the /etc/vfstab File

The following example shows how to mount the disk slice /dev/dsk/c0t3d0s7 as a UFS file system to the mount point directory /files1. The raw character device /dev/rdsk/c0t3d0s7 is specified as the device to fsck. The fsck pass value of 2 means that the file system will be checked, but not sequentially.

```
#device           device             mount    FS       fsck   mount    mount
#to mount         to fsck            point    type     pass   at boot  options
#
/dev/dsk/c0t3d0s7 /dev/rdsk/c0t3d0s7 /files1  ufs      2      yes       -
```

The following example shows how to mount the /export/man directory from the system pluto as an NFS file system on mount point /usr/man. A device to fsck nor a fsck pass is specified because it's an NFS file system. In this example, mount options are ro (read-only) and soft.

```
#device           device             mount    FS       fsck   mount    mount
#to mount         to fsck            point    type     pass   at boot  options
pluto:/export/man   -                /usr/man nfs      -      yes       ro,soft
```

The following example shows how to mount the root (/) file system on a loopback mount point, /tmp/newroot. LOFS file systems must always be mounted after the file systems that are in the LOFS file system.

```
#device           device             mount    FS       fsck   mount    mount
#to mount         to fsck            point    type     pass   at boot  options
#
/                   -                /tmp/newroot lofs -      yes       -
```

▼ How to Mount a File System (`/etc/vfstab` File)

1. **Become superuser or assume an equivalent role.**
2. **Mount a file system listed in the `/etc/vfstab` file.**

   # **mount** */mount-point*

   /mount-point specifies an entry in the `mount point` or `device to mount` field in the `/etc/vfstab` file. It is usually easier to specify the mount point.

Example—Mounting a File System (`/etc/vfstab` File)

The following example shows how to mount the `/usr/dist` file system that is listed in the `/etc/vfstab` file.

```
# mount /usr/dist
```

Examples—Mounting All File Systems (`/etc/vfstab` File)

The following example shows the messages that are displayed if file systems are already mounted when you use the `mountall` command.

```
# mountall
/dev/rdsk/c0t0d0s7 already mounted
mount: /tmp already mounted
mount: /dev/dsk/c0t0d0s7 is already mounted, /export/home is busy,
        or the allowable number of mount points has been exceeded
```

All the file systems with a `device to fsck` entry are checked and fixed, if necessary, before they are mounted.

The following example shows how to mount all the local systems that are listed in the `/etc/vfstab` file.

```
# mountall -l
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror= ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/ ...
/proc on /proc read/write/setuid/dev=38c0000 on Tue Oct 30 15:45:32 2001
/etc/mnttab on mnttab read/write/setuid/dev=3980000 on Tue Oct 30  ...
/dev/fd on fd read/write/setuid/dev=39c0000 on Tue Oct 30 15:45:36 2001
```

```
/var/run on swap read/write/setuid/xattr/dev=1 on Tue Oct 30 15:45:39 ...
/tmp on swap read/write/setuid/xattr/dev=2 on Tue Oct 30 16:05:57 2001
/datab on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr/ ...
```

The following example shows how to mount all of the remote file systems that are listed in the `/etc/vfstab` file.

```
# mountall -r
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror= ...
/usr on /dev/dsk/c0t0d0s6 read/write/setuid/intr/largefiles/xattr/onerror= ...
/proc on /proc read/write/setuid/dev=38c0000 on Tue Oct 30 15:45:32 2001
/etc/mnttab on mnttab read/write/setuid/dev=3980000 on Tue Oct 30 ...
/dev/fd on fd read/write/setuid/dev=39c0000 on Tue Oct 30 15:45:36 2001
/var/run on swap read/write/setuid/xattr/dev=1 on Tue Oct 30 15:45:39 2001
/tmp on swap read/write/setuid/xattr/dev=2 on Tue Oct 30 16:05:57 2001
/datab on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr/ ...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/xattr ...
```

▼ How to Mount a UFS File System (`mount` Command)

1. **Become superuser or assume an equivalent role.**
2. **Create a mount point for the file system to be mounted, if necessary.**

   # **mkdir** */mount-point*

   There must be a mount point on the local system to mount a file system. A mount point is a directory to which the mounted file system is attached.
3. **Mount the UFS file system.**

   # **mount** [**-o** *mount-options*] **/dev/dsk/***device-name* */mount-point*

`-o` *mount-options*	Specifies mount options that you can use to mount a UFS file system. For a list of options, see Table 40–2 or `mount_ufs`(1M).
`/dev/dsk/`*device-name*	Specifies the disk device name for the slice that contains the file system (for example, `/dev/dsk/c0t3d0s7`). To get slice information for a disk, see "How to Display Disk Slice Information" on page 441.
/mount-point	Specifies the directory on which to mount the file system.

Example—Mounting a UFS File System (`mount` Command)

The following example shows how to mount `/dev/dsk/c0t3d0s7` on the `/files1` directory.

```
# mount /dev/dsk/c0t3d0s7 /files1
```

Example—Mounting a UFS File System With Logging Enabled (`mount` Command)

UFS logging eliminates file system inconsistency, which can significantly reduce the time of system reboots. The following example shows how to mount `/dev/dsk/c0t3d0s7` on the `/files1` directory with logging enabled.

```
# mount -o logging /dev/dsk/c0t3d0s7 /files1
```

▼ How to Mount a UFS File System Without Large Files (`mount` Command)

When you mount a file system, the `largefiles` option is selected by default, which enables you to create files larger than 2 Gbytes. Once a file system contains large files, you cannot remount the file system with the `nolargefiles` option or mount it on a system that is running Solaris 2.6 or compatible versions, until you remove any large files and run the `fsck` command to reset the state to `nolargefiles`.

This procedure assumes that the file system is in the `/etc/vfstab` file.

1. **Become superuser or assume an equivalent role.**
2. **Create a mount point for the file system to be mounted, if necessary.**

   # **mkdir** */mount-point*

   There must be a mount point on the local system to mount a file system. A mount point is a directory to which the mounted file system is attached.
3. **Make sure there are no large files in the file system.**

   # **cd** */mount-point*
   # **find . -xdev -size +20000000 -exec ls -l {} \;**

   /mount-point identifies the mount point of the file system you want to check for large files.
4. **Remove or move any large files in this file system to another file system, if necessary.**
5. **Unmount the file system.**

```
# umount /mount-point
```

6. **Reset the file system state.**

```
# fsck /mount-point
```

7. **Remount the file system with the `nolargefiles` option.**

```
# mount -o nolargefiles /mount-point
```

Example—Mounting a File System Without Large Files (`mount` Command)

The following example shows how to check the `/datab` file system and remount it with the `nolargefiles` option.

```
# cd /datab
# find . -xdev -size +20000000 -exec ls -l {} \;
# umount /datab
# fsck /datab
# mount -o nolargefiles /datab
```

▼ How to Mount an NFS File System (`mount` Command)

1. **Become superuser or assume an equivalent role.**

2. **Create a mount point for the file system to be mounted, if necessary.**

```
# mkdir /mount-point
```

There must be a mount point on the local system to mount a file system. A mount point is a directory to which the mounted file system is attached.

3. **Make sure the resource (file or directory) is available from a server.**

To mount an NFS file system, the resource must be made available on the server by using the `share` command. For information on how to share resources, see "About the NFS Service" in *System Administration Guide: Resource Management and Network Services*.

4. **Mount the NFS file system.**

```
# mount -F nfs [-o mount-options] server:/directory /mount-point
```

`-o` *mount-options*	Specifies `mount` options that you can use to mount an NFS file system. See Table 40–2 for the list of commonly used `mount` options or `mount_nfs`(1M) for a complete list of options.
server:/directory	Specifies the server's host name that contains the shared resource, and the path to the file or directory to mount.
/mount-point	Specifies the directory on which to mount the file system.

Example—Mounting an NFS File System (`mount` Command)

The following example shows how to mount the `/export/packages` directory on `/mnt` from the server `pluto`.

```
# mount -F nfs pluto:/export/packages /mnt
```

▼ x86: How to Mount a PCFS (DOS) File System From a Hard Disk (`mount` Command)

Use the following procedure to mount a PCFS (DOS) file system from a hard disk.

1. **Become superuser or assume an equivalent role.**
2. **Create a mount point for the file system to be mounted, if necessary.**

   # **mkdir** */mount-point*

   There must be a mount point on the local system to mount a file system. A mount point is a directory to which the mounted file system is attached.
3. **Mount the PCFS file system.**

   # **mount -F pcfs** [**-o rw** | **ro**] **/dev/dsk/***device-name***:***logical-drive* */mount-point*

`-o rw | ro`	Specifies that you can mount a PCFS file system read/write (rw) or read-only (ro). If you do not specify this option, the default is rw.
`/dev/dsk/`*device-name*	Specifies the device name of the whole disk (for example, `/dev/dsk/c0t0d0p0`).
logical-drive	Specifies either the DOS logical drive letter (c through z) or a drive number (1 through 24). Drive c is equivalent to drive 1 and represents the Primary DOS slice on the drive. All other letters or numbers represent DOS logical drives within the Extended DOS slice.

/mount-point	Specifies the directory on which to mount the file system.

Note that the *device-name* and *logical-drive* must be separated by a colon.

x86: Examples—Mounting a PCFS (DOS) File System From a Hard Disk (`mount` Command)

The following example shows how to mount the logical drive in the primary DOS slice on the `/pcfs/c` directory.

```
# mount -F pcfs /dev/dsk/c0t0d0p0:c /pcfs/c
```

The following example shows how to mount the first logical drive in the extended DOS slice read-only on `/mnt`.

```
# mount -F pcfs -o ro /dev/dsk/c0t0d0p0:2 /mnt
```

Unmounting File Systems

The unmounting of a file system removes it from the file system mount point, and deletes the entry from the `/etc/mnttab` file. Some file system administration tasks cannot be performed on mounted file systems. You should unmount a file system when the following occurs:

- The file system is no longer needed or has been replaced by a file system that contains more current software.
- You need to check and repair the file system by using the `fsck` command. For more information about the `fsck` command, see Chapter 43.

  It is a good idea to unmount a file system before doing a complete backup. For more information about doing backups, see Chapter 47.

Note – File systems are automatically unmounted as part of the system shutdown procedure.

You can use the `umount -f` option to forcibly unmount a file system that is busy in an emergency situation. This practice is not recommended under normal circumstances because the unmounting of a file system with open files could cause a loss of data. This option is only available for UFS and NFS file systems.

Prerequisites for Unmounting File Systems

The prerequisites for unmounting file systems include the following:

- You must be superuser or assume an equivalent role.
- A file system must be available for unmounting. You cannot unmount a file system that is busy. A file system is considered busy if a user is accessing a directory in the file system, if a program has a file open in that file system, or if it is being shared. You can make a file system available for unmounting by doing the following:
  - Changing to a directory in a different file system.
  - Logging out of the system.
  - Using the `fuser` command to list all processes that are accessing the file system and to stop them if necessary. For more details, see "How to Stop All Processes Accessing a File System" on page 542.

    Notify users if you need to unmount a file system that they are using.
  - Unsharing the file system. For information about unsharing a file system, see `unshare`(1M).

▼ How to Verify a File System is Unmounted

To verify that you unmounted a file system or a number of file systems, examine the output from the `mount` command.

```
$ mount | grep unmounted-file-system
$
```

▼ How to Stop All Processes Accessing a File System

1. **Become superuser or assume an equivalent role.**
2. **List all the processes that are accessing the file system so that you know which processes you are going to stop.**

   ```
   # fuser -c [ -u ] /mount-point
   ```

`-c`	Reports on files that are mount points for file systems and any files within those mounted file systems.
`-u`	Displays the user login name for each process ID.
/mount-point	Specifies the name of the file system for which you want to stop processes.

3. **Stop all processes that are accessing the file system.**

```
# fuser -c -k /mount-point
```

A SIGKILL is sent to each process that is using the file system.

Note – You should not stop a user's processes without first warning the user.

---

4. **Verify that there are no processes that are accessing the file system.**

```
# fuser -c /mount-point
```

Example—Stopping All Processes Accessing a File System

The following example shows how to stop process 4006c that is using the /export/home file system.

```
# fuser -c /export/home
/export/home:     4006c
# fuser -c -k /export/home
/export/home:     4006c
# fuser -c /export/home
/export/home:
```

▼ How to Unmount a File System

Use the following procedure to unmount a file system, except for the root (/), /usr, or /var file systems.

Note – The root (/), /usr, and /var file systems can be unmounted only during a shutdown, since the system needs these file systems to function.

---

1. **Make sure that you have met the prerequisites listed in "Prerequisites for Unmounting File Systems" on page 542.**

2. **Unmount the file system.**

```
# umount /mount-point
```

/mount-point is the name of the file system that you want to unmount. This can be one of the following:

- The directory name where the file system is mounted
- The device name path of the file system
- The resource for an NFS file system
- The loopback directory for a LOFS file system

Examples—Unmounting a File System

The following example shows how to unmount a local home file system.

```
# umount /export/home
```

The following example shows how to unmount the file system on slice 7.

```
# umount /dev/dsk/c0t0d0s7
```

The following example shows how to forcibly unmount the `/export` file system.

```
# umount -f /export
#
```

The following example shows how to unmount all file systems in the `/etc/vfstab` file, except for the root (`/`), `/proc`, `/var`, and `/usr` file systems.

```
# umountall
```

All file systems are unmounted, except for those file systems that are busy.

CHAPTER 41

Using The CacheFS File System (Tasks)

This chapter describes how to set up and maintain CacheFS™ file systems.

This is a list of task maps in this chapter.

- "High-Level View of Using the CacheFS File System (Task Map)" on page 545
- "Creating and Mounting a CacheFS File System (Task Map)" on page 548
- "Maintaining a CacheFS File System (Task Map)" on page 553
- "Packing a Cached File System (Task Map)" on page 559
- "Collecting CacheFS Statistics (Task Map)" on page 568

For information on troubleshooting CacheFS errors, see "Troubleshooting `cachefspack` Errors" on page 564.

High-Level View of Using the CacheFS File System (Task Map)

Use this task map to identify all the tasks for using CacheFS file systems. Each task in this map points to a series of additional tasks such as creating and mounting the CacheFS file systems, and packing and maintaining the cache.

Task	Description	For Instructions
1. Create and mount a CacheFS file system	Create the cache and mount the file system in the cache.	"Creating and Mounting a CacheFS File System (Task Map)" on page 548

Task	Description	For Instructions
2. Maintain a CacheFS file system	Display and modify a CacheFS file system by unmounting, removing, or re-creating the cache.	"Maintaining a CacheFS File System (Task Map)" on page 553
3. (Optional) Pack and unpack a CacheFS file system	Determine whether you want to pack the cache and use packing lists. Packing the cache ensures that certain files and directories are always updated in the cache.	"Packing a Cached File System (Task Map)" on page 559
4. Collect CacheFS statistics	Determine cache performance and appropriate cache size.	"Collecting CacheFS Statistics (Task Map)" on page 568

Overview of the CacheFS File System

The CacheFS file system is a general purpose caching mechanism that improves NFS server performance and scalability by reducing server and network load. Designed as a layered file system, the CacheFS file system provides the ability to cache one file system on another. In an NFS environment, the CacheFS file system increases the client per server ratio, reduces server and network loads, and improves performance for clients on slow links, such as Point-to-Point Protocol (PPP).

How a CacheFS File System Works

You create a CacheFS file system on a client system so that file systems you cache can be accessed by the client locally instead of across the network. The following figure shows the relationship of the components that are involved in using CacheFS file system.

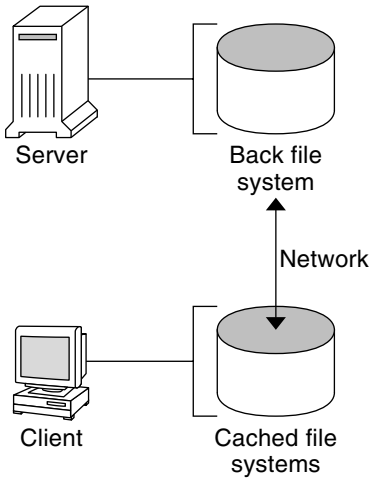


FIGURE 41–1 How a CacheFS File System Works

The *back* file system is the file system that you specify to be mounted in the cache, which can be either NFS or HSFS (High Sierra File System). When the user attempts to access files that are part of the back file system, those files are placed in the cache. The *front* file system is the file system that is mounted in the cache and is accessed from the local mount point. The front file system type must be UFS.

To the user, the initial request to access a file in a CacheFS file system might seem slow, but subsequent uses of the same file are faster.

CacheFS File System Structure and Behavior

Each cache has a set of parameters that determines the cache structure and how it behaves. The parameters are set to default values which are listed in the following table. The default values specify that the entire front file system is used for caching, which is the recommended method of caching file systems.

TABLE 41–1 CacheFS File System Parameters and Their Default Values

CacheFS File System Parameter	Default Value	Definition
`maxblocks`	90%	Sets the maximum number of blocks that a CacheFS file system is allowed to claim within the front file system.

TABLE 41–1 CacheFS File System Parameters and Their Default Values *(Continued)*

CacheFS File System Parameter	Default Value	Definition
`minblocks`	0%	Sets the minimum number of blocks that a CacheFS file system is allowed to claim within the front file system.
`threshblocks`	85%	Sets the number of blocks that must be available in the front file system before a CacheFS file system can claim more than the blocks specified by `minblocks`.
`maxfiles`	90%	Sets the maximum number of available inodes (number of files) that a CacheFS file system is allowed to claim within the front file system.
`minfiles`	0%	Sets the minimum number of available inodes that a CacheFS file system is allowed to claim within the front file system.
`threshfiles`	85%	Sets the number of inodes that must be available in the front file system before a CacheFS file system can claim more than the files specified in `minfiles`.

Typically, you should not change any of these parameter values. They are set to default values to achieve optimal cache behavior. However, you might want to modify the `maxblocks` and `maxfiles` values if you have some room in the front file system that is not used by the cache, and you want to use it for some other file system. You do so by using the `cfsadmin` command. For example:

```
$ cfsadmin -o maxblocks=60
```

Creating and Mounting a CacheFS File System (Task Map)

Use the procedures in this table to create and mount a CacheFS file system.

Task	Description	For Instructions
1. Share the file system to be cached	Verify that the file system you want to cache is shared.	`share`(1M)

Task	Description	For Instructions
2. Create the cache	Use the `cfsadmin` command to create the cache.	"How to Create the Cache" on page 549
3. Mount a file system in the cache	Mount a file system in a cache by using one of the following methods:	
	Mount a CacheFS file system by using the `mount` command.	"How to Mount a CacheFS File System (`mount`)" on page 550
	Mount a CacheFS file system by editing the `/etc/vfstab` file.	"How to Mount a CacheFS File System (`/etc/vfstab`)" on page 552
	Mount a cached a file system by using AutoFS.	"How to Mount a CacheFS File System (AutoFS)" on page 553

▼ How to Create the Cache

1. **Become superuser on the client system.**
2. **Create the cache.**

   # **cfsadmin -c** */cache-directory*

   cache-directory indicates the name of the directory where the cache resides.

   For more information, see `cfsadmin`(1M).

Note – After you have created the cache, do not perform any operations within the cache directory itself. Doing so could cause conflicts within the CacheFS software.

Example—Creating the Cache

The following example shows how to create a cache in the `/local/mycache` directory by using the default cache parameter values.

```
# mkdir /local
# cfsadmin -c /local/mycache
```

Mounting a File System in the Cache

You specify a file system to be mounted in the cache so that users can locally access files in that file system. The files do not actually get placed in the cache until the user accesses the files.

The following table describes three ways to mount a CacheFS file system.

Mount Type for CacheFS File System	Frequency of CacheFS Mount Type
Using the `mount` command	Every time the system reboots in order to access the same file system.
Editing the `/etc/vfstab` file	Only once. The `/etc/vfstab` file remains unchanged after the system reboots.
Using AutoFS	Only once. AutoFS maps remain unchanged after the system reboots.

Choose the method of mounting file systems that best suits your environment.

You can mount only file systems that are shared. For information on sharing file systems, see `share`(1M).

Note – The caching of the root (`/`) and `/usr` file systems is not supported in a CacheFS file system.

▼ How to Mount a CacheFS File System (`mount`)

1. **Become superuser on the client system.**
2. **Create the mount point, if necessary.**

   `#` **`mkdir`** */mount-point*

   You can create the mount point from anywhere but it must be a UFS file system. The CacheFS options used with the `mount` command, as shown in the next step, determine that the mount point you create is cached in the cache directory you specify.
3. **Mount a file system in the cache.**

   `#` **`mount -F cachefs -o backfstype=`***fstype*`,cachedir=`*/cache-directory* `[,`*options*`]` */back-filesystem /mount-point*

fstype	Indicates the file system type of the back file system, which can be either `NFS` or `HSFS`.
/cache-directory	Indicates the name of the UFS directory where the cache resides. This name is the same name you specified when you created the cache in "How to Create the Cache" on page 549.

options	Specifies other mount options that you can include when you mount a file system in a cache. For a list of CacheFS `mount` options, see `mount_cachefs`(1M).
/back-filesystem	Specifies the mount point of the back file system to cache. If the back file system is an NFS file system, you must specify the host name of the server from which you are mounting the file system and the name of the file system to cache, separated by a colon. For example, *merlin: /data/abc*.
/mount-point	Indicates the directory where the file system is mounted.

4. **Verify that the cache you created was actually mounted.**

   # **cachefsstat** */mount-point*

   The */mount-point* is the CacheFS file system that you created.

   For example:

```
# cachefsstat /docs
/docs
             cache hit rate:   100% (0 hits, 0 misses)
         consistency checks:      1 (1 pass, 0 fail)
                   modifies:      0
         garbage collection:      0
```

   If the file system was not mounted in the cache, you see an error message similar to the following:

```
# cachefsstat /mount-point
cachefsstat: mount-point: not a cachefs mountpoint
```

   For more information about the `cachefsstat` command, see "Collecting CacheFS Statistics" on page 568.

Examples—Mounting a CacheFS File System (`mount`)

The following example shows how to mount the NFS file system `merlin:/docs` as a CacheFS file system named `/docs` in the cache named `/local/mycache`.

```
# mkdir /docs
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache merlin:/docs /docs
```

The following example shows how to make a Solaris 9 SPARC CD (HSFS file system) available as a CacheFS file system named `/cfssrc`. Because you cannot write to the CD, the `ro` argument is specified to make the CacheFS file system read-only. This example assumes that `vold` is not running.

```
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /sol9
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/sol9 /dev/dsk/c0t6d0s0 /cfssrc
# ls /cfssrc
Copyright  Solaris_9
```

The following example shows how to mount a Solaris 9 SPARC CD as a CacheFS file system with `vold` running.

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/cdrom/sol_9_sparc/s0 /vol/dev/dsk/c0t2d0/sol_9_sparc/s0 /cfssrc
```

The following example shows how to mount a CD as a CacheFS file system with `vold` running.

```
# mount -F cachefs -o backfstype=hsfs,cachedir=/cfs/cache,ro,noconst,
backpath=/cdrom/epson /vol/dev/dsk/c0t2d0/epson /drvrs
```

The following example uses the `demandconst` option to specify consistency checking on demand for the NFS CacheFS file system `/docs`, whose back file system is `merlin:/docs`. For more information, see "Consistency Checking of a CacheFS File System" on page 556.

```
# mount -F cachefs -o backfstype=nfs,cachedir=/local/mycache,demandconst merlin:/docs /docs
```

▼ How to Mount a CacheFS File System (`/etc/vfstab`)

1. **Become superuser on the client system.**
2. **Using an editor, specify the file systems to be mounted in the `/etc/vfstab` file.**

   See the example that follows.

   For more information on the `/etc/vfstab` file, see "Field Descriptions for the `/etc/vfstab` File" on page 532.
3. **Mount the CacheFS file system.**

   # **mount** */mount-point*

   Or, reboot the system.

Example—Mounting a CacheFS File System (`/etc/vfstab`)

The following example shows the `/etc/vfstab` entry for the `/data/abc` directory from remote system `starbug` that is mounted in the cached directory, `/opt/cache`.

```
#device           device             mount      FS     fsck  mount   mount
#to mount         to fsck            point      type   pass  at boot options
#
starbug:/data/abc /local/abc         /opt/cache cachefs 7    yes     local-access,bg,
nosuid,demandconst,backfstype=nfs,cachedir=/opt/cache
```

▼ How to Mount a CacheFS File System (AutoFS)

You can mount a file system in a cache with AutoFS by specifying the `-fstype=cachefs` mount option in your automount map. Note that the CacheFS mount options, for example, `backfstype` and `cachedir`, are also specified in the automount map. For details on automount maps, see `automount`(1M).

1. **Become superuser on the client system.**
2. **Using an editor, add the following line to the `auto_direct` map:**

   /mount-point `-fstype=cachefs,cachedir=`*/directory*`,backfstype=nfs` *server:/file-system*
3. **Using an editor, add the following line to the `auto_master` map:**

   `/-`

   The `/-` entry is a pointer to check the `auto_direct` map.
4. **Reboot the system.**
5. **Verify that the entry was made correctly by changing to the file system you mounted in the cache, and then list the contents, as follows:**

   ```
   # cd /filesystem
   # ls
   ```

   For more information about AutoFS and how to edit the maps, refer to "Task Overview for Autofs Administration" in *System Administration Guide: Resource Management and Network Services*.

Example—Mounting a CacheFS File System (AutoFS)

The following `auto_direct` entry automatically mounts the CacheFS file system in the `/docs` directory.

```
/docs -fstype=cachefs,cachedir=/local/mycache,backfstype=nfs merlin:/docs
```

Maintaining a CacheFS File System (Task Map)

After a CacheFS file system is set up, it requires little maintenance. Use the optional procedures in this table if you need to perform maintenance tasks on your CacheFS file systems.

Task	Description	For Instructions
1. Modify a CacheFS file system	Modify CacheFS file system behavior by unmounting, deleting, or re-creating the cache.	"Modifying a CacheFS File System" on page 554
2. Display CacheFS file system information	Display information about CacheFS file systems by using the `cfsadmin` command.	"How to Display Information About a CacheFS File System" on page 555
3. Perform consistency checking	Perform consistency checking on demand by using the `cfsadmin` command.	"How to Specify Cache Consistency Checking on Demand" on page 556
4. Delete a CacheFS file system	Delete a CacheFS file system by using the `umount` command and the `cfsadmin` command.	"How to Delete a CacheFS File System" on page 556
5. Check the integrity of a CacheFS file system	Check the integrity of a CacheFS file system by using the `fsck_cachefs` command.	"How to Check the Integrity of a CacheFS File System" on page 558

Maintaining a CacheFS File System

This section describes how to maintain a CacheFS file system.

If you are using the `/etc/vfstab` file to mount file systems, you modify the cache by editing the file system options in the `/etc/vfstab` file. If you are using AutoFS, you modify the cache by editing the file system options in the AutoFS maps.

Modifying a CacheFS File System

When you modify a file system in the cache, you need to delete the cache and then re-create it. You might also need to reboot your machine in single-user mode, depending on how your file systems are shared and accessed.

In the following example, the cache is deleted, re-created, and then mounted again with the `demandconst` option specified for the `/docs` file system.

```
# shutdown -g30 -y
.
.
.
```

```
Type Cntrl-d to proceed with normal startup,
(or give root password for system maintenance):
# enter password:
.
.
.
```
Here is where you might be prompted to run fsck on the file system where the cache is located.

```
# fsck /local
# mount /local
# cfsadmin -d all /local/mycache
# cfsadmin -c /local/mycache
# init 6
.
.
.
console login:
password:
# mount -F cachefs -o backfstype=nfs,cachedir=/local/cache1,demandconst
merlin:/docs /docs
#
```

▼ How to Display Information About a CacheFS File System

1. **Become superuser on the client system.**
2. **Display information about all file systems cached under a specified cache.**

   # **cfsadmin -l** */cache-directory*

   /cache-directory is the name of the directory where the cache resides.

Example—Displaying Information About CacheFS File Systems

The following example shows information about the `/local/mycache` cache directory. In this example, the `/docs` file system is cached in `/local/mycache`. The last line displays the name of the CacheFS file system.

```
# cfsadmin -l /local/mycache
cfsadmin: list cache FS information
   maxblocks     90%
   minblocks      0%
   threshblocks  85%
   maxfiles      90%
   minfiles       0%
   threshfiles   85%
```

```
   maxfilesize    3MB
merlin:_docs:_docs
#
```

Consistency Checking of a CacheFS File System

To ensure that the cached directories and files remain current, the CacheFS software periodically checks the consistency of files stored in the cache. To check consistency, the CacheFS software compares the current modification time to the previous modification time. If the modification times are different, all data and attributes for the directory or file are purged from the cache. And, new data and attributes are retrieved from the back file system.

Consistency Checking on Demand

Consistency checks can be performed only when you explicitly request checks for file systems that are mounted with `-o demandconst` option. If you mount a file system in a cache with this option, then use the `cfsadmin` command with the `-s` option to request a consistency check. By default, consistency checking is performed file by file as the files are accessed. If no files are accessed, no checks are performed. Using the `-o demandconst` option avoids the situation where the network is flooded with consistency checks.

For more information, see `mount_cachefs`(1M).

▼ How to Specify Cache Consistency Checking on Demand

1. **Become superuser on the client system.**
2. **Mount the file system in the cache and specify cache consistency checking.**

   # **mount -F cachefs -o backfstype=nfs,cachedir=***/directory*,**demandconst** *server:/file-system /mount-point*
3. **Initiate consistency checking on a specific CacheFS file system.**

   # **cfsadmin -s** */mount-point*

▼ How to Delete a CacheFS File System

1. **Become superuser on the client system.**
2. **Unmount the CacheFS file system.**

   # **umount** */mount-point*

/mount-point specifies the CacheFS file system that you want to delete.

3. **Determine the name of the CacheFS file system (cache ID).**

```
# cfsadmin -l /cache-directory
cfsadmin: list cache FS information
   maxblocks     90%
   minblocks      0%
   threshblocks  85%
   maxfiles      90%
   minfiles       0%
   threshfiles   85%
   maxfilesize    3MB
cache-ID
#
```

4. **Delete the CacheFS file system from the specified cache.**

```
# cfsadmin -d cache-ID /cache-directory
```

cache-ID	Indicates the name of the CacheFS file system, which is the last line of the `cfsadmin -l` output. For more information, see "How to Display Information About a CacheFS File System" on page 555. You can delete all the CacheFS file systems in a particular cache by specifying `all` for *cache-ID*.
/cache-directory	Specifies the directory where the cache resides.

5. **Verify that the file system has been deleted.**

   The cache ID of the file system you just deleted should be missing from the `cfsadmin -l` output.

```
# cfsadmin -l /cache-directory
cfsadmin: list cache FS information
   maxblocks     90%
   minblocks      0%
   threshblocks  85%
   maxfiles      90%
   minfiles       0%
   threshfiles   85%
   maxfilesize    3MB
#
```

   For more information about the fields that are specified in the command output, refer to `cfsadmin`(1M).

6. **Update the resource counts for the cache by running the `fsck -F cachefs` command.**

   For more information, see "How to Check the Integrity of a CacheFS File System" on page 558.

Examples—Deleting a CacheFS File System

The following example shows how to delete the file systems from the cache.

```
# umount /cfssrc
# cfsadmin -l /cfssrc
# cfsadmin -d _dev_dsk_c0t6d0s0:_cfssrc
# cfsadmin -l
```

▼ How to Check the Integrity of a CacheFS File System

Use the `fsck` command to check the integrity of CacheFS file systems. The CacheFS version of the `fsck` command automatically corrects problems without requiring user interaction. You should not need to run the `fsck` command manually for CacheFS file systems because the `fsck` command is run automatically at boot time or when the file system is mounted. If you want to manually check the integrity, you can use the following procedure.

For more information, see `fsck_cachefs`(1M).

1. **Become superuser on the client system.**
2. **Check the file systems in the specified cache.**

   # **fsck -F cachefs** [**-m -o noclean**] */cache-directory*

`-m`	Causes the `fsck` command to check a CacheFS file system without making any repairs.
`-o noclean`	Forces a check on the CacheFS file systems only. Does not make any repairs.
/cache-directory	Indicates the name of the directory where the cache resides.

Example—Checking the Integrity of CacheFS File Systems

The following example shows how to check the file systems cached in the `/local/mycache` cache.

```
# fsck -F cachefs /local/mycache
#
```

Packing a Cached File System (Task Map)

The following task map describes the procedures that are associated with packing a CacheFS file system. All of these procedures are optional.

Task	Description	For Instructions
Pack files in the cache	Identify files and directories to be loaded in the cache and pack them. Packing ensures that current copies of these files are available in the cache.	"How to Pack Files in the Cache" on page 560
Create a packing list	Create a packing list if you do not want to specify each individual file that you want packed in the cache.	"How to Create a Packing List" on page 562
Pack files in the cache with a packing list	Specify the name of the packing list of the files to be packed in the cache.	"How to Pack Files in the Cache With a Packing List " on page 563
Unpack files or packing lists from the cache	Remove a file from the cache that is no longer needed.	"How to Unpack Files or Packing Lists From the Cache" on page 563
Display packed files information	View information about the files that you've packed, including their packing status.	"How to Display Packed Files Information" on page 561

Packing a CacheFS File System

For general use, the CacheFS software operates automatically after it is set up, without requiring any action from the user. Files are cached on a most recently used basis. With the *packing* feature, you can take a more active role in managing your cache by ensuring that certain files or directories are always updated in the cache.

You can specify files and directories to be loaded in the cache with the `cachefspack` command. This command ensures that current copies of these files are available in the cache.

The *packing list* contains the names of specific files and directories. The packing list can also contain other packing lists. This feature saves you having to specify individual files and directories when you have many items to pack in your cache.

You can print out a brief help summary of all the `cachefspack` options by using the `-h` option as follows:

```
$ cachefspack -h
Must select 1 and only 1 of the following 5 options
-d Display selected filenames
-i Display selected filenames packing status
-p Pack selected filenames
-u Unpack selected filenames
-U Unpack all files in directory 'dir'
-f Specify input file containing rules
-h Print usage information
-r Interpret strings in LIST rules as regular expressions
-s Strip './' from the beginning of a pattern name
-v Verbose option
files - a list of filenames to be packed/unpacked
```

▼ How to Pack Files in the Cache

Pack files in the cache by using the `cachefspack` command.

```
$ cachefspack -p filename
```

`-p`	Specifies that you want the file or files to be packed. This option is also the default.
filename	Specifies the name of the files or directory you want packed in the cache. When you specify a directory, all of its subdirectories are also packed. For more information, see `cachefspack`(1M).

Examples—Packing Files in the Cache

The following example shows the `projects` file being packed in the cache.

```
$ cachefspack -p projects
```

The following example shows three files being packed in the cache.

```
$ cachefspack -p projects updates master_plan
```

The following example shows a directory being packed in the cache.

```
$ cachefspack -p /data/abc/bin
```

▼ How to Display Packed Files Information

Display packed files information by using the `cachefspack -i` command.

```
$ cachefspack -i[v] filename
```

`-i`	Specifies that you want to view information about your packed files.
`-v`	Is the verbose option.
cached-filename-or-directory	Specifies the name of the file or directory for which to display information.

Example—Displaying Packed Files Information

The following example shows that the `doc_file` file is successfully packed.

```
$ cachefspack -i doc_file
cachefspack: file doc_file marked packed YES, packed YES
```

In the following example, the `/data/abc` directory contains the `bin` subdirectory. The `bin` subdirectory has three files: `big`, `medium`, and `small`. Although the `big` and `small` files are specified to be packed, they are not. The `medium` file is successfully packed.

```
$ cd /data/abc
$ cachefspack -i bin
.
.
.
cachefspack: file /bin/big marked packed YES, packed NO
cachefspack: file /bin/medium marked packed YES,
packed YES
cachefspack: file /bin/small marked packed YES,
packed NO
.
.
.
```

If you use the `-iv` options together, you get additional information as to whether the file or directory specified has been flushed from the cache. For example:

```
$ cd /data/bin
$ cachefspack -iv bin
.
.
.
cachefspack: file /bin/big marked packed YES, packed NO,
nocache YES
cachefspack: file /bin/medium marked packed YES,
packed YES, nocache NO
```

```
cachefspack: file /bin/small marked packed YES,
packed NO
nocache NO
.
.
.
```

The last line of the preceding example shows that the directory contents have not been flushed from the cache.

Using Packing Lists

One feature of the `cachefspack` command is the ability to create packing lists.

A packing list contains files or directories to be packed in the cache. If a directory is in the packing list, all of its subdirectories and files will also be packed.

This feature saves the time of having to specify each individual file that you want packed in the cache.

▼ How to Create a Packing List

To create a packing list, open a file by using `vi` or the editor of your choice. The packing list file format uses the same format as the `filesync` command. For more information, see `filesync`(1).

Two packing list features are the following:

- You can identify files in the packing list as regular expressions rather than literal file names so that you don't have to specify each individual file name.
- You can pack files from a shared directory by ensuring that you pack only those files that you own.

For more information on using these features, see `cachefspack`(1M).

Example—Creating a Packing List

The following example shows the contents of a packing list file.

```
BASE /home/ignatz
LIST plans
LIST docs
IGNORE *.ps
```

- The path identified with the `BASE` statement is the directory where you have items you want to pack.

- The two `LIST` statements identify specific files within that directory to pack.
- The `IGNORE` statement identifies the file type of `.ps`, which you do not want to pack.

▼ How to Pack Files in the Cache With a Packing List

Pack files in the packing list by using the `cachefspack -f` command, as follows:

```
$ cachefspack -f packing-list
```

`-f`	Specifies that you want to use a packing list.
packing-list	Specifies the name of the packing list.

Example—Packing Files in the Cache With a Packing List

This example uses the `list.pkg` file as the packing list for the `cachefspack` command.

```
$ cachefspack -f list.pkg
```

Unpacking Files or Packing Lists From the Cache

You might need to remove, or unpack, a file from the cache. Perhaps you have some files or directories that have a higher priority than others, so you need to unpack the less critical files. For example, you finished up a project and have archived the files that are associated with that project. You are now working on a new project, and therefore, a new set of files.

▼ How to Unpack Files or Packing Lists From the Cache

Unpack files or packing lists from the cache by using the `-u` or `-U` option of the `cachefspack` command.

```
$ cachefspack -u filename | -U cache-directory
```

`-u`	Specifies that you want the file or files unpacked. You must specify a filename with this option.

filename	Specifies the name of the file or packing list that you want unpacked in the cache.
`-U`	Specifies that you want to unpack all files in the cache.

For more information about the `cachefspack` command, see the man page.

Examples—Unpacking Files or Packing Lists From the Cache

The following example shows the file `/data/abc/bin/big` being unpacked from the cache.

```
$ cachefspack -u /data/abc/bin/big
```

The following example shows several files being unpacked from the cache.

```
$ cd /data/abc/bin/big
$ cachefspack -u big small medium
```

The following example shows how to unpack a packing list, which is a file that contains the path to a directory of files, as follows:

```
$ cachefspack -uf list.pkg
```

The following example uses the `-U` option to specify that all files in a cache directory being unpacked.

```
$ cachefspack -U /local/mycache
```

You cannot unpack a cache that does not have at least one file system mounted. With the `-U` option, if you specify a cache that does not contain mounted file systems, you see output similar to the following:

```
$ cachefspack -U /local/mycache
cachefspack: Could not unpack cache /local/mycache, no mounted
filesystems in the cache.
```

Troubleshooting `cachefspack` Errors

You might see the following error messages when you use the `cachefspack` command.

`cachefspack:` *pathname* `- can't open directory: permission denied`

Cause
You might not have the correct permissions to access the file or directory.

Action
Set the correct permissions.

`cachefspack: `*`pathname`*` - can't open directory: no such file or directory`

Cause
You might not have the correct file or directory.

Action
Check for a possible typo.

`cachefspack: `*`pathname`*` - can't open directory: stale NFS file handle`

Cause
The file or directory might have been moved or deleted from the server at the time you attempted to access it.

Action
Verify that the file or directory on the server is still accessible.

`cachefspack: `*`pathname`*` - can't open directory: interrupted system call`

Cause
You might have inadvertently pressed Control-C while issuing the command.

Action
Reissue the command.

`cachefspack: `*`pathname`*` - can't open directory: I/O error`

Cause
You might have a hardware problem.

Action
Check your hardware connections.

`cachefspack: error opening dir`

Cause
You might not have the correct file or directory. The path identified after the `BASE` command in the file format could be a file and not a directory. The path specified must be a directory.

Action
Check for a possible typo. Check the path identified after the `BASE` command in your file format. Make sure the path identifies a directory, not a file.

`cachefspack: unable to get shared objects`

Cause
The executable might be corrupt or in a format that is not recognizable.

Action
Replace the executable.

`cachefspack: `*`filename`*` - can't pack file: permission denied`

Cause
You might not have the correct permissions to access the file or directory.

Action
Set the correct permissions.

`cachefspack: `*filename*` - can't pack file: no such file or directory`

Cause
You might not have the correct file or directory.

Action
Check for a possible typo.

`cachefspack: `*filename*`- can't pack file: stale NFS file handle`

Cause
The file or directory might have been moved or deleted from the server at the time you attempted to access it.

Action
Verify that the file or directory on the server is still accessible.

`cachefspack: `*filename*`- can't pack file: interrupted system call`

Cause
You might have inadvertently pressed Control-C while issuing the command.

Action
Reissue the command.

`cachefspack: `*filename*`- can't pack file: I/O error`

Cause
You might have a hardware problem.

Action
Check your hardware connections.

`cachefspack: `*filename*`- can't pack file: no space left on device.`

Cause
The cache is out of disk space.

Action
You need to increase the size of the cache by increasing disk space.

`cachefspack: `*filename*` - can't unpack file: permission denied`

Cause
You might not have the correct permissions to access the file or directory.

Action
Set the correct permissions.

`cachefspack: `*filename*` - can't unpack file: no such file or directory`

Cause
You might not have the correct file or directory.

Action
Check for a possible typo.

`cachefspack: `*filename*`- can't unpack file: stale NFS file handle`

Cause
The file or directory might have been moved or deleted from the server at the time you attempted to access it.

Action
Verify that the file or directory on the server is still accessible.

`cachefspack: `*filename*` - can't unpack file: interrupted system call`

Cause
You might have pressed Control-C inadvertently while issuing the command.

Action
Reissue the command.

`cachefspack: `*filename*`- can't unpack file I/O error`

Cause
You might have a hardware problem.

Action
Check your hardware connections.

`cachefspack: only one 'd', 'i', 'p', or 'u' option allowed`

Cause
You entered more than one of these options in a command session.

Action
Select one option for the command session.

`cachefspack: can't find environment variable.`

Cause
You forgot to set a corresponding environment variable to match the `$` in your configuration file.

Action
Define the environment variable in the proper location.

`cachefspack: skipping LIST command - no active base`

Cause
A `LIST` command is present in your configuration file that has no corresponding `BASE` command.

Action
Define the `BASE` command.

Collecting CacheFS Statistics (Task Map)

The following task map shows the steps involved in collecting CacheFS statistics. All the procedures in this table are optional.

Task	Description	For Instructions
Set up logging	Set up logging on a CacheFS file system using the `cachefslog` command.	"How to Set Up CacheFS Logging" on page 570
Locate the log file	Locate the log file with the `cachefslog` command.	"How to Locate the CacheFS Log File" on page 570
Stop logging	Stop logging with the `cachefslog` command.	"How to Stop CacheFS Logging" on page 571
View the cache size	View the cache size by using the `cachefswssize` command.	"How to View the Working Set (Cache) Size" on page 571
View the cache statistics	View the statistics by using the `cachefsstat` command.	"How to View CacheFS Statistics" on page 572

Collecting CacheFS Statistics

Collecting CacheFS statistics enable you to do the following:

- Determine an appropriate cache size
- Observe the performance of the cache

These statistics will help you determine the trade-off between your cache size and the desired performance of the cache.

The CacheFS statistics commands consist of the following:

Command	Man Page	Description
`cachefslog`	`cachefslog`(1M)	Specifies the location of the log file. This command also displays where the statistics are currently being logged, and enables you to stop logging.
`cachefswssize`	`cachefswssize`(1M)	Interprets the log file to give a recommended cache size.
`cachefsstat`	`cachefsstat`(1M)	Displays statistical information about a specific file system or all CacheFS file systems. The information provided in the output of this command is taken directly from the cache.

Note – You can issue the CacheFS statistics commands from any directory. You must be superuser to issue the `cachefswssize` command.

The CacheFS statistics begin accumulating when you create the log file. When the work session is over, stop the logging by using the `cachefslog -h` command, as described in "How to Stop CacheFS Logging" on page 571.

Before using the CacheFS statistics commands, you must do the following:

- Set up your cache by using the `cfsadmin` command.
- Decide on an appropriate length of time to allow statistical information to collect in the log file you create. The length of time should equal a typical work session. For example, a day, a week, or a month.
- Select a location or path for the log file. Make sure that there is enough space to allow for the growth of the log file. The longer you intend to allow statistical information to collect in the log file, the more space you need.

Note – The following procedures are presented in a recommended order. The order is not required.

How to Set Up CacheFS Logging

1. **Set up logging.**

   ```
   $ cachefslog -f log-file-path /mount-point
   ```

`-f`	Sets up logging.
log-file-path	Specifies the location of the log file. The log file is a standard file you create with an editor, such as vi.
/mount-point	Designates the mount point (CacheFS file system) for which statistics are being collected.

2. **Verify that you correctly set up the log file.**

   ```
   $ cachefslog /mount-point
   ```

Example—Setting Up CacheFS Logging

The following example shows how to set up the `/var/tmp/samlog` log file to collect statistics about the `/home/sam` directory.

```
$ cachefslog -f /var/tmp/samlog /home/sam
  /var/tmp/samlog: /home/sam
```

How to Locate the CacheFS Log File

You can also use the `cachefslog` command with no options to locate a log file for a particular mount point.

```
$ cachefslog /mount-point
```

/mount-point specifies the CacheFS file system for which you want to view the statistics.

The following example shows what you would see if a log file has been set up. The location of the log file is `/var/tmp/stufflog`.

```
$ cachefslog /home/stuff
     /var/tmp/stufflog: /home/stuff
```

The following example shows that no log file has been set up for the specified file system.

```
$ cachefslog /home/zap
    not logged: /home/zap
```

How to Stop CacheFS Logging

Use the `cachefslog -h` option to stop logging.

```
$ cachefslog -h /mount-point
```

The following example shows how to stop logging on `/home/stuff`.

```
$ cachefslog -h /home/stuff
not logged: /home/stuff
```

If you get a system response other than the one specified here, you did not successfully stop logging. Check to see if you are using the correct log file name and mount point.

How to View the Working Set (Cache) Size

You might want to check if you need to increase the size of the cache. Or, you might want to determine what the ideal cache size is based on your activity since you last used the `cachefslog` command for a particular mount point.

1. **Become superuser on the client system.**
2. **View the current cache size and highest logged cache size.**

   ```
   # cachefswssize log-file-path
   ```

   For more information, see `cachefswssize`(1M).

Example—Viewing the Working Set (Cache) Size

In the following example, the `end size` is the size of the cache at the time you issued the `cachefswssize` command. The `high water size` is the largest size of the cache during the time frame in which logging occurred.

```
# cachefswssize /var/tmp/samlog

    /home/sam
           end size:  10688k
    high water size:  10704k

    /
           end size:   1736k
    high water size:   1736k

    /opt
           end size:    128k
    high water size:    128k

    /nfs/saturn.dist
           end size:   1472k
```

```
high water size:   1472k

/data/abc
       end size:   7168k
high water size:   7168k

/nfs/venus.svr4
       end size:   4688k
high water size:   5000k

/data
       end size:   4992k
high water size:   4992k

total for cache
   initial size: 110960k
       end size:  30872k
high water size:  30872k
```

Viewing CacheFS Statistics

You might want to view certain information about a specific CacheFS file system. The following table explains the terminology that is displayed in the statistics output.

TABLE 41–2 CacheFS Statistics Terminology

Output Term	Description
`cache hit rate`	The rate of cache hits versus cache misses, followed by the actual number of hits and misses. A cache hit occurs when the user wants to perform an operation on a file or files, and the file or files are actually in the cache. A cache miss occurs when the file is not in the cache. The load on the server is the sum of cache misses, consistency checks, and modifications (modifies).
`consistency checks`	The number of consistency checks performed, followed by the number that passed, and the number that failed.
`modifies`	The number of modify operations. For example, writes or creates.

How to View CacheFS Statistics

View the statistics with the `cachefsstat` command. You can view the statistics at any time. For example, you do not have to set up logging in order to view the statistics.

```
$ cachefsstat /mount-point
```

/mount-point specifies the CacheFS file system for which you want to view the statistics.

If you do not specify the mount point, statistics for all mounted CacheFS file systems will be displayed.

For more information, see `cachefsstat`(1M).

Example—Viewing CacheFS Statistics

This example shows how to view statistics on the cached file system, `/home/sam`.

```
$ cachefsstat /home/sam
        cache hit rate: 73% (1234 hits, 450 misses)
    consistency checks: 700 (650 pass, 50 fail)
               modifies: 321
garbage collection:  0
```

CHAPTER 42

Configuring Additional Swap Space (Tasks)

This chapter provides guidelines and step-by-step instructions for configuring additional swap space after the Solaris release is installed.

This is a list of step-by-step instructions in this chapter.

- "How to Create a Swap File and Make It Available" on page 582
- "How to Remove Unneeded Swap Space" on page 583

This is a list of the overview information in this chapter.

- "About Swap Space" on page 575
- "How Do I Know If I Need More Swap Space?" on page 577
- "How Swap Space Is Allocated" on page 578
- "Planning for Swap Space" on page 579
- "Monitoring Swap Resources" on page 580
- "Adding More Swap Space" on page 581

About Swap Space

System administrators should understand the features of the SunOS swap mechanism to determine the following:

- Swap space requirements
- The relationship between swap space and the TMPFS file system
- Recovery from error messages related to swap space

Swap Space and Virtual Memory

The Solaris software uses some disk slices for temporary storage rather than for file systems. These slices are called *swap* slices. Swap slices are used as virtual memory storage areas when the system does not have enough physical memory to handle current processes.

The virtual memory system maps physical copies of files on disk to virtual addresses in memory. Physical memory pages that contain the data for these mappings can be backed by regular files in the file system, or by swap space. If the memory is backed by swap space it is referred to as *anonymous* memory because there is no identity assigned to the disk space that is backing the memory.

The Solaris environment uses the concept of *virtual swap space*, a layer between anonymous memory pages and the physical storage (or disk-backed swap space) that actually back these pages. A system's virtual swap space is equal to the sum of all its physical (disk-backed) swap space plus a portion of the currently available physical memory.

Virtual swap space has these advantages:

- The need for large amounts of physical swap space is reduced because virtual swap space does not necessarily correspond to physical (disk) storage.
- A pseudo file system called SWAPFS provides addresses for anonymous memory pages. Because SWAPFS controls the allocation of memory pages, it has greater flexibility in deciding what happens to a page. For example, SWAPFS might change the page's requirements for disk-backed swap storage.

Swap Space and the TMPFS File System

The TMPFS file system is activated automatically in the Solaris environment by an entry in the `/etc/vfstab` file. The TMPFS file system stores files and their associated information in memory (in the `/tmp` directory) rather than on disk, which speeds access to those files. This feature results in a major performance enhancement for applications such as compilers and DBMS products that use `/tmp` heavily.

The TMPFS file system allocates space in the `/tmp` directory from the system's swap resources. This feature means that as you use up space in the `/tmp` directory, you are also using up swap space. So if your applications use the `/tmp` directory heavily and you do not monitor swap space usage, your system could run out of swap space.

Use the following if you want to use TMPFS but your swap resources are limited:

- Mount the TMPFS file system with the size option (`-o` *size*) to control how much swap resources TMPFS can use.
- Use your compiler's `TMPDIR` environment variable to point to another larger directory.

Using your compiler's `TMPDIR` variable only controls whether the compiler is using the `/tmp` directory. This variable has no effect on other programs' use of the `/tmp` directory.

Swap Space as a Dump Device

A dump device is usually disk space that is reserved to store system crash dump information. By default, a system's dump device is configured to be an appropriate swap partition. If possible, you should configure a alternate disk partition as a *dedicated dump device* instead to provide increased reliability for crash dumps and faster reboot time after a system failure. You can configure a dedicated dump device by using the `dumpadm` command. For more information, see "Managing System Crash Information (Tasks)" in *System Administration Guide: Advanced Administration*.

If you are using a volume manager to manage your disks, such as Solaris Volume Manager, do not configure your dedicated dump device to be under the control of Solaris Volume Manager. You can keep your swap areas under Solaris Volume Manager's control, which is a recommended practice. However, for accessibility and performance reasons, configure another disk as a dedicated dump device outside of Solaris Volume Manager's control.

How Do I Know If I Need More Swap Space?

Use the `swap -l` command to determine if your system needs more swap space.

For example, the following `swap -l` output shows that this system's swap space is almost entirely consumed or at 100% allocation.

```
% swap -l
swapfile             dev   swaplo blocks   free
/dev/dsk/c0t0d0s1   136,1      16 1638608    88
```

When a system's swap space is at 100% allocation, an application's memory pages become temporarily locked. Application errors might not occur, but system performance will likely suffer.

For information on adding more swap space to your system, see "How to Create a Swap File and Make It Available" on page 582.

Swap-Related Error Messages

These messages indicate that an application was trying to get more anonymous memory, and there was no swap space left to back it.

application `is out of memory`

```
malloc error 0
```

```
messages.1:Sep 21 20:52:11 mars genunix: [ID 470503 kern.warning]
WARNING: Sorry, no swap space to grow stack for pid 100295 (myprog)
```

TMPFS-Related Error Messages

The following message is displayed if a page could not be allocated when writing a file. This problem can occur when TMPFS tries to write more than it is allowed or if currently executed programs are using a lot of memory.

directory`: File system full, swap space limit exceeded`

The following message means TMPFS ran out of physical memory while attempting to create a new file or directory.

directory`: File system full, memory allocation failed`

For information on recovering from the TMPFS-related error messages, see `TMPFS`(7FS).

How Swap Space Is Allocated

Initially, swap space is allocated as part of the Solaris installation process. If you use the installation program's automatic layout of disk slices and do not manually change the size of the swap slice, the Solaris installation program allocates a default swap area of 512 Mbytes.

Starting in the Solaris 9 release, the installation program allocates swap space starting at the first available disk cylinder (typically cylinder 0). This placement provides maximum space for the root (/) file system during the default disk layout and enables the growth of the root (/) file system during an upgrade.

For general guidelines on allocating swap space, see "Planning for Swap Space" on page 579.

You can allocate additional swap space to the system by creating a swap file. For information about creating a swap file, see "Adding More Swap Space" on page 581.

The /etc/vfstab File

After the system is installed, swap slices and swap files are listed in the /etc/vfstab file. They are activated by the /sbin/swapadd script when the system is booted.

An entry for a swap device in the /etc/vfstab file contains the following:

- The full path name of the swap slice or swap file
- File system type of swap

The file system that contains a swap file must be mounted before the swap file is activated. So, in the /etc/vfstab file, make sure that the entry that mounts the file system comes before the entry that activates the swap file.

Planning for Swap Space

The most important factors in determining swap space size are the requirements of the system's software applications. For example, large applications such as computer-aided-design simulators, database-management products, transaction monitors, and geologic analysis systems can consume as much as 200-1000 Mbytes of swap space.

Consult your application vendor for swap space requirements for their applications.

If you are unable to determine swap space requirements from your application vendor, use the following general guidelines based on your system type to allocate swap space:

System Type	Swap Space Size	Dedicated Dump Device Size
Workstation with approximately 4 Gbytes of physical memory	1 Gbyte	1 Gbyte
Mid-range server with approximately 8 Gbytes of physical memory	2 Gbytes	2 Gbytes
High-end server with approximately 16 to 128 Gbytes of physical memory	4 Gbytes	4 Gbytes

In addition to the general guidelines, consider allocating swap or disk space for the following:

- A dedicated dump device.

- Determine whether large applications (like compilers) will be using the `/tmp` directory. Then allocate additional swap space to be used by TMPFS. For information about TMPFS, see "Swap Space and the TMPFS File System" on page 576.

Monitoring Swap Resources

The `/usr/sbin/swap` command is used to manage swap areas. Two options, `-l` and `-s`, display information about swap resources.

Use the `swap -l` command to identify a system's swap areas. Activated swap devices or files are listed under the `swapfile` column.

```
# swap -l
swapfile             dev  swaplo blocks   free
/dev/dsk/c0t0d0s1   136,1      16 1638608 1600528
```

Use the `swap -s` command to monitor swap resources.

```
# swap -s
total: 57416k bytes allocated + 10480k reserved = 67896k used,
833128k available
```

The `used` value plus the `available` value equals the total swap space on the system, which includes a portion of physical memory and swap devices (or files).

You can use the amount of available and used swap space (in the `swap -s` output) as a way to monitor swap space usage over time. If a system's performance is good, use `swap -s` to see how much swap space is available. When the performance of a system slows down, check the amount of available swap space to see if it has decreased. Then you can identify what changes to the system might have caused swap space usage to increase.

When using this command, keep in mind that the amount of physical memory available for swap usage changes dynamically as the kernel and user processes lock down and release physical memory.

Note – The `swap -l` command displays swap space in 512-byte blocks and the `swap -s` command displays swap space in 1024-byte blocks. If you add up the blocks from `swap -l` and convert them to Kbytes, the result will be less than `used` + `available` (in the `swap -s` output) because `swap -l` does not include physical memory in its calculation of swap space.

The output from the `swap -s` command is summarized in the following table.

TABLE 42–1 Output of the `swap -s` Command

Keyword	Description
`bytes allocated`	The total amount of swap space in 1024-byte blocks that is currently allocated as backing store (disk-backed swap space).
`reserved`	The total amount of swap space in 1024-byte blocks that is not currently allocated, but claimed by memory for possible future use.
`used`	The total amount of swap space in 1024-byte blocks that is either allocated or reserved.
`available`	The total amount of swap space in 1024-byte blocks that is currently available for future reservation and allocation.

Adding More Swap Space

As system configurations change and new software packages are installed, you might need to add more swap space. The easiest way to add more swap space is to use the `mkfile` and `swap` commands to designate a part of an existing UFS or NFS file system as a supplementary swap area. These commands, described in the following sections, enable you to add more swap space without repartitioning a disk.

Alternative ways to add more swap space are to repartition an existing disk or add another disk. For information on how to repartition a disk, see Chapter 32.

Creating a Swap File

The following general steps are involved in creating a swap file:

- Creating a swap file with the `mkfile` command
- Activating the swap file with the `swap` command
- Adding an entry for the swap file in the `/etc/vfstab` file so that the swap file is activated automatically when the system is booted.

The `mkfile` Command

The `mkfile` command creates a file that is suitable for use as either an NFS-mounted or a local swap area. The sticky bit is set, and the file is filled with zeros. You can specify the size of the swap file in bytes (the default) or in Kbytes, blocks, or Mbytes by using the `k`, `b`, or `m` suffixes, respectively.

The following table shows the `mkfile` command options.

TABLE 42–2 Options to the `mkfile` Command

Option	Description
`-n`	Creates an empty file. The size is noted, but the disk blocks are not allocated until data is written to them.
`-v`	Reports the names and sizes of created files.

Caution – Use the `-n` option only when you create an NFS swap file.

▼ How to Create a Swap File and Make It Available

1. **Become superuser.**

   You can create a swap file without root permissions. However, to avoid accidental overwriting, root should be the owner of the swap file.

2. **Create a directory for the swap file, if needed.**

3. **Create the swap file.**

   `#` **`mkfile`** *nnn* **`[k|b|m]`** *filename*

   The swap file of the size *nnn* (in Kbytes, bytes, or Mbytes) and filename you specify is created.

4. **Activate the swap file.**

   `#` **`/usr/sbin/swap -a`** */path/filename*

   You must use the absolute path name to specify the swap file. The swap file is added and available until the file system is unmounted, the system is rebooted, or the swap file is removed. Keep in mind that you can't unmount a file system while some process or program is swapping to the swap file.

5. **Add an entry for the swap file to the `/etc/vfstab` file that specifies the full path name of the file, and designates `swap` as the file system type, as follows:**

   /path/filename `-  -  swap  -  no  -`

6. **Verify that the swap file is added.**

   `$` **`/usr/sbin/swap -l`**

Example—Creating a Swap File and Making It Available

The following examples shows how to create a 100–Mbyte swap file called `/files/swapfile`.

```
# mkdir /files
# mkfile 100m /files/swapfile
# swap -a /files/swapfile
# vi /etc/vfstab
(An entry is added for the swap file):
/files/swapfile   -      -       swap     -     no     -
# swap -l
swapfile             dev  swaplo blocks   free
/dev/dsk/c0t0d0s1   136,1      16 1638608 1600528
/files/swapfile        -       16 204784  204784
```

Removing a Swap File From Use

If you have unneeded swap space, you can remove it.

▼ How to Remove Unneeded Swap Space

1. **Become superuser.**
2. **Remove the swap space.**

   # **/usr/sbin/swap -d** */path/filename*

   The swap file name is removed so that it is no longer available for swapping. The file itself is not deleted.
3. **Edit the `/etc/vfstab` file and delete the entry for the swap file.**
4. **Recover the disk space so that you can use it for something else.**

   # **rm** */path/filename*

   If the swap space is a file, remove it. Or, if the swap space is on a separate slice and you are sure you will not need it again, make a new file system and mount the file system.

   For information on mounting a file system, see Chapter 40.
5. **Verify that the swap file is no longer available.**

   # **swap -l**

Example—Removing Unneeded Swap Space

The following examples shows how to delete the `/files/swapfile` swap file.

```
# swap -d /files/swapfile
#  (Remove the swap entry from the /etc/vfstab file)
# rm /files/swapfile
# swap -l
swapfile             dev  swaplo  blocks   free
/dev/dsk/c0t0d0s1   136,1      16 1638608 1600528
```

CHAPTER 43

Checking UFS File System Consistency (Tasks)

This chapter provides overview information and step-by-step instructions about checking UFS file system consistency.

This is a list of step-by-step instructions in this chapter.

- "How to See If a File System Needs Checking" on page 595
- "How to Check File Systems Interactively" on page 595
- "How to Preen a UFS File System" on page 597
- "How to Restore a Bad Superblock" on page 598

This is a list of the overview information in this chapter.

- "File System Consistency" on page 585
- "How the File System State Is Recorded" on page 586
- "What the `fsck` Command Checks and Tries to Repair" on page 588
- "Interactively Checking and Repairing a UFS File System" on page 594
- "Restoring a Bad Superblock" on page 598
- "Syntax and Options for the `fsck` Command" on page 600

For information about `fsck` error messages, see "Resolving UFS File System Inconsistencies (Tasks)" in *System Administration Guide: Advanced Administration*.

For background information on the UFS file system structures referred to in this chapter, see Chapter 44.

File System Consistency

The UFS file system relies on an internal set of tables to keep track of inodes used and available blocks. When these internal tables are not properly synchronized with data on a disk, inconsistencies result and file systems need to be repaired.

File systems can be inconsistent because of abrupt termination of the operating system in these ways:

- Power failure
- Accidental unplugging of the system
- Turning off the system without proper shutdown procedure
- A software error in the kernel

File system inconsistencies, while serious, are not common. When a system is booted, a check for file system consistency is automatically performed (with the `fsck` command). Most of the time, this file system check repairs problems it encounters.

The `fsck` command places files and directories that are allocated but unreferenced in the `lost+found` directory. A inode number is assigned as the name of unreferenced file and directory. If the `lost+found` directory does not exist, the `fsck` command creates it. If there is not enough space in the `lost+found` directory, the `fsck` command increases its size.

For a description of inodes, see "Inodes" on page 610.

How the File System State Is Recorded

The `fsck` command uses a state flag, which is stored in the superblock, to record the condition of the file system. This flag is used by the `fsck` command to determine whether a file system needs to be checked for consistency. The flag is used by the `/sbin/rcS` script during booting and by the `fsck -m` command. If you ignore the result from the `fsck -m` command, all file systems can be checked regardless of the setting of the state flag.

For a description of the superblock, see "The Superblock" on page 610.

The possible state flag values are described in the following table.

TABLE 43–1 Values of File System State Flags

State Flag Values	Description
`FSACTIVE`	When a file system is mounted and then modified, the state flag is set to `FSACTIVE`. The file system might contain inconsistencies. A file system is marked as `FSACTIVE` before any modified metadata is written to the disk. When a file system is unmounted gracefully, the state flag is set to `FSCLEAN`. A file system with the `FSACTIVE` flag must be checked by the `fsck` command because it might be inconsistent.

TABLE 43–1 Values of File System State Flags *(Continued)*

State Flag Values	Description
FSBAD	If the root (/) file system is mounted when its state is not FSCLEAN or FSSTABLE, the state flag is set to FSBAD. The kernel will not change this file system state to FSCLEAN or FSSTABLE. If a root (/) file system is flagged FSBAD as part of the boot process, it will be mounted read-only. You can run the fsck command on the raw root device. Then remount the root (/) file system with read and write access.
FSCLEAN	If a file system is unmounted properly, the state flag is set to FSCLEAN. Any file system with an FSCLEAN state flag is not checked when the system is booted.
FSLOG	If a file system is mounted with UFS logging, the state flag is set to FSLOG. Any file system with an FSLOG state flag is not checked when the system is booted.
FSSTABLE	The file system is (or was) mounted but has not changed since the last checkpoint (sync or fsflush) that normally occurs every 30 seconds. For example, the kernel periodically checks if a file system is idle and, if so, flushes the information in the superblock back to the disk and marks it as FSSTABLE. If the system crashes, the file system structure is stable, but users might lose a small amount of data. File systems that are marked as FSSTABLE can skip the checking before mounting. The mount command will not mount a file system for read and write access if the file system state is not FSCLEAN, FSSTABLE, or FSLOG.

The following table shows how the state flag is modified by the fsck command, based on its initial state.

TABLE 43–2 How the State Flag is Modified by fsck

Initial State: Before fsck	State After fsck		
	No Errors	All Errors Corrected	Uncorrected Errors
unknown	FSSTABLE	FSSTABLE	unknown
FSACTIVE	FSSTABLE	FSSTABLE	FSACTIVE
FSSTABLE	FSSTABLE	FSSTABLE	FSACTIVE
FSCLEAN	FSCLEAN	FSSTABLE	FSACTIVE
FSBAD	FSSTABLE	FSSTABLE	FSBAD
FSLOG	FSLOG	FSLOG	FSLOG

What the `fsck` Command Checks and Tries to Repair

This section describes what happens in the normal operation of a file system, what can go wrong, what problems the `fsck` command (the checking and repair utility) looks for, and how this command corrects the inconsistencies it finds.

Why Inconsistencies Might Occur

Every working day hundreds of files might be created, modified, and removed. Each time a file is modified, the operating system performs a series of file system updates. These updates, when written to the disk reliably, yield a consistent file system.

When a user program does an operation to change the file system, such as a write, the data to be written is first copied into an in-core buffer in the kernel. Normally, the disk update is handled asynchronously. The user process is allowed to proceed even though the data write might not happen until long after the write system call has returned. Thus, at any given time, the file system, as it resides on the disk, lags behind the state of the file system that is represented by the in-core information.

The disk information is updated to reflect the in-core information when the buffer is required for another use or when the kernel automatically runs the `fsflush` daemon (at 30-second intervals). If the system is halted without writing out the in-core information, the file system on the disk might be in an inconsistent state.

A file system can develop inconsistencies in several ways. The most common causes are operator error and hardware failures.

Problems might result from an *unclean shutdown*, if a system is shut down improperly, or when a mounted file system is taken offline improperly. To prevent unclean shutdowns, the current state of the file systems must be written to disk (that is, "synchronized") before you shut down the system, physically take a disk pack out of a drive, or take a disk offline.

Inconsistencies can also result from defective hardware. Blocks can become damaged on a disk drive at any time, or a disk controller can stop functioning correctly.

The UFS Components That Are Checked for Consistency

This section describes the kinds of consistency checks that the `fsck` command applies to these UFS file system components: superblock, cylinder group blocks, inodes, indirect blocks, and data blocks.

For information about UFS file system structures, see “The Structure of Cylinder Groups for UFS File Systems” on page 609.

Superblock Checks

The superblock stores summary information, which is the most commonly corrupted component in a UFS file system. Each change to the file system inodes or data blocks also modifies the superblock. If the CPU is halted and the last command is not a `sync` command, the superblock almost certainly becomes corrupted.

The superblock is checked for inconsistencies in the following:

- File system size
- Number of inodes
- Free block count
- Free inode count

File System Size and Inode List Size Checks

The file system size must be larger than the number of blocks used by the superblock and the list of inodes. The number of inodes must be less than the maximum number allowed for the file system. An inode represents all the information about a file. The file system size and layout information are the most critical pieces of information for the `fsck` command. Although there is no way to actually check these sizes because they are statically determined when the file system is created. However, the `fsck` command can check that the sizes are within reasonable bounds. All other file system checks require that these sizes be correct. If the `fsck` command detects corruption in the static parameters of the primary superblock, it requests the operator to specify the location of an alternate superblock.

For more information about the structure of the UFS file system, see “The Structure of Cylinder Groups for UFS File Systems” on page 609.

Free Block Checks

Free blocks are stored in the cylinder group block maps. The `fsck` command checks that all the blocks marked as free are not claimed by any files. When all the blocks have been accounted for, the `fsck` command checks to see if the number of free blocks plus the number of blocks that are claimed by the inodes equal the total number of blocks in the file system. If anything is wrong with the block maps, the `fsck` command rebuilds them, leaving out blocks already allocated.

The summary information in the superblock includes a count of the total number of free blocks within the file system. The `fsck` command compares this count to the number of free blocks it finds within the file system. If the counts do not agree, the `fsck` command replaces the count in the superblock with the actual free-block count.

Free Inode Checks

The summary information in the superblock contains a count of the free inodes within the file system. The `fsck` command compares this count to the number of free inodes it finds within the file system. If the counts do not agree, `fsck` replaces the count in the superblock with the actual free inode count.

Inodes

The list of inodes is checked sequentially starting with inode 2 (inode 0 and inode 1 are reserved). Each inode is checked for inconsistencies in the following:

- Format and type
- Link count
- Duplicate block
- Bad block numbers
- Inode size

Format and Type of Inodes

Each inode contains a mode word, which describes the type and state of the inode. Inodes might be one of nine types:

- Regular
- Directory
- Block special
- Character special
- FIFO (named-pipe)
- Symbolic link
- Shadow (used for ACLs)
- Attribute directory
- Socket

Inodes might be in one of three states:

- Allocated
- Unallocated
- Partially allocated

When the file system is created, a fixed number of inodes are set aside, but they are not allocated until they are needed. An allocated inode is one that points to a file. An unallocated inode does not point to a file and, therefore, should be empty. The partially allocated state means that the inode is incorrectly formatted. An inode can get into this state if, for example, bad data is written into the inode list because of a hardware failure. The only corrective action the `fsck` command can take is to clear the inode.

Link Count Checks

Each inode contains a count of the number of directory entries linked to it. The `fsck` command verifies the link count of each inode by examining the entire directory structure, starting from the root directory, and calculating an actual link count for each inode.

Discrepancies between the link count stored in the inode and the actual link count as determined by the `fsck` command might be of three types:

- The stored count is *not* 0 and the actual count is 0.

  This condition can occur if no directory entry exists for the inode. In this case, the `fsck` command puts the disconnected file in the `lost+found` directory.
- The stored count is *not* 0 and the actual count is *not* 0, but the counts are *unequal*.

  This condition can occur if a directory entry has been added or removed, but the inode has not been updated. In this case, the `fsck` command replaces the stored link count with the actual link count.
- The stored count is 0 and the actual count is not 0.

  In this case, the `fsck` command changes the link count of the inode to the actual count.

Duplicate Block Checks

Each inode contains a list, or pointers to lists (indirect blocks), of all the blocks claimed by the inode. Because indirect blocks are owned by an inode, inconsistencies in indirect blocks directly affect the inode that owns the indirect block.

The `fsck` command compares each block number claimed by an inode to a list of allocated blocks. If another inode already claims a block number, the block number is put on a list of duplicate blocks. Otherwise, the list of allocated blocks is updated to include the block number.

If there are any duplicate blocks, the `fsck` command makes a second pass of the inode list to find the other inode that claims each duplicate block. (A large number of duplicate blocks in an inode might be caused by an indirect block not being written to the file system.) It is not possible to determine with certainty which inode is in error. The `fsck` command prompts you to choose which inode should be kept and which should be cleared.

Bad Block Number Checks

The `fsck` command checks each block number claimed by an inode to see that its value is higher than that of the first data block and lower than that of the last data block in the file system. If the block number is outside this range, it is considered a bad block number.

Bad block numbers in an inode might be caused by an indirect block not being written to the file system. The `fsck` command prompts you to clear the inode.

Inode Size Checks

Each inode contains a count of the number of data blocks that it references. The number of actual data blocks is the sum of the allocated data blocks and the indirect blocks. The `fsck` command computes the number of data blocks and compares that block count against the number of blocks that the inode claims. If an inode contains an incorrect count, the `fsck` command prompts you to fix it.

Each inode contains a 64-bit size field. This field shows the number of characters (data bytes) in the file associated with the inode. A rough check of the consistency of the size field of an inode is done by using the number of characters shown in the size field to calculate how many blocks should be associated with the inode, and then comparing that to the actual number of blocks claimed by the inode.

Indirect Blocks

Indirect blocks are owned by an inode. Therefore, inconsistencies in an indirect block affect the inode that owns it. Inconsistencies that can be checked are the following:

- Blocks already claimed by another inode
- Block numbers outside the range of the file system

These consistency checks listed are also performed for indirect blocks.

Data Blocks

An inode can directly or indirectly reference three kinds of data blocks. All referenced blocks must be of the same kind. The three types of data blocks are the following:

- Plain data blocks
- Symbolic-link data blocks
- Directory data blocks

Plain data blocks contain the information stored in a file. Symbolic-link data blocks contain the path name stored in a symbolic link. Directory data blocks contain directory entries. The `fsck` command can check only the validity of directory data blocks.

Directories are distinguished from regular files by an entry in the `mode` field of the inode. Data blocks associated with a directory contain the directory entries. Directory data blocks are checked for inconsistencies involving the following:

- Directory inode numbers that point to unallocated inodes
- Directory inode numbers that are greater than the number of inodes in the file system

- Incorrect directory inode numbers for " . " and " . . " directories
- Directories that are disconnected from the file system

Directory Unallocated Checks

If the inode number in a directory data block points to an unallocated inode, the `fsck` command removes the directory entry. This condition can occur if the data blocks that contain a new directory entry are modified and written out, but the inode does not get written out. This condition can occur if the CPU is shutdown abruptly.

Bad Inode Number Checks

If a directory entry inode number points beyond the end of the inode list, the `fsck` command removes the directory entry. This condition can occur when bad data is written into a directory data block.

Incorrect " . " and " . . " Entry Checks

The directory inode number entry for " . " must be the first entry in the directory data block. The directory inode number must reference itself; that is, its value must be equal to the inode number for the directory data block.

The directory inode number entry for " . . " must be the second entry in the directory data block. The directory inode number value must be equal to the inode number of the parent directory (or the inode number of itself if the directory is the root directory).

If the directory inode numbers for " . " and " . . " are incorrect, the `fsck` command replaces them with the correct values. If there are multiple hard links to a directory, the first hard link found is considered the real parent to which " . . " should point. In this case, the `fsck` command recommends that you have it delete the other names.

Disconnected Directories

The `fsck` command checks the general connectivity of the file system. If a directory is found that is not linked to the file system, the `fsck` command links the directory to the `lost+found` directory of the file system. This condition can occur when inodes are written to the file system, but the corresponding directory data blocks are not.

Regular Data Blocks

Data blocks associated with a regular file hold the contents of the file. The `fsck` command does not attempt to check the validity of the contents of a regular file's data blocks.

The `fsck` Summary Message

When you run the `fsck` command interactively and it completes successfully, a message similar to the following is displayed:

```
# fsck /dev/rdsk/c0t0d0s7
** /dev/rdsk/c0t0d0s7
** Last Mounted on /export/home
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
2 files, 9 used, 2833540 free (20 frags, 354190 blocks, 0.0% fragmentation)
#
```

The last line of `fsck` output describes the following information about the file system:

# `files`	Number of inodes in use
# `used`	Number of fragments in use
# `free`	Number of unused fragments
# `frags`	Number of unused non-block fragments
# `blocks`	Number of unused full blocks
% `fragmentation`	Percentage of fragmentation, where: free fragments x 100 / total fragments in the file system

For information about fragments, see "Fragment Size" on page 614.

Interactively Checking and Repairing a UFS File System

You might need to interactively check file systems in the following instances:

- When they cannot be mounted
- When they develop inconsistences while in use

When an in-use file system develops inconsistencies, error messages might be displayed in the console window or the system might crash.

Before using the `fsck` command, you might want to refer to "Syntax and Options for the `fsck` Command" on page 600 and "Resolving UFS File System Inconsistencies (Tasks)" in *System Administration Guide: Advanced Administration* for more information.

▼ How to See If a File System Needs Checking

1. **Become superuser or assume an equivalent role.**
2. **Unmount the file system if it is mounted.**

   # **umount** */mount-point*

3. **Check the file system.**

   # **fsck -m /dev/rdsk/***device-name*

   The state flag in the superblock of the file system you specify is checked to see whether the file system is clean or requires checking.

   If you omit the device argument, all the UFS file systems listed in the `/etc/vfstab` file with a `fsck pass` value greater than 0 are checked.

Example—Seeing If a File System Needs Checking

The following example shows that the file system needs checking.

```
# fsck -m /dev/rdsk/c0t0d0s6
** /dev/rdsk/c0t0d0s6
ufs fsck: sanity check: /dev/rdsk/c0t0d0s6 needs checking
```

▼ How to Check File Systems Interactively

1. **Become superuser or assume an equivalent role.**
2. **Unmount the local file systems except root (/) and `/usr`.**

   # **umountall -l**

3. **Check the file systems.**

   # **fsck**

   All file systems in the `/etc/vfstab` file with entries in the `fsck pass` field greater than `0` are checked. You can also specify the mount point directory or `/dev/rdsk/`*device-name* as arguments to the `fsck` command. Any inconsistency messages are displayed.

   For information about how to respond to the error message prompts while interactively checking one or more UFS file systems, see "Resolving UFS File System Inconsistencies (Tasks)" in *System Administration Guide: Advanced Administration*.

Caution – Running the `fsck` command on a mounted file system might cause a system to crash if the `fsck` command makes any changes, unless stated otherwise, such as running the `fsck` command in single-user mode to repair a file system.

4. **If you corrected any errors, type `fsck` and press Return.**

   The `fsck` command might be unable to fix all errors in one execution. If you see the message `FILE SYSTEM STATE NOT SET TO OKAY`, run the command again. If that does not work, see "Fixing a UFS File System That the `fsck` Command Cannot Repair" on page 597.

5. **Rename and move any files put in the `lost+found` directory.**

   Individual files put in the `lost+found` directory by the `fsck` command are renamed with their inode numbers. If possible, rename the files and move them where they belong. You might be able to use the `grep` command to match phrases with individual files and the `file` command to identify file types. When whole directories are put into the `lost+found` directory, it is easier to figure out where they belong and to move them back.

Example—Checking File Systems Interactively

The following example shows how to check the `/dev/rdsk/c0t0d0s6` file system and corrects the incorrect block count.

```
# fsck /dev/rdsk/c0t0d0s6
checkfilesys: /dev/rdsk/c0t0d0s6
** Phase 1 - Check Block and Sizes
INCORRECT BLOCK COUNT I=2529 (6 should be 2)
CORRECT? y

** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Cylinder Groups
929 files, 8928 used, 2851 free (75 frags, 347 blocks, 0.6%
fragmentation)
/dev/rdsk/c0t0d0s6 FILE SYSTEM STATE SET TO OKAY

***** FILE SYSTEM WAS MODIFIED *****
```

Preening UFS File Systems

The `fsck -o p` command (p is for preen) checks UFS file systems and automatically fixes the problems that normally result from an unexpected system shutdown. This command exits immediately if it encounters a problem that requires operator intervention. This command also permits parallel checking of file systems.

You can run the `fsck -o p` command to preen the file systems after an unclean shutdown. In this mode, the `fsck` command does not look at the clean flag and does a full check. These actions are a subset of the actions that the `fsck` command takes when it runs interactively.

▼ How to Preen a UFS File System

1. **Become superuser or assume an equivalent role.**
2. **Unmount the UFS file system.**

   # **umount** */mount-point*
3. **Check the UFS file system with the preen option.**

   # **fsck -o p /dev/rdsk/***device-name*

   You can preen individual file systems by using */mount-point* or `/dev/rdsk/`*device-name* as arguments to the `fsck` command.

Example—Preening a UFS File System

The following example shows how to preen the `/usr` file system.

```
# fsck -o p /usr
```

Fixing a UFS File System That the `fsck` Command Cannot Repair

Sometimes, you need to run the `fsck` command a few times to fix a file system because problems corrected on one pass might uncover other problems not found in earlier passes. The `fsck` command does not keep running until it comes up clean, so you must rerun it manually.

Pay attention to the information displayed by the `fsck` command. This information might help you fix the problem. For example, the messages might point to a damaged directory. If you delete the directory, you might find that the `fsck` command runs cleanly.

If the `fsck` command still cannot repair the file system, you can try to use the `fsdb`, `ff`, `clri`, and `ncheck` commands to figure out and fix what is wrong. For information about how to use these commands, see `fsdb`(1M), `ff`(1M), `clri`(1M), and `ncheck`(1M). You might, ultimately, need to re-create the file system and restore its contents from backup media.

For information about restoring complete file systems, see Chapter 49.

If you cannot fully repair a file system but you can mount it read-only, try using the `cp`, `tar`, or `cpio` commands to retrieve all or part of the data from the file system.

If hardware disk errors are causing the problem, you might need to reformat and divide the disk into slices again before re-creating and restoring file systems. Hardware errors usually display the same error again and again across different commands. The `format` command tries to work around bad blocks on the disk. If the disk is too severely damaged, however, the problems might persist, even after reformatting. For information about using the `format` command, see `format`(1M). For information about installing a new disk, see Chapter 34 or Chapter 35.

Restoring a Bad Superblock

When the superblock of a file system becomes damaged, you must restore it. The `fsck` command tells you when a superblock is bad. Fortunately, copies of the superblock are stored within a file system. You can use the `fsck -o b` command to replace the superblock with one of the copies.

For more information about the superblock, see "The Superblock" on page 610.

If the superblock in the root (/) file system becomes damaged and you cannot restore it, you have two choices:

- Reinstall the system
- Boot from the network or local CD, and attempt the following steps. If these steps fail, recreate the root (/) file system with the `newfs` command and restore it from a backup copy.

▼ How to Restore a Bad Superblock

1. **Become superuser or assume an equivalent role.**
2. **Determine whether the bad superblock is in the root (/) or `/usr` file system and select one of the following:**

   a. **Stop the system and boot from the network or a locally-connected CD if the bad superblock is in the root (/) or `/usr` file system.**

      From a locally-connected CD, use the following command:

      ```
      ok boot -s cdrom
      ```

      From the network where a boot or install server is already setup, use the following command:

      ```
      ok boot -s net
      ```

If you need help stopping the system, see "SPARC: How to Stop the System for Recovery Purposes" on page 186 or "x86: How to Stop a System for Recovery Purposes" on page 198.

b. Change to a directory outside the damaged file system and unmount the file system if the bad superblock is not in the root (/) or `/usr` file system.

```
# umount /mount-point
```

Caution – Be sure to use the `newfs -N` in the next step. If you omit the `-N` option, you will create a new, empty file system.

3. Display the superblock values with the `newfs -N` command.

```
# newfs -N /dev/rdsk/device-name
```

The output of this command displays the block numbers that were used for the superblock copies when the `newfs` command created the file system, unless the file system was created with special parameters. For information on creating a customized file system, see "Custom File System Parameters" on page 613.

4. Provide an alternate superblock with the `fsck` command.

```
# fsck -F ufs -o b=block-number /dev/rdsk/device-name
```

The `fsck` command uses the alternate superblock you specify to restore the primary superblock. You can always try `32` as an alternate block, or use any of the alternate blocks shown by the `newfs -N` command.

Example—Restoring a Bad Superblock

The following example shows how to restore the superblock copy `5264`.

```
# newfs -N /dev/rdsk/c0t3d0s7
/dev/rdsk/c0t3d0s7: 163944 sectors in 506 cylinders of 9 tracks, 36 sectors
 83.9MB in 32 cyl groups (16 c/g, 2.65MB/g, 1216 i/g)
super-block backups (for fsck -b #) at:
 32, 5264, 10496, 15728, 20960, 26192, 31424, 36656, 41888,
 47120, 52352, 57584, 62816, 68048, 73280, 78512, 82976, 88208,
 93440, 98672, 103904, 109136, 114368, 119600, 124832, 130064, 135296,
 140528, 145760, 150992, 156224, 161456,
# fsck -F ufs -o b=5264 /dev/rdsk/c0t3d0s7
Alternate superblock location: 5264.
** /dev/rdsk/c0t3d0s7
** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
```

```
36 files, 867 used, 75712 free (16 frags, 9462 blocks, 0.0% fragmentation)
/dev/rdsk/c0t3d0s7 FILE SYSTEM STATE SET TO OKAY

***** FILE SYSTEM WAS MODIFIED *****
#
```

Syntax and Options for the `fsck` Command

The `fsck` command checks and repairs inconsistencies in file systems. If you run the `fsck` command without any options, it interactively asks for confirmation before making repairs. This command has four options:

Command and Option	Description
`fsck -m`	Checks whether a file system can be mounted
`fsck -y`	Assumes a yes response for all repairs
`fsck -n`	Assumes a no response for all repairs
`fsck -o p`	Noninteractively preens the file system, fixing all expected (innocuous) inconsistencies, but exits when a serious problem is encountered

CHAPTER 44

UFS File System (Reference)

This is a list of the reference information in this chapter.

- "Default Directories for root (/) and /usr File Systems" on page 601
- "The Platform-Dependent Directories" on page 609
- "The Structure of Cylinder Groups for UFS File Systems" on page 609
- "Custom File System Parameters" on page 613
- "Commands for Creating a Customized File System" on page 617

Default Directories for root (/) and /usr File Systems

The /kernel directory contains only platform-independent objects, including a platform-independent kernel, genunix. For a description of /platform and /usr/platform, the platform-dependent directories, see Table 44–3.

The following table describes the directories that are contained in the root (/) file system.

TABLE 44–1 Default Directories in the root (/) File System

Directory	Description
/	Root of the overall file system name space
/dev	Primary location for logical device files
/dev/cfg	Symbolic links to physical ap_ids
/dev/cua	Device files for uucp

TABLE 44–1 Default Directories in the root (/) File System *(Continued)*

Directory	Description
/dev/dsk	Block disk devices
/dev/fbs	Frame buffer device files
/dev/fd	File descriptors
/dev/md	Volume management device names
/dev/printers	USB printer device files
/dev/pts	pty slave devices
/dev/rdsk	Raw disk devices
/dev/rmt	Raw tape devices
/dev/sad	Entry points for the STREAMS Administrative Driver
/dev/sound	Audio device and audio device control files
/dev/swap	Default swap device
/dev/term	Serial devices
/devices	Physical device files
/etc	Host-specific system administration configuration files and databases
/etc/acct	Accounting configuration information
/etc/apache	Apache configuration files
/etc/cron.d	Configuration information for cron
/etc/default	Defaults information for various programs
/etc/dfs	Configuration information for shared file systems
/etc/dhcp	Dynamic Host Configuration Protocol (DHCP) configuration files
/etc/dmi	Solstice Enterprise Agents configuration files
/etc/fn	Federated Naming Service and x.500 support files
/etc/fs	Binaries organized by file system types
/etc/ftpd	ftpd configuration files
/etc/gss	Generic Security Service (GSS) Application Program Interface configuration files
/etc/gtk	GNOME (GNU Network Object Model Environment) configuration files

TABLE 44–1 Default Directories in the root (/) File System *(Continued)*

Directory	Description
/etc/inet	Configuration files for Internet services
/etc/init.d	Scripts for changing run levels
/etc/iplanet	iPlanet configuration files
/etc/krb5	Kerberos configuration files
/etc/lib	Dynamic linking libraries that are needed when /usr is not available
/etc/llc2	Logical link control (llc2) driver configuration files
/etc/lp	Configuration information for the printer subsystem
/etc/lu	Solaris Live Upgrade configuration files
/etc/lvm	Solaris Volume Manager configuration files
/etc/mail	Mail subsystem configuration information
/etc/nca	Solaris Network Cache and Accelerator (NCA) configuration files
/etc/net	Configuration information for TI (transport- independent) network services
/etc/nfs	NFS server logging configuration file
/etc/openwin	OpenWindows configuration files
/etc/opt	Configuration information for optional packages
/etc/ppp	Solaris PPP configuration files
/etc/rc0.d	Scripts for entering or leaving run level 0
/etc/rc1.d	Scripts for entering or leaving run level 1
/etc/rc2.d	Scripts for entering or leaving run level 2
/etc/rc3.d	Scripts for entering or leaving run level 3
/etc/rcS.d	Scripts for bringing the system to single-user mode
/etc/rcm	Directory for reconfiguration manager (RCM) custom scripts
/etc/rpcsec	Might contain an NIS+ authentication configuration file
/etc/saf	Service access facility files (including FIFOs)
/etc/security	Basic Security Module (BSM) configuration files
/etc/sfw	Samba configuration files
/etc/skel	Default profile scripts for new user accounts

TABLE 44–1 Default Directories in the root (/) File System *(Continued)*

Directory	Description
/etc/smartcard	Solaris SmartCards configuration files
/etc/snmp	Solstice Enterprise Agents configuration files
/etc/ssh	Secure shell configuration files
/etc/sysevent	sysevent configuration files
/etc/tm	Trademark files, whose contents are displayed at boot time
/etc/usb	USB configuration information
/etc/uucp	uucp configuration information
/etc/wrsm	WCI Remote Shared Memory (WRSM) configuration information
/export	Default directory for users' home directories, client file systems, or other shared file systems
/home	Default directory or mount point for a user's home directory on a standalone system. When AutoFS is running, you cannot create any new entries in this directory.
/kernel	Directory of platform-independent loadable kernel modules that are required as part of the boot process. Includes the generic part of the core kernel that is platform-independent, /kernel/genunix. See Table 44–3 for the /platform and /usr/platform directory structure.
/mnt	Convenient, temporary mount point for file systems
/opt	Default directory or mount point for add-on application packages
/platform	Supported platform files. For more information, see Table 44–3.
/proc	Process information
/sbin	Essential executables used in the booting process and in manual system failure recovery
/tmp	Temporary files, whose contents are cleared during boot sequence
/usr	Mount point for the /usr file system. For more information, see Table 44–2.
/var	Directory for varying files, which usually includes temporary files, logging files, or status files
/var/adm	System logging files and accounting files

TABLE 44–1 Default Directories in the root (/) File System *(Continued)*

Directory	Description
/var/apache	Scripts, icons, logs, and cache pages for Apache web server
/var/audit	Basic Security Module (BSM) audit files
/var/crash	Default depository for kernel crash dumps
/var/cron	cron's log file
/var/dmi	Solstice Enterprise Agents Desktop Management Interface (DMI) run-time components
/var/dt	dtlogin configuration files
/var/inet	IPv6 router state files
/var/krb5	Database and log files for Kerberos
/var/ld	Configuration files for run-time linker
/var/ldap	LDAP client configuration files
/var/log	System log files
/var/lp	Line printer subsystem logging information
/var/mail	Directory where user mail is kept
/var/news	Community service messages. These messages are not the same as USENET-style news.
/var/nfs	NFS server log files
/var/nis	NIS+ databases
/var/ntp	Network Time Protocol (NTP) server state directory
/var/opt	Root of a subtree for varying files that are associated with software packages
/var/preserve	Backup files for vi and ex
/var/run	Temporary system files that are not needed across system reboots. A TMPFS-mounted directory.
/var/sadm	Databases that are maintained by the software package management utilities
/var/saf	saf (service access facility) logging files and accounting files
/var/samba	Log files and lock files for Samba
/var/snmp	SNMP status and configuration information
/var/spool	Directories for spooled temporary files

TABLE 44–1 Default Directories in the root (/) File System *(Continued)*

Directory	Description
/var/spool/clientmqueue	Sendmail client files
/var/spool/cron	cron and at spool files
/var/spool/locks	Spooling lock files
/var/spool/lp	Line printer spool files
/var/spool/mqueue	Mail queued for delivery
/var/spool/pkg	Spooled packages
/var/spool/print	LP print service client-side request staging area
/var/spool/samba	Samba print queue
/var/spool/uucp	Queued uucp jobs
/var/spool/uucppublic	Files deposited by uucp
/var/statmon	Network status monitor files
/var/tmp	Directory for temporary files that are not cleared during boot sequence
/var/uucp	uucp log files and status files
/var/yp	NIS databases

The following table describes the default directories in the /usr file system.

TABLE 44–2 Default Directories in the /usr File System

Directory	Description
4lib	SunOS 4.1 binary compatibility package libraries
5bin	Symbolic link to the /usr/bin directory
X	Symbolic link to the /usr/openwin directory
adm	Symbolic link to the /var/adm directory
apache	Apache executables, loadable modules, and documentation
aset	Directory for Automated Security Enhancement Tools (ASET) programs and files
bin	Location for standard system commands
ccs	C compilation programs and libraries
demo	Demo programs and data

TABLE 44–2 Default Directories in the `/usr` File System *(Continued)*

Directory	Description
`dict`	Symbolic link to the `/usr/share/lib/dict` directory, which contains the dictionary file used by the UNIX spell program
`dt`	Directory or mount point for CDE software
`games`	An empty directory, which is a remnant of the SunOS 4.0-4.1 software
`include`	Header files for C programs, and so on.
`iplanet`	Directory server executables, loadable modules, and documentation
`j2se`	Java 2 SDK executables, loadable modules, and documentation
`java*`	Directories that contain Java programs and libraries
`kernel`	Additional kernel modules
`kvm`	Obsolete
`lib`	Various program libraries, architecture-dependent databases, and binaries not invoked directly by the user
`local`	Commands local to a site
`mail`	Symbolic link to the `/var/mail` directory
`man`	Symbolic link to the `/usr/share/man` directory
`net`	Directory for network listener services
`news`	Symbolic link to the `/var/news` directory
`oasys`	Files for the Form and Menu Language Interpreter (FMLI) execution environment
`old`	Programs that are being phased out
`openwin`	Directory or mount point for OpenWindows software
`perl5`	Perl 5 programs and documentation
`platform`	Supported platform files. For more information, see Table 44–3.
`preserve`	Symbolic link to the `/var/preserve` directory
`proc`	Directory for the `proc` tools
`pub`	Files for online man page and character processing

TABLE 44–2 Default Directories in the /usr File System *(Continued)*

Directory	Description
sadm	Various files and directories related to system administration
sbin	Executables for system administration
sbin/install.d	Custom JumpStart scripts and executables
sbin/static	Statically linked version of selected programs from /usr/bin and /usr/sbin
sbin/sparc7 and sparc9	32-bit and 64–bit versions of commands
sfw	GNU and open source executables, libraries, and documentation
share	Architecture-independent sharable files
share/admserv5.1	iPlanet Console and Administration Server 5.0 documentation
share/audio	Sample audio files
share/ds5	iPlanet Directory Server 5.1 Documentation
share/lib	Architecture-independent databases
share/man	Solaris manual pages
share/src	Source code for kernel, libraries, and utilities
snadm	Programs and libraries related to system and network administration
spool	Symbolic link to the /var/spool directory
src	Symbolic link to the share/src directory
tmp	Symbolic link to the var/tmp directory
ucb	Berkeley compatibility package binaries
ucbinclude	Berkeley compatibility package header files
ucblib	Berkeley compatibility package libraries
vmsys	Directory for Framed Access Command Environment (FACE) programs
xpg4	Directory for POSIX-compliant utilities

The Platform-Dependent Directories

The following table describes the platform-dependent objects in the `/platform` and `/usr/platform` directories.

TABLE 44–3 The `/platform` and `/usr/platform` Directories

Directory	Description
`/platform`	Contains a series of directories, one directory per supported platform that needs to reside in the root (`/`) file system.
`/platform/*/kernel`	Contains platform-dependent kernel components, including the file `unix`, the core kernel that is platform–dependent. For more information, see `kernel`(1M).
`/usr/platform`	Contains platform-dependent objects that do not need to reside in the root (`/`) file system.
`/usr/platform/*/lib`	Contains platform-dependent objects similar to those objects found in the `/usr/lib` directory.
`/usr/platform/*/sbin`	Contains platform-dependent objects similar to those objects found in the `/usr/sbin` directory.

The Structure of Cylinder Groups for UFS File Systems

When you create a UFS file system, the disk slice is divided into *cylinder groups*, which is made up of one or more consecutive disk cylinders. The cylinder groups are then further divided into addressable blocks to control and organize the structure of the files within the cylinder group. Each type of block has a specific function in the file system. A UFS file system has these four types of blocks:

Block Type	Type of Information Stored
Boot block	Information used when booting the system
Superblock	Detailed information about the file system
Inode	All information about a file

Block Type	Type of Information Stored
Storage or data block	Data for each file

The following sections provide additional information about the organization and function of these blocks.

The Boot Block

The boot block stores objects that are used in booting the system. If a file system is not to be used for booting, the boot block is left blank. The boot block appears only in the first cylinder group (cylinder group 0) and is the first 8 Kbytes in a slice.

The Superblock

The superblock stores much of the information about the file system, which includes the following:

- Size and status of the file system
- Label, which includes file system name and volume name
- Size of the file system logical block
- Date and time of the last update
- Cylinder group size
- Number of data blocks in a cylinder group
- Summary data block
- File system state
- Path name of the last mount point

Superblocks are distributed throughout the file system, one in each cylinder group. Because the superblock contains critical data, multiple superblocks are made when the file system is created.

A summary information block is kept within the superblock. The summary information block is not replicated, but is grouped with the primary superblock, usually in cylinder group 0. The summary block records changes that take place as the file system is used. In addition, the summary block lists the number of inodes, directories, fragments, and storage blocks within the file system.

Inodes

An inode contains all the information about a file except its name, which is kept in a directory. An inode is 128 bytes. The inode information is kept in the cylinder information block, and contains the following:

- The type of the file:
  - Regular
  - Directory
  - Block special
  - Character special
  - FIFO, also known as named pipe
  - Symbolic link
  - Socket
  - Other inodes – attribute directory and shadow (used for ACLs)
- The mode of the file (the set of read-write-execute permissions)
- The number of hard links to the file
- The user ID of the owner of the file
- The group ID to which the file belongs
- The number of bytes in the file
- An array of 15 disk-block addresses
- The date and time the file was last accessed
- The date and time the file was last modified
- The date and time the file was created

The array of 15 disk addresses (0 to 14) points to the data blocks that store the contents of the file. The first 12 are direct addresses. That is, they point directly to the first 12 logical storage blocks of the file contents. If the file is larger than 12 logical blocks, the 13th address points to an indirect block, which contains direct block addresses instead of file contents. The 14th address points to a double indirect block, which contains addresses of indirect blocks. The 15th address is for triple indirect addresses. The following figure shows this chaining of address blocks starting from the inode.

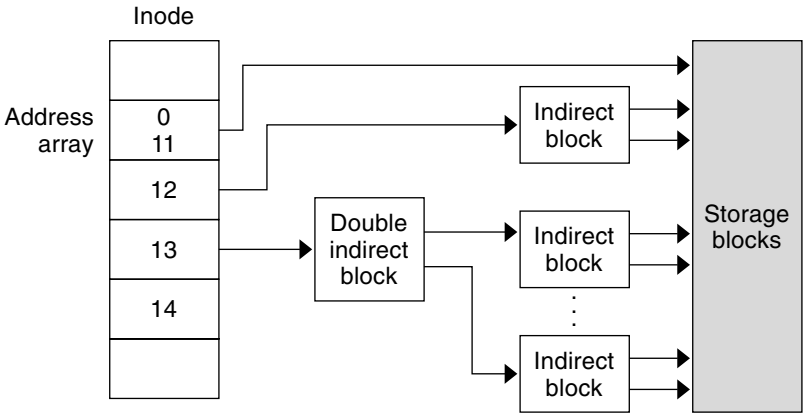


FIGURE 44–1 Address Chain for a UFS File System

Data Blocks

Data blocks, also called storage blocks, contain the rest of the space that is allocated to the file system. The size of these data blocks is determined at the time a file system is created. Data blocks are allocated, by default, in two sizes: an 8-Kbyte logical block size, and a 1-Kbyte fragment size.

For a regular file, the data blocks contain the contents of the file. For a directory, the data blocks contain entries that give the inode number and the file name of the files in the directory.

Free Blocks

Blocks that are not currently being used as inodes, as indirect address blocks, or as storage blocks are marked as free in the cylinder group map. This map also keeps track of fragments to prevent fragmentation from degrading disk performance.

To give you an idea of the appearance of a typical UFS file system, the following figure shows a series of cylinder groups in a generic UFS file system.

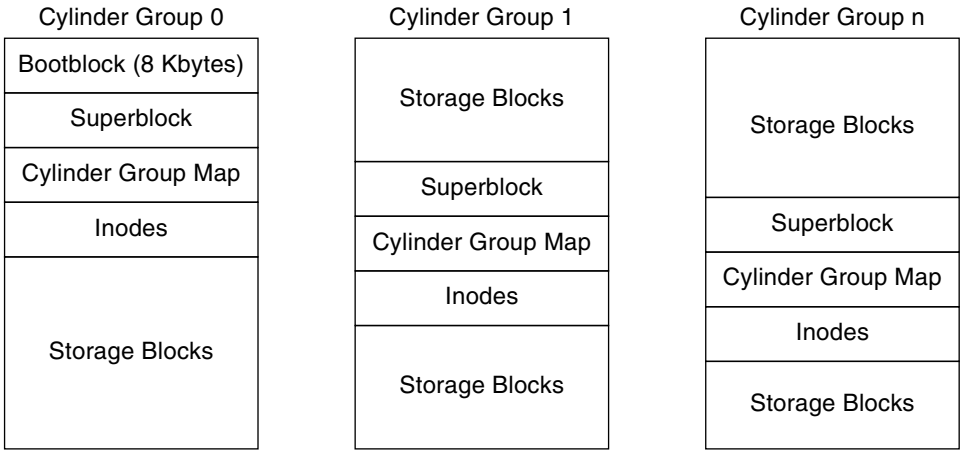


FIGURE 44–2 A Typical UFS File System

Custom File System Parameters

Before you choose to alter the default file system parameters that are assigned by the `newfs` command, you need to understand them. This section describes each of these parameters:

- Logical block size
- Fragment size
- Minimum free space
- Rotational delay
- Optimization type
- Number of files

Logical Block Size

The logical block size is the size of the blocks that the UNIX kernel uses to read or write files. The logical block size is usually different from the physical block size. The physical block size is usually 512 bytes, which is the size of the smallest block that the disk controller can read or write.

Logical block size is set to the page size of the system by default. The default logical block size is 8192 bytes (8 Kbytes) for UFS file systems. The UFS file system supports block sizes of 4096 or 8192 bytes (4 or 8 Kbytes). The recommended logical block size is 8 Kbytes.

SPARC only – You can specify only the 8192–byte block size on the sun4u platform.

To choose the best logical block size for your system, consider both the performance desired and the available space. For most UFS systems, an 8-Kbyte file system provides the best performance, offering a good balance between disk performance and the use of space in primary memory and on disk.

As a general rule, to increase efficiency, use a larger logical block size for file systems where most of the files are very large. Use a smaller logical block size for file systems where most of the files are very small. You can use the `quot -c` *file-system* command on a file system to display a complete report on the distribution of files by block size.

However, the page size set when the file system is created is probably best in most cases.

Fragment Size

As files are created or expanded, they are allocated disk space in either full logical blocks or portions of logical blocks called *fragments*. When disk space is needed for a file, full blocks are allocated first, and then one or more fragments of a block are allocated for the remainder. For small files, allocation begins with fragments.

The ability to allocate fragments of blocks to files, rather than just whole blocks, saves space by reducing *fragmentation* of disk space that results from unused holes in blocks.

You define the *fragment size* when you create a UFS file system. The default fragment size is 1 Kbyte. Each block can be divided into 1, 2, 4, or 8 fragments, which results in fragment sizes from 8192 bytes to 512 bytes (for 4-Kbyte file systems only). The lower bound is actually tied to the disk sector size, typically 512 bytes.

Note – The upper bound for the fragment is the logical block size, in which case the fragment is not a fragment at all. This configuration might be optimal for file systems with very large files when you are more concerned with speed than with space.

When choosing a fragment size, look at the trade-off between time and space: a small fragment size saves space, but requires more time to allocate. As a general rule, to increase storage efficiency, use a larger fragment size for file systems where most of the files are large. Use a smaller fragment size for file systems where most of the files are small.

Minimum Free Space

The *minimum free space* is the percentage of the total disk space that is held in reserve when you create the file system. The default reserve is ((64 Mbytes/partition size) * 100), rounded down to the nearest integer and limited between 1 percent and 10 percent, inclusively.

Free space is important because file access becomes less and less efficient as a file system gets full. As long as an adequate amount of free space exists, UFS file systems operate efficiently. When a file system becomes full, using up the available user space, only root can access the reserved free space.

Commands such as `df` report the percentage of space that is available to users, excluding the percentage allocated as the minimum free space. When the command reports that more than 100 percent of the disk space in the file system is in use, some of the reserve has been used by root.

If you impose quotas on users, the amount of space available to the users does not include the reserved free space. You can change the value of the minimum free space for an existing file system by using the `tunefs` command.

Rotational Delay

This parameter is obsolete for modern disks. If you need to use this parameter, the default value provided when the file system is created is probably best for most cases.

Optimization Type

The *optimization type* parameter is set to either *space* or *time*.

- **Space –** When you select space optimization, disk blocks are allocated to minimize fragmentation and disk use is optimized.
- **Time –** When you select time optimization, disk blocks are allocated as quickly as possible, with less emphasis on their placement. When there is enough free space, it is relatively easy to allocate disk blocks effectively, without resulting in too much fragmentation. The default is *time*.

  You can change the value of the optimization type parameter for an existing file system by using the `tunefs` command.

For more information, see `tunefs`(1M).

Number of Files

The number of inodes parameter determines the number of files you can have in the file system: one inode for each file. The *number of bytes per inode* determines the total number of inodes that are created when the file system is made: the total size of the file system divided by the number of bytes per inode. Once the inodes are allocated, you cannot change the number without re-creating the file system.

The default number of bytes per inode is 2048 bytes (2 Kbytes) if the file system is less than one Gbyte. If the file system is larger than one Gbyte, the following formula is used:

File System Size	Number of Bytes Per Inode
Less than or equal to 1 Gbyte	2048
Less than 2 Gbytes	4096
Less than 3 Gbytes	6144
3 Gbytes or greater	8192

If you have a file system with many symbolic links, they can lower the average file size. If your file system is going to have many small files, you can give this parameter a lower value. Note, however, that having too many inodes is much better than running out of inodes. If you have too few inodes, you could reach the maximum number of files on a disk slice that is practically empty.

Maximum UFS File and File System Size

The maximum size of a UFS file system is approximately one Tbyte of usable space, minus approximately one percent overhead. A *sparse* file can have a logical size of one Tbyte. However, the actual amount of data that can be stored in a file is approximately one percent less than one Tbyte because of the file system overhead.

Maximum Number of UFS Subdirectories

The maximum number of subdirectories per directory in a UFS file system is 32,767. This limit is predefined and cannot be changed.

Commands for Creating a Customized File System

This section describes the two commands that you use to create a customized file system:

- `newfs`
- `mkfs`

The `newfs` Command Syntax, Options, and Arguments

The `newfs` command is a friendlier version of the `mkfs` command that is used to create file systems.

The syntax is as follows:

`/usr/sbin/newfs [-Nv] [`*mkfs_options*`] `*raw_device*

The following table describes the options and arguments for the `newfs` command.

TABLE 44–4 The `newfs` Command Options and Arguments

Option	Description
`-N`	Displays the file system parameters that would be used in creating the file system without actually creating it. This option does not display the parameters that were used to create an existing file system.
`-v`	Displays the parameters that are passed to the `mkfs` command.
mkfs-options	Use the options in this table (from `-s size` to `-C maxcontig` to set the `mkfs` parameters. The options are listed in the order they are passed to the `mkfs` command. Separate the options with spaces.
`-s` *size*	The size of the file system in sectors. The default is automatically determined from the disk label.
`-t` *ntrack*	The number of tracks per cylinder on the disk. The default is determined from the disk label.

TABLE 44–4 The `newfs` Command Options and Arguments *(Continued)*

Option	Description
`-b` *bsize*	The logical block size in bytes to use for data transfers. Specify the size of 4096 or 8192 (4 or 8 Kbytes). The default is 8192 bytes (8 Kbytes).
`-f` *fragsize*	The smallest amount of disk space in bytes that is allocated to a file. Specify the fragment size in powers of two in the range from 512 to 8192 bytes. The default is 1024 bytes (1 Kbyte).
`-c` *cgsize*	The number of disk cylinders per cylinder group. The default value is calculated by dividing the number of sectors in the file system by the number of sectors in 1 Gbyte, and then multiplying the result by 32. The default value ranges from 16 to 256.
`-m` *free*	The minimum percentage of free disk space to allow. The default is ((64 Mbytes/partition size) * 100), rounded down to the nearest integer and limited between 1% and 10%, inclusively.
`-r` *rpm*	The speed of the disk, in revolutions per minute. This setting is driver- or device-specific. If the drive can report how fast it spins, the `mkfs` command uses this value. If not, the default is 3600. This parameter is converted to revolutions per second before it is passed to the `mkfs` command.
`-i` *nbpi*	The number of bytes per inode to use in computing how many inodes to create. For the default values, see "Number of Files" on page 616.
`-o` *opt*	Optimization type to use for allocating disk blocks to files: `space` or `time`. The default is `time`.
`-a` *apc*	The number of alternate blocks per disk cylinder (SCSI devices only) to reserve for bad block placement. The default is 0.
`-d` *gap*	(Rotational delay) The expected minimum number of milliseconds it takes the CPU to complete a data transfer and initiate a new data transfer on the same disk cylinder. The default is `0`.
`-n` *nrpos*	The number of different rotation positions in which to divide a cylinder group. The default is 8.

TABLE 44–4 The `newfs` Command Options and Arguments *(Continued)*

Option	Description
`-C` *maxcontig*	The maximum number of blocks, belonging to one file, that will be allocated contiguously before inserting a rotational delay. The default varies from drive to drive. Drives without internal (track) buffers (or drives or controllers that don't advertise the existence of an internal buffer) default to 1. Drives with buffers default to 7. This parameter is limited in the following way: *blocksize* x *maxcontig* must be <= *maxphys* *maxphys* is a read-only kernel variable that specifies the maximum block transfer size (in bytes) that the I/O subsystem is capable of satisfying. This limit is enforced by the `mount` command, not by `newfs` or `mkfs` command. This parameter also controls clustering. Regardless of the value of *rotdelay*, clustering is enabled only when *maxcontig* is greater than 1. Clustering allows higher I/O rates for sequential I/O and is described in `tunefs`(1M).
raw_device	The special character (raw) device file name of the partition that will contain the file system. This argument is required.

Examples—`newfs` Command Options and Arguments

This example shows how to use the `-N` option to display file system information, including the backup superblocks.

```
# newfs -N /dev/rdsk/c0t0d0s0
/dev/rdsk/c0t0d0s0:  37260 sectors in 115 cylinders of 9 tracks, 36 sectors
        19.1MB in 8 cyl groups (16 c/g, 2.65MB/g, 1216 i/g)
superblock backups (for fsck -b #) at:
 32, 5264, 10496, 15728, 20960, 26192, 31424, 36656,
#
```

The Generic `mkfs` Command

The generic `mkfs` command calls a file system-specific `mkfs` command, which then creates a file system of a specified type on a specified disk slice. Although the `mkfs` command can support different types of file systems, in practice you would use it to create UFS, UDFS, or PCFS file systems. To make other types of file systems, you would have to write the software for the file system–specific versions of the `mkfs` command to use. Normally, you do not run the `mkfs` command directly. The `mkfs` command is called by the `newfs` command.

The generic `mkfs` command is located in the `/usr/sbin` directory. For a description of the arguments and options, see `mkfs`(1M).

CHAPTER 45

Backing Up and Restoring Files and File Systems Topics

This topic map lists the chapters that provide information on backing up and restoring files and file systems.

Chapter 46	Provides guidelines and planning information on backing up and restoring files and file systems.
Chapter 47	Provides step-by-step instructions for backing up individual files and complete file systems from local devices or remote devices.
Chapter 48	Provides step-by-step instructions for creating snapshots of UFS file systems.
Chapter 49	Provides step-by-step instructions for restoring individual files and complete file systems.
Chapter 50	Describes how the `ufsdump` command works, and the syntax and options for the `ufsdump` and `ufsrestore` commands.
Chapter 51	Provides step-by-step instructions for for using the `dd`, `pax`, `cpio`, and `tar` commands with different backup media, and for copying files with a different header format.
Chapter 52	Provides step-by-step instructions for how to add a tape device, how to determine the type of tape device and backup device names, and working with magnetic tape cartridges.

CHAPTER 46

Backing Up and Restoring File Systems (Overview)

This chapter provides guidelines and planning information on the backing up and restoring of file systems by using the `ufsdump` and `ufsrestore` commands.

This is a list of the overview information in this chapter.

- "What's New in Backing Up and Restoring File Systems?" on page 623
- "Where to Find Backup and Restore Tasks" on page 624
- "Definition: Backing Up and Restoring File Systems" on page 624
- "Why You Should Back Up File Systems" on page 625
- "Planning Which File Systems to Back Up" on page 625
- "Choosing the Type of Backup" on page 627
- "Choosing a Tape Device" on page 628
- "High-Level View of Backing Up and Restoring File Systems (Task Map)" on page 629
- "Guidelines for Scheduling Backups" on page 630
- "Sample Backup Schedules" on page 632

What's New in Backing Up and Restoring File Systems?

This section describes new backup and restore features in the Solaris 9 release.

UFS Snapshots

The Solaris 9 release includes the `fssnap` command for the backing up of file systems while the file system is mounted.

You can use the `fssnap` command to create a read-only snapshot of a file system. A *snapshot* is a file system's temporary image that is intended for backup operations.

For more information, see Chapter 48.

Where to Find Backup and Restore Tasks

Backup or Restore Task	For More Information
Back up file systems with the `ufsdump` command	Chapter 47
Create UFS snapshots with the `fssnap` command	Chapter 48
Restore file systems with the `ufsrestore` command	Chapter 49
Copy files and directories with the `cpio`, `dd`, `pax`, and `cpio` commands	Chapter 51

Definition: Backing Up and Restoring File Systems

Backing up file systems means the copying of file systems to removable media, such as tape, to safeguard against loss, damage, or corruption. Restoring file systems means the copying of reasonably current backup files from removable media to a working directory.

This chapter describes the `ufsdump` and `ufsrestore` commands for backing up and restoring UFS file systems. Other commands are available for copying files and file systems for the purpose of sharing or transporting files. The following table provides pointers to all commands that copy individual files and file systems to media.

TABLE 46–1 Commands for Backing Up and Restoring Files and File Systems

Task	Command	For More Information
Back up one or more file systems to a local tape device or a remote tape device	`ufsdump`	Chapter 47 or Chapter 50
Create read-only copies of file systems	`fssnap`	Chapter 48
Back up all file systems for systems on a network from a backup server	Solstice Backup™ software	*Solstice Backup 5.1 Administration Guide*
Back up and restore an NIS+ master server	`nisbackup` and `nisrestore`	*System Administration Guide: Naming and Directory Services (FNS and NIS+)*
Copy, list, and retrieve files on tape or diskette	`tar`, `cpio`, or `pax`	Chapter 51
Copy master disk to a clone disk	`dd`	Chapter 51
Restore complete file systems or individual files from removable media to a working directory	`ufsrestore`	Chapter 49

Why You Should Back Up File Systems

Backing up files is one of the most crucial system administration functions. You should perform regularly scheduled backups to prevent loss of data due to the following:

- System crashes
- Accidental deletion of files
- Hardware failures
- Natural disasters such as fire, hurricanes, or earthquakes
- Problems when you reinstall or upgrade a system

Planning Which File Systems to Back Up

You should back up all file systems that are critical to users, including file systems that change frequently. The following tables provide general guidelines on the file systems to back up for standalone systems and servers.

TABLE 46–2 File Systems to Back Up for Standalone Systems

File System to Back Up	Description	Back Up Interval
root (/) – slice 0	This file system contains the kernel and possibly contains the /var directory. The /var directory might include frequently modified files such as mail and accounting files.	At regular intervals such as weekly or daily.
/usr – slice 6, /opt	The installation of new software and adding new commands typically affects the /usr and /opt file systems. The /opt directory is either part of root (/) or is its own file system.	Occasionally.
/export/home – slice 7	This file system contains the directories and subdirectories of all users on the standalone system.	More often than root (/) or /usr, perhaps as often as once a day, depending on your site's needs.
/export , /var, or other file systems	During installation of Solaris software, you might have created these file systems.	As your site requires.

TABLE 46–3 File Systems to Back Up for Servers

File System to Back Up	Description	Back Up Interval
root (/) – slice 0	This file system contains the kernel and executables.	Once a day to once a month depending on your site's needs. If you frequently add and remove users and systems on the network, you have to change configuration files in this file system. In this case, you should do a full backup of the root (/) file system at interverals between once a week and once a month. If your site keeps user mail in the /var/mail directory on a mail server, which client systems then mount, you might want to back up root (/) daily. Or, backup the /var directory, if it is a separate file system.

TABLE 46–3 File Systems to Back Up for Servers *(Continued)*

File System to Back Up	Description	Back Up Interval
`/export` – slice 3	This file system can contain the kernel and executables for diskless clients.	Once a day to once a month depending on your site's needs. Because the information in this file system is similar to the server's root directory in slice 0, the file system does not change frequently. You need to back up this file system only occasionally, unless your site delivers mail to client systems. Then, you should back up `/export` more frequently.
`/usr` – slice 6, `/opt`		Once a day to once a month depending on your site's needs. These file systems are fairly static and need to be backed up once a week to once a month.
`/export/home` – slice 7	This file system contains the home directories of all the users on the system. The files in this file system are volatile.	Once a day to once a week.

Choosing the Type of Backup

You can perform full or incremental backups with the `ufsdump` command. You can create a temporary image of a file system with the `fssnap` command. The following table lists the differences between these types of backup procedures.

TABLE 46–4 Differences Between Types of Backups

Backup Type	Result	Advantages	Disadvantages
Full	Copies a complete file system or directory	All data is in one place	Requires large numbers of backup tapes that take a long time to write. Takes longer to retrieve individual files because the drive has to move sequentially to the point on the tape where the file is located. You might have to search multiple tapes.
Snapshot	Creates a temporary image of a file system	System can be in multiuser mode	System performance might degrade while the snapshot is created.
Incremental	Copies only those files in the specified file system that have changed since a previous backup	Easier to retrieve small changes in file systems	Finding which incremental tape contains a file can take time. You might have to go back to last full dump.

Choosing a Tape Device

The following table shows typical tape devices that are used for storing file systems during the backup process. The capacity depends on the type of drive and the data being written to the tape. For more detailed information on tape devices, see Chapter 52.

TABLE 46–5 Typical Media for Backing Up File Systems

Media	Capacity
1/2-inch reel tape	140 Mbytes (6250 bpi)
2.5-Gbyte 1/4 inch cartridge (QIC) tape	2.5 Gbytes
DDS3 4-mm cartridge tape (DAT)	12–24 Gbytes
14-Gbyte 8-mm cartridge tape	14 Gbytes
DLT 7000 1/2-inch cartridge tape	35–70 Gbytes

High-Level View of Backing Up and Restoring File Systems (Task Map)

Use this task map to identify all the tasks for the backing up and restoring of file systems. Each task points to a series of additional tasks such as determining the type of backup to perform.

Task	Description	For Instructions
1. Identify the file systems to back up	Identify which file systems need to be backed up on a daily, weekly, or monthly basis.	"Planning Which File Systems to Back Up" on page 625
2. Determine the type of backup	Determine the type of backup you need for the file systems at your site.	"Choosing the Type of Backup" on page 627
3. Create the backup	Use one of the following methods: If you want to have full and incremental backups of your file systems, use the `ufsdump` command. If you would like to create a snapshot of file system while it is active and mounted, consider using the `fssnap` command. If you just want to have full backups of your personal home directory or smaller, less-important file systems, use the `tar`, `cpio`, or `pax` commands.	 Chapter 47 Chapter 48 Chapter 51
4. Restore a file system	(Optional) Select the restoration method that is based on the command used to back up the files or file system.	
	Restore a file system backup that was created with the `ufsdump` command.	Chapter 49
	Restore a file system that was created with the `tar`, `cpio`, or `pax` command.	Chapter 51

Task	Description	For Instructions
5. Restore the root (/) or /usr file system	Optional Restoring the root (/) or /usr file system is more complicated than restoring a non-critical file system because you need to boot from a local CD or from the network while these file systems are being restored.	"How to Restore the root (/) and /usr File Systems" on page 670

Guidelines for Scheduling Backups

A *backup schedule* is the schedule that you establish to run the `ufsdump` command. This section discusses guidelines on the factors to weigh when you create a backup schedule and how often to back up file systems. This section also includes sample backup schedules.

The backup schedule that you create depends on the following:

- Your need to minimize the number of tapes that are used for backups
- Time available for doing backups
- Time available for doing a full restore of a damaged file system
- Time available for retrieving individual files that are accidentally deleted

How Often Should You Do Backups?

If you do not need to minimize time and the amount of media that is used for backups, you can do full backups every day. However, this backup method is not realistic for most sites, so incremental backups are used most often. In this case, you should back up your site enough to restore files from the last four weeks. This schedule requires at least four sets of tapes—one for each week, which you would reuse each month. In addition, you should archive the monthly backups for at least a year, and then keep yearly backups for a number of years.

Using Dump Levels to Create Incremental Backups

The dump level you specify in the `ufsdump` command (0-9) determines which files are backed up. Dump level 0 creates a full backup. Levels 1–9 are used to schedule incremental backups, but have *no defined meanings*. Levels 1–9 are just a range of numbers that are used to schedule cumulative or discrete backups. The only meaning levels 1–9 have is in relationship to each other, as a higher or lower number.

The following examples show the flexibility of the incremental dump procedure using levels 1–9.

Dump Levels for Daily, Cumulative Backups

Doing daily, cumulative incremental backups is the most commonly used backup scheme and is recommended for most situations. The following example shows a schedule that uses a level 9 dump each day, and a level 5 dump on Friday to restart the process.

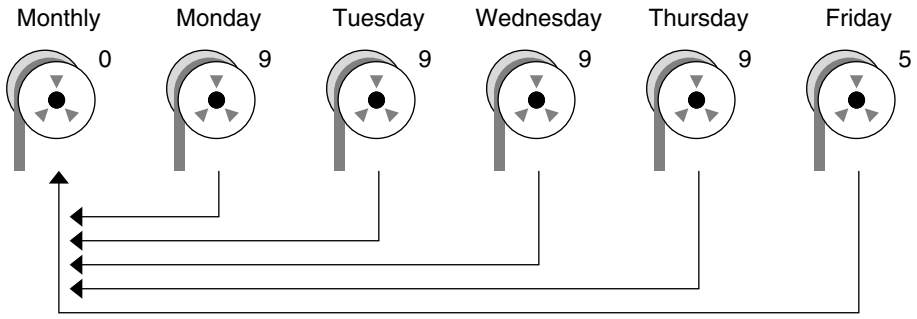


FIGURE 46–1 Incremental Backup: Daily Cumulative

In the preceding example, you could have used other numbers in the 1–9 range to produce the same results. The key is having the same number each day, with any *lower* number on Friday. For example, you could have specified levels 4, 4, 4, 4, 2 or 7, 7, 7, 7, 5.

Dump Levels for Daily, Discrete Backups

The following example shows a schedule where you capture only a day's work on different tapes. In this case, sequential dump level numbers are used during the week (3,4,5,6) with a lower number (2) on Friday.

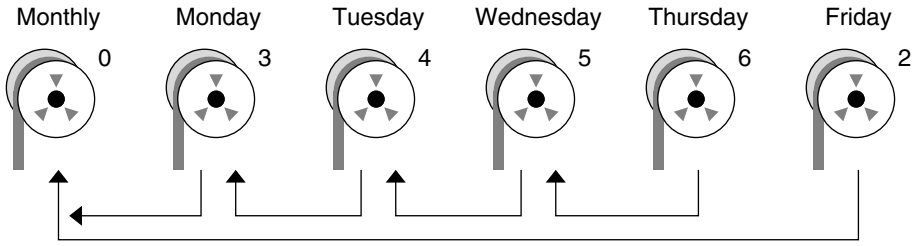


FIGURE 46–2 Incremental Backup: Daily Discrete

In the preceding example, you could have used the sequence 6, 7, 8, 9 followed by 2, or 5, 6, 7, 8 followed by 3. Remember, the numbers themselves have no defined meaning. You attribute meaning by ordering them in a high/low sequence.

Sample Backup Schedules

This section provides sample backup schedules. All schedules assume that you begin with a full backup (dump level 0), and that you use the `-u` option to record each backup.

Example—Daily Cumulative, Weekly Cumulative Backups

The following table shows the most commonly used incremental backup schedule. This schedule is recommended for most situations. With this schedule, the following occurs:

- All files that have changed since the lower-level backup at the end of the previous week are saved each day.
- For each weekday level 9 backup, the previous level 0 or level 5 is the closest backup at a lower level. Therefore, each weekday tape contains all the files that changed since the end of the previous week or the initial level 0 for the first week.
- For each Friday level 5 backup, the nearest lower-level backup is the level 0 done at the beginning of the month. Therefore, each Friday's tape contains all the files changed during the month to that point.

TABLE 46–6 Daily Cumulative or Weekly Cumulative Backup Schedule

	Floating	Mon	Tues	Wed	Thurs	Fri
1st of Month	0					
Week 1		9	9	9	9	5
Week 2		9	9	9	9	5
Week 3		9	9	9	9	5
Week 4		9	9	9	9	5

The following table shows how the contents of the tapes can change across two weeks with the daily cumulative, weekly cumulative schedule. Each letter represents a different file.

TABLE 46–7 Contents of Tapes for Daily Cumulative/Weekly Cumulative Backup Schedule

	Mon	Tues	Wed	Thurs	Fri
Week 1	a b	a b c	a b c d	a b c d e	a b c d e f
Week 2	g	g h	g h i	g h i j	a b c d e f g h i j k

Tape Requirements for the Daily Cumulative, Weekly Cumulative Schedule

With this schedule, you need six tapes if you want to reuse daily tapes, or nine tapes if you want to use four different daily tapes: one tape for the level 0, four tapes for Fridays, and one or four daily tapes.

If you need to restore a complete file system, you need the following tapes: the level 0, the most recent Friday tape, and the most recent daily tape since the last Friday tape, if any.

Example—Daily Cumulative, Weekly Incremental Backups

The following table shows a schedule where each weekday tape accumulates all files that changed since the beginning of the week, or the initial level 0 for the first week, and each Friday's tape contains all the files that changed that week.

TABLE 46–8 Daily Cumulative, Weekly Incremental Backup Schedule

	Floating	Mon	Tues	Wed	Thurs	Fri
1st of Month	0					
Week 1		9	9	9	9	3
Week 2		9	9	9	9	4
Week 3		9	9	9	9	5
Week 4		9	9	9	9	6

The following table shows how the contents of the tapes can change across two weeks with the daily cumulative, weekly incremental backup schedule. Each letter represents a different file.

TABLE 46–9 Contents of Tapes for Daily Cumulative, Weekly Incremental Backup Schedule

	Mon	Tues	Wed	Thurs	Fri
Week 1	a b	a b c	a b c d	a b c d e	a b c d e f
Week 2	g	g h	g h i	g h i j	g h i j k

Tape Requirements for the Daily Cumulative, Weekly Incremental Backup Schedule

With this schedule, you need six tapes, if you want to reuse daily tapes, or nine tapes, if you want to use four different daily tapes: one tape for the level 0, four tapes for Fridays, and one or four daily tapes.

If you need to restore a complete file system, you need the following tapes: the level 0, all the Friday tapes, and the most recent daily tape since the last Friday tape, if any.

Example—Daily Incremental, Weekly Cumulative Backups

The following table shows a schedule where each weekday tape contains only the files that changed since the previous day, and each Friday's tape contains all files changed since the initial level 0 at the beginning of the month.

TABLE 46–10 Daily Incremental, Weekly Cumulative Backup Schedule

	Floating	Mon	Tues	Wed	Thurs	Fri
1st of Month	0					
Week 1		3	4	5	6	2
Week 2		3	4	5	6	2
Week 3		3	4	5	6	2
Week 4		3	4	5	6	2

The following table shows how the contents of the tapes can change across two weeks with the daily incremental, weekly cumulative schedule. Each letter represents a different file.

TABLE 46–11 Contents of Tapes for Daily Incremental, Weekly Cumulative Backup Schedule

	Mon	Tues	Wed	Thurs	Fri
Week 1	a b	c d	e f g	hi	a b c d e f g h i
Week 2	j k l	m	n o	p q	a b c d e f g h i j k l m n o p q r s

Tape Requirements for Daily Incremental, Weekly Cumulative Schedule

With this schedule, you need at least nine tapes if you want to reuse daily tapes, which is not recommended. Or, you need 21 tapes if you save weekly tapes for a month: one tape for the level 0, four tapes for the Fridays, and four or 16 daily tapes.

If you need to restore the complete file system, you need the following tapes: the level 0, the most recent Friday tape, and all the daily tapes since the last Friday tape, if any.

Example—Monthly Backup Schedule for a Server

The following table shows an example backup strategy for a heavily used file server on a small network where users are doing file-intensive work, such as program development or document production. This example assumes that the backup period begins on a Sunday and consists of four seven-day weeks.

TABLE 46–12 Example of Monthly Backup Schedule for a Server

Directory	Date	Level	Tape Name
root (/)	1st Sunday	0	*n* tapes
/usr	1st Sunday	0	*n* tapes
/export	1st Sunday	0	*n* tapes
/export/home	1st Sunday	0	*n* tapes
	1st Monday	9	A
	1st Tuesday	9	B
	1st Wednesday	5	C
	1st Thursday	9	D
	1st Friday	9	E
	1st Saturday	5	F
root (/)	2nd Sunday	0	*n* tapes
/usr	2nd Sunday	0	"
/export	2nd Sunday	0	"
/export/home	2nd Sunday	0	"
	2nd Monday	9	G
	2nd Tuesday	9	H
	2nd Wednesday	5	I
	2nd Thursday	9	J
	2nd Friday	9	K
	2nd Saturday	5	L
root (/)	3rd Sunday	0	*n* tapes
/usr	3rd Sunday	0	"
/export	3rd Sunday	0	"
/export/home	3rd Sunday	0	"
	3rd Monday	9	M
	3rd Tuesday	9	N
	3rd Wednesday	5	O
	3rd Thursday	9	P

TABLE 46–12 Example of Monthly Backup Schedule for a Server *(Continued)*

Directory	Date	Level	Tape Name
	3rd Friday	9	Q
	3rd Saturday	5	R
root (/)	4th Sunday	0	*n* tapes
/usr	4th Sunday	0	"
/export	4th Sunday	0	"
/export/home	4th Sunday	0	"
	4th Monday	9	S
	4th Tuesday	9	T
	4th Wednesday	5	U
	4th Thursday	9	V
	4th Friday	9	W
	4th Saturday	5	X

With this schedule, you use 4*n* tapes, the number of tapes needed for four full backups of the root (/), /usr, /export, and /export/home file systems, plus 24 additional tapes for the incremental backups of the /export/home file systems. This schedule assumes that each incremental backup uses one tape and that you save the tapes for a month.

Here's how this schedule works:

1. On each Sunday, do a full backup (level 0) of the root (/), /usr, /export, and /export/home file systems. Save the level 0 tapes for at least 3 months.
2. On the first Monday of the month, use tape A to do a level 9 backup of the /export/home file system. The ufsdump command copies all files changed since the previous lower-level backup. In this case, the level 0 backup that you did on Sunday.
3. On the first Tuesday of the month, use tape B to do a level 9 backup of the /export/home file system. Again, the ufsdump command copies all files changed since the last lower-level backup, which is Sunday's level 0 backup.
4. On the first Wednesday, use tape C to do a level 5 backup. The ufsdump command copies all files that changed since Sunday.
5. Do the Thursday and Friday level 9 backups on tapes D and E. The ufsdump command copies all files that changed since the last lower-level backup, which is Wednesday's level 5 backup.

6. On the first Saturday of the month, do a level 5 backup of `/export/home`, which copies all files changed since the previous lower-level backup (in this case, the level 0 backup you did on Sunday). Store tapes A-F until the first Monday of the next 4-week period, when you use them again.
7. Repeat steps 1–6 for the next three weeks, using tapes G-L and 4*n* tapes for the level 0 on Sunday, and so on.
8. For each 4-week period, repeat steps 1–7, using a new set of tapes for the level 0s and reusing tapes A–X for the incremental backups. The level 0 tapes could be reused after 3 months.

   This schedule lets you save files in their various states for a month. This plan requires many tapes, but ensures that you have a library of tapes to draw upon. To reduce the number of tapes, you could reuse Tapes A-F each week.

Suggestions for Scheduling Backups

The following table provides other suggestions for scheduling backups.

TABLE 46–13 Suggestions for Backup Schedules

File Restoration Need	Backup Interval	Comments
To restore different versions of files (for example, file systems that are used for word processing)	■ Do daily incremental backups every working day. ■ Do *not* reuse the same tape for daily incremental backups.	This schedule saves all files modified that day, as well as those files still on disk that were modified since the last backup of a lower level. However, with this schedule, you should use a different tape each day because a file that changed on Tuesday, and again on Thursday, goes onto Friday's lower-level backup looking like it did Thursday night—not Tuesday night. If a user needs the Tuesday version, you cannot restore it unless you have a Tuesday backup tape (or a Wednesday backup tape). Similarly, a file that is present on Tuesday and Wednesday, but removed on Thursday, does not appear on the Friday lower-level backup.
To quickly restore a complete file system	Do lower-level backups more frequently.	—
To backup a number of file systems on the same server	Consider offsetting the schedule for different file systems.	This way you're not doing all level 0 backups on the same day.

TABLE 46–13 Suggestions for Backup Schedules *(Continued)*

File Restoration Need	Backup Interval	Comments
To minimize tapes	Increase the level of incremental backups that are done across the week.	Only changes from day to day are saved on each daily tape.
	Increase the level of backups that are done at the end of the week. Put each day's and week's incremental backups onto the same tape.	Only changes from week to week (rather than the entire month) are saved on the weekly tapes.
	Put each day's and week's incremental backups onto the same tape.	To do so, use the no rewind option in the `ufsdump` command.

CHAPTER 47

Backing Up Files and File Systems (Tasks)

This chapter describes the procedures for backing up file systems by using the `ufsdump` command.

For information on these procedures, see "Backing Up Files and File System (Task Map)" on page 641.

For overview information about performing backups, see Chapter 46.

For detailed information on `ufsdump` syntax, options, and arguments, see Chapter 50.

Backing Up Files and File System (Task Map)

Task	Description	For Instructions
1. Prepare for file system backups	Identify the file systems, the type of backup, and the tape device to be used for the backups.	"Preparing for File System Backups" on page 642
2. Determine the number of tapes needed to back up a file system	Determine the number of tapes that are needed for a full backup of a file system.	"How to Determine the Number of Tapes Needed for a Full Backup" on page 643

Task	Description	For Instructions
3. Back up your file systems	Perform a full backup of your file systems to get baseline copies of all files. Perform an incremental backup of your file systems based on whether keeping copies of files that have changed on a daily basis is important at your site.	"How to Backup a File System to Tape" on page 644

Preparing for File System Backups

The preparation for backing up file systems begins with planning, which is described in Chapter 46 and includes choosing the following:

- The file systems to back up
- The type of backup (full or incremental) to perform
- A backup schedule
- A tape drive

This section describes other tasks you might need to perform before you back up file systems, including the following:

- Finding names of file systems to back up
- Determining the number of tapes that are needed for a full backup

▼ How to Find File System Names

1. **Display the contents of the `/etc/vfstab` file.**

   ```
   $ more /etc/vfstab
   ```

2. **Look in the `mount point` column for the name of the file system.**

3. **Use the directory name listed in the `mount point` column when you back up the file system.**

Example—Finding File System Names

The file systems to be backed up in this example are root (/), `/usr`, `/datab`, and `/export/home`.

```
$ more /etc/vfstab
#device           device             mount        FS   fsck mount   mount
```

```
#to mount         to fsck            point        type pass at boot options
#
fd                -                  /dev/fd      fd    -    no      -
/proc             -                  /proc        proc  -    no      -
/dev/dsk/c0t0d0s1 -                  -            swap  -    no      -
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 /            ufs   1    no      -
/dev/dsk/c0t0d0s6 /dev/rdsk/c0t0d0s6 /usr         ufs   1    no      -
/dev/dsk/c0t0d0s5 /dev/rdsk/c0t0d0s5 /datab       ufs   2    yes     -
/dev/dsk/c0t0d0s7 /dev/rdsk/c0t0d0s7 /export/home ufs   2    yes     -
swap              -                  /tmp         tmpfs -    yes     -
```

▼ How to Determine the Number of Tapes Needed for a Full Backup

1. **Become superuser or assume an equivalent role.**
2. **Estimate the size of the backup in bytes.**

   # **ufsdump S** *file-system*

   The `S` displays the estimated number of bytes that are needed to do the backup.
3. **Divide the estimated size by the capacity of the tape to see how many tapes you need.**

   For a list of tape capacities, see Table 46–5.

Example—Determining Number of Tapes

In this example, the file system of 489,472 bytes easily fits on a 150-Mbyte tape.

```
# ufsdump S /export/home
489472
```

Backing Up a File System

The following are general guidelines for performing backups:

- Use single-user mode or unmount the file system, unless you are creating a snapshot of a file system. For information about UFS snapshots, see Chapter 48.
- Be aware that the backing up of file systems when there are directory-level operations (such as creating, removing, and renaming files) and file-level activity occurring means that some data will not be included in the backup.
- You can run the `ufsdump` command from a single system and remotely back up groups of systems across the network through remote shell or remote login, and direct the output to the system on which the tape drive is located. (Typically, the

tape drive is located on the system from which you run the `ufsdump` command, but it does not have to be.)

Another way to back up files to a remote drive is to pipe the output from the `ufsdump` command to the `dd` command. For information about using the `dd` command, see Chapter 51.

- If you are doing remote backups across the network, the system with the tape drive must have entries in its `/.rhosts` file for each client that will be using the drive. Also, the system that initiates the backup must be included in the `/.rhosts` file on each system that it will back up.
- To specify a remote tape device on a system, use the naming convention that matches the OS release of the system with the remote tape drive. For example, use the `/dev/rst0` device for a remote drive on a system that is running the SunOS 4.1.1 release or compatible versions. Use the `/dev/rmt/0` device for a system running the Solaris 9 release or compatible versions.

Note – Use the `nisbackup` command to back up a NIS+ master server. For information on using this command, see *System Administration Guide: Naming and Directory Services (FNS and NIS+)*.

▼ How to Backup a File System to Tape

The following steps provide the general steps for backing up file systems using the `ufsdump` command. The examples show specific uses of options and arguments.

1. **Become superuser or assume an equivalent role.**
2. **Bring the system to run level S (single-user mode).**

   ```
   # shutdown -g30 -y
   ```

3. **(Optional) Check the file system for consistency.**

   ```
   # fsck -m /dev/rdsk/device-name
   ```

   The `fsck -m` command checks for the consistency of file systems. For example, power failures can leave files in an inconsistent state. For more information on the `fsck` command, see Chapter 43.

4. **If you need to back up file systems to a remote tape drive:**

   a. **On the system to which the tape drive is attached (the tape server), add the following entry to its `/.rhosts` file.**

      host `root`

      The *host* entry specifies the name of the system on which you will run the `ufsdump` command to perform the backup.

b. **On the tape server, verify that the host added to the `/.rhosts` file is accessible through the name service.**

5. **Identify the device name of the tape drive.**

   The default tape drive is the `/dev/rmt/0` device.

6. **Insert a tape that is not write-protected into the tape drive.**

7. **Back up file systems.**

   # **ufsdump** *options arguments filenames*

   The following examples show how to use the most common `ufsdump` options and arguments:

   - "Example—Performing a Full Backup of root (/)" on page 645
   - "Example—Performing an Incremental Backup of root (/)" on page 646
   - "Example—Performing a Full Backup, Individual Home Directory" on page 647
   - "Example—Performing a Full Backup to Remote System (Solaris 9 Data to Solaris 9 System)" on page 648

   For other `ufsdump` options and arguments, see Chapter 50.

8. **If prompted, remove the tape and insert the next tape volume.**

9. **Label each tape with the volume number, dump level, date, system name, disk slice, and file system.**

10. **Bring the system back to run level 3 by pressing Control-D.**

11. **Verify that the backup was successful.**

    # **ufsrestore tf** *device-name*

Example—Performing a Full Backup of root (/)

The following example shows how to do a full backup of the root (/) file system. The system in this example is brought to single-user mode before the backup. The following `ufsdump` options are included:

- `0` specifies that this is a `0` level dump (or a full backup)
- `u` specifies that the `/etc/dumpdates` file is updated with the date of this backup
- `c` identifies a cartridge tape device
- `f /dev/rmt/0` identifies the tape device
- `/` is the file system being backed up

```
# shutdown -g30 -y
# ufsdump 0ucf /dev/rmt/0 /
  DUMP: Writing 63 Kilobyte records
  DUMP: Date of this level 0 dump: Wed Sep 05 13:27:20 2001
  DUMP: Date of last level 0 dump: the epoch
```

```
DUMP: Dumping /dev/rdsk/c0t1d0s0 (earth:/) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Estimated 57150 blocks (27.91MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: Tape rewinding
DUMP: 57076 blocks (27.87MB) on 1 volume at 265 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Wed Sep 05 13:27:20 2001
# ufsrestore tf /dev/rmt/0
       2      .
       3      ./lost+found
    3776      ./usr
    7552      ./var
   11328      ./export
   15104      ./export/home
   18880      ./etc
   22656      ./etc/default
   22657      ./etc/default/sys-suspend
   22673      ./etc/default/cron
   22674      ./etc/default/devfsadm
   22675      ./etc/default/dhcpagent
   22676      ./etc/default/fs
   22677      ./etc/default/inetinit
   22678      ./etc/default/kbd
   22679      ./etc/default/mpathd
   22680      ./etc/default/nfslogd
   22681      ./etc/default/passwd
              .
              .
              .
# (Press Control-d to bring system to run level 3)
```

Example—Performing an Incremental Backup of root (/)

The following example shows how to do an incremental backup of the root (/) file system. The following `ufsdump` options are included:

- `9` specifies that this is a `9` level dump (or an incremental backup)
- `u` specifies that the `/etc/dumpdates` file is updated with the date of this backup
- `c` identifies a cartridge tape device
- `f /dev/rmt/0` identifies the tape device
- `/` is the file system being backed up

```
# ufsdump 9ucf /dev/rmt/0 /
  DUMP: Writing 63 Kilobyte records
  DUMP: Date of this level 9 dump: Fri Jul 13 10:58:12 2001
  DUMP: Date of last level 0 dump: Fri Jul 13 10:46:09 2001
  DUMP: Dumping /dev/rdsk/c0t0d0s0 (starbug:/) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
```

```
  DUMP: Mapping (Pass II) [directories]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 200 blocks (100KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 124 blocks (62KB) on 1 volume at 8 KB/sec
  DUMP: DUMP IS DONE
  DUMP: Level 9 dump on Fri Jul 13 10:58:12 2001
# ufsrestore tf /dev/rmt/0
         2      .
         3      ./lost+found
      5696      ./usr
     11392      ./var
     17088      ./export
     22784      ./export/home
     28480      ./opt
      5697      ./etc
     11393      ./etc/default
     11394      ./etc/default/sys-suspend
     11429      ./etc/default/cron
     11430      ./etc/default/devfsadm
     11431      ./etc/default/dhcpagent
     11432      ./etc/default/fs
     11433      ./etc/default/inetinit
     11434      ./etc/default/kbd
     11435      ./etc/default/nfslogd
     11436      ./etc/default/passwd
     11437      ./etc/default/tar
                .
                .
                .
```

Example—Performing a Full Backup, Individual Home Directory

The following example shows how to do a full backup of the `/export/home/kryten` directory. The following `ufsdump` options are included:

- `0` specifies that this is a `0` level dump (or a full backup)
- `u` specifies that the `/etc/dumpdates` file is updated with the date of this backup
- `c` identifies a cartridge tape device
- `f /dev/rmt/0` identifies the tape device
- `/export/home/kryten` is the directory being backed up

```
# ufsdump 0ucf /dev/rmt/0 /export/home/kryten
  DUMP: Writing 63 Kilobyte records
  DUMP: Date of this level 0 dump: Fri Jul 13 11:30:45 2001
```

```
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t3d0s7 (pluto:/export/home) to /dev/rmt/0.
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 232 blocks (116KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 124 blocks (62KB) on 1 volume at 8 KB/sec
  DUMP: DUMP IS DONE
# ufsrestore tf /dev/rmt/0
         2       .
      2688       ./kryten
      5409       ./kryten/letters
      5410       ./kryten/letters/letter1
      5411       ./kryten/letters/letter2
      5412       ./kryten/letters/letter3
      2689       ./kryten/.profile
      8096       ./kryten/memos
        30       ./kryten/reports
        31       ./kryten/reports/reportA
        32       ./kryten/reports/reportB
        33       ./kryten/reports/reportC
#
```

Example—Performing a Full Backup to Remote System (Solaris 9 Data to Solaris 9 System)

The following example shows how to do a full backup of a local `/export/home` file system on a Solaris 9 system (`starbug`) to a tape device on a remote Solaris 9 system (`earth`).The following `ufsdump` options are included:

- `0` specifies that this is a `0` level dump (or a full backup)
- `u` specifies that the `/etc/dumpdates` file is updated with the date of this backup
- `c` identifies a cartridge tape device
- `f earth:/dev/rmt/0` identifies the remote system name and tape device
- `/export/home` is the file system being backed up

```
# ufsdump 0ucf earth:/dev/rmt/0 /export/home
  DUMP: Writing 63 Kilobyte records
  DUMP: Date of this level 0 dump: Wed Sep 05 14:52:31 2001
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/rdsk/c0t0d0s7 (starbug:/export/home) to earth:/dev ...
  DUMP: Mapping (Pass I) [regular files]
  DUMP: Mapping (Pass II) [directories]
  DUMP: Estimated 266 blocks (133KB).
  DUMP: Dumping (Pass III) [directories]
  DUMP: Dumping (Pass IV) [regular files]
  DUMP: Tape rewinding
  DUMP: 250 blocks (125KB) on 1 volume at 247 KB/sec
```

```
  DUMP: DUMP IS DONE
  DUMP: Level 0 dump on Wed Sep 05 14:52:31 2001
# ufsrestore tf earth:/dev/rmt/0
         2      .
         3      ./lost+found
      7168      ./rimmer
      7169      ./rimmer/.profile
     21504      ./rimmer/scdir
     21505      ./rimmer/scdir/scd557
     21506      ./rimmer/scdir/scd772
     10752      ./lister
     10753      ./lister/.profile
     10754      ./lister/filea
     10755      ./lister/fileb
     10756      ./lister/filec
     14336      ./pmorph
     14337      ./pmorph/.profile
      3584      ./pmorph/bigdir
      3585      ./pmorph/bigdir/bigfile
     17920      ./pmorph/smalldir
     17921      ./pmorph/smalldir/smallfile
 #
```

CHAPTER 48

Using UFS Snapshots (Tasks)

This chapter describes how to create and back up UFS snapshots.

For information on the procedures associated with creating UFS snapshots, see "Using UFS Snapshots (Task Map)" on page 651.

For overview information about performing backups, see Chapter 46.

Using UFS Snapshots (Task Map)

Task	Description	For Instructions
1. Create a UFS snapshot	Create a read-only copy of a file system with the `fssnap` command.	"How to Create a UFS Snapshot" on page 654
2. Display UFS snapshot information	Identify UFS snapshot information such as the raw snapshot device.	"How to Display UFS Snapshot Information" on page 654
3. (Optional) Delete a UFS snapshot	Delete a snapshot that is already backed up or no longer needed.	"How to Delete a UFS Snapshot" on page 655
4. Back up a UFS snapshot	Choose one of the following backup methods:	
	Create a full backup of a UFS snapshot with the `ufsdump` command.	"How to Create a Full Backup of a UFS Snapshot (`ufsdump`)" on page 656

Task	Description	For Instructions
	Create an incremental backup of a UFS snapshot with the `ufsdump` command.	"How to Create an Incremental Backup of a UFS Snapshot (`ufsdump`)" on page 657
	Back up a UFS snapshot with the `tar` command.	"How to Back Up a UFS Snapshot (`tar`)" on page 658
5. (Optional) Restore data from a UFS snapshot	Restore the UFS snapshot the same way as you would restore data with the `ufsrestore` command.	"How to Restore a Complete File System" on page 667

UFS Snapshots Overview

The Solaris 9 release includes the `fssnap` command for backing up file systems while the file system is mounted.

You can use the `fssnap` command to create a read-only snapshot of a file system. A *snapshot* is a file system's temporary image that is intended for backup operations.

When the `fssnap` command is run, it creates a virtual device and a backing-store file. You can back up the *virtual device*, which looks and acts like a real device, with any of the existing Solaris backup commands. The *backing-store* file is a bitmapped file that contains copies of pre-snapshot data that has been modified since the snapshot was taken.

Why Use UFS Snapshots?

UFS snapshots enables you to keep the file system mounted and the system in multiuser mode during backups. Previously, you were advised to bring the system to single-user mode to keep the file system inactive when you used the `ufsdump` command to perform backups. You can also use additional Solaris backup commands like `tar` and `cpio` to back up a UFS snapshot for more reliable backups.

The `fssnap` command gives administrators of non-enterprise-level systems the power of enterprise-level tools like Sun StorEdge™ Instant Image without the large storage demands.

UFS snapshots is similar to the Instant Image product. Instant Image allocates space equal to the size of the entire file system that is being captured. However, the backing-store file that is created by UFS snapshots occupies only as much disk space as needed, and you can place a maximum size on the backing-store file.

This table describes specific differences between UFS snapshots and Instant Image.

UFS Snapshots	Instant Image
Size of the backing-store file depends on how much data has changed since the snapshot was taken	Size of the backing-store file equivalent equals the size of the entire file system being copied
Does not persist across system reboots	Persists across system reboots
Works on UFS file systems	Cannot be used with root (/) or `/usr` file systems
Part of the Solaris 8 1/01, 4/01, 7/01, 10/01, and 10/02 releases and the Solaris 9 release	Part of Sun StorEdge products

Although UFS snapshots can make copies of large file systems, Instant Image is better suited for enterprise-level systems. UFS snapshots is better suited for smaller systems.

UFS Snapshots Performance Issues

When the UFS snapshot is first created, users of the file system might notice a slight pause. The length of the pause increases with the size of the file system to be captured. While the snapshot is active, users of the file system might notice a slight performance impact when the file system is written to, but they will see no impact when the file system is read.

Creating and Deleting UFS Snapshots

When you use the `fssnap` command to create a UFS snapshot, observe how much disk space the backing-store file consumes. The backing-store file uses no space, and then it grows quickly, especially on heavily used systems. Make sure the backing-store file has enough space to grow, or limit its size with the `-o maxsize=`*n* `[k,m,g]` option, where *n* [`k,m,g`] is the maximum size of the backing-store file.

Caution – If the backing-store file runs out of space, the snapshot might delete itself, which causes the backup to fail. Check the `/var/adm/messages` file for possible snapshot errors.

For more information, see `fssnap_ufs`(1M).

▼ How to Create a UFS Snapshot

1. **Become superuser or assume an equivalent role.**
2. **Make sure that the file system has enough disk space for the backing-store file.**

   # **df -k**

3. **Make sure that a backing-store file of the same name and location does not already exist.**

   # **ls** */backing-store-file*

4. **Create the UFS snapshot.**

   # **fssnap -F ufs -o bs=***/backing-store-file /file-system*

   Note – The backing-store file must reside on a different file system than the file system that is being snapshot.

   ---

5. **Verify that the snapshot has been created.**

   # **/usr/lib/fs/ufs/fssnap -i** */file-system*

Examples—Creating a UFS Snapshot

The following example shows how to create a snapshot of the `/usr` file system. The backing-store file is `/scratch/usr.back.file`, and the virtual device is `/dev/fssnap/1`.

```
# fssnap -F ufs -o bs=/scratch/usr.back.file /usr
/dev/fssnap/1
```

The following example shows how to limit the backing-store file to 500 Mbytes.

```
# fssnap -F ufs -o maxsize=500m,bs=/scratch/usr.back.file /export/home
/dev/fssnap/1
```

▼ How to Display UFS Snapshot Information

You can display the current snapshots on the system by using the `fssnap -i` option. If you specify a file system, you see detailed information about that snapshot. If you don't specify a file system, you see information about all of the current UFS snapshots and their corresponding virtual devices.

Note – Use the UFS file system-specific `fssnap` command to view the extended snapshot information as shown in the following examples.

1. **Become superuser or assume an equivalent role.**
2. **List current snapshots.**

```
# /usr/lib/fs/ufs/fssnap -i
Snapshot number               : 0
Block Device                  : /dev/fssnap/0
Raw Device                    : /dev/rfssnap/0
Mount point                   : /export/home
Device state                  : idle
Backing store path            : /var/tmp/bs.file
Backing store size            : 0 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Wed Aug 29 15:22:06 2001
Copy-on-write granularity     : 32 KB
```

To display detailed information about a specific snapshot, use the following:

```
# /usr/lib/fs/ufs/fssnap -i /usr
Snapshot number               : 0
Block Device                  : /dev/fssnap/0
Raw Device                    : /dev/rfssnap/0
Mount point                   : /usr
Device state                  : idle
Backing store path            : /var/tmp/bs.file
Backing store size            : 0 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Wed Aug 29 15:23:35 2001
Copy-on-write granularity     : 32 KB
```

Deleting a UFS Snapshot

When you create a UFS snapshot, you can specify that the backing-store file is unlinked, which means that the backing-store file is removed after the snapshot is deleted. If you don't specify the `-o unlink` option when you create a UFS snapshot, you will have to delete it manually.

The backing-store file occupies disk space until the snapshot is deleted, whether you use the `-o unlink` option to remove the backing-store file or you remove it manually.

▼ How to Delete a UFS Snapshot

You can delete a snapshot either by rebooting the system or by using the `fssnap -d` command and specifying the path of the file system that contains the UFS snapshot.

1. **Become superuser or assume an equivalent role.**
2. **Identify the snapshot to be deleted.**

   ```
   # /usr/lib/fs/ufs/fssnap -i
   ```

3. **Delete the snapshot.**

   ```
   # fssnap -d /file-system
   Deleted snapshot 1.
   ```

4. **(Optional) If you did not use the `-o unlink` option when you created the snapshot, you need to delete the backing-store file manually.**

   ```
   # rm /file-system/backing-store-file
   ```

Example—Deleting a UFS Snapshot

The following example shows how to delete a snapshot and assumes that the `unlink` option was not used.

```
# fssnap -i
    0    /    1    /usr
# fssnap -d /usr
 Deleted snapshot 1.
# rm /scratch/usr.back.file
```

Backing Up a UFS Snapshot

You can create a full or incremental back up of UFS snapshot. You can use the standard Solaris backup commands to back up a UFS snapshot.

The virtual device that contains the UFS snapshot acts as a standard read-only device. This means you can back up the virtual device as if you were backing up a file system device.

If you are using the `ufsdump` command to back up a UFS snapshot, you can specify the snapshot name during the backup. See the following section for more information.

▼ How to Create a Full Backup of a UFS Snapshot (`ufsdump`)

1. **Become superuser or assume an equivalent role.**
2. **Identify the UFS snapshot to be backed up.**

```
# /usr/lib/fs/ufs/fssnap -i /file-system
```

For example:

```
# /usr/lib/fs/ufs/fssnap -i /usr
Snapshot number               : 0
Block Device                  : /dev/fssnap/0
Raw Device                    : /dev/rfssnap/0
Mount point                   : /usr
Device state                  : idle
Backing store path            : /var/tmp/back.store
Backing store size            : 576 KB
Maximum backing store size    : Unlimited
Snapshot create time          : Wed Dec 12 09:39:37 2001
Copy-on-write granularity     : 32 KB
```

3. **Back up the UFS snapshot.**

```
# ufsdump 0ucf /dev/rmt/0 /snapshot-name
```

For example:

```
# ufsdump 0ucf /dev/rmt/0 /dev/rfssnap/1
```

4. **Verify that the snapshot is backed up.**

```
# ufsrestore tf /dev/rmt/0
```

▼ How to Create an Incremental Backup of a UFS Snapshot (`ufsdump`)

If you want to back up a UFS snapshot incrementally, which means only the files that have been modified since the last snapshot are backed up, use the `ufsdump` command with the new `N` option. This option specifies the file system device name to be inserted into the `/etc/dumpdates` file for tracking incremental dumps.

The following `ufsdump` command specifies an embedded `fssnap` command to create an incremental backup of a file system.

1. **Become superuser or assume an equivalent role.**

2. **Create an incremental backup of a UFS snapshot.**

For example:

```
# ufsdump 1ufN /dev/rmt/0 /dev/rdsk/c0t1d0s0 `fssnap -F ufs -o raw,bs=
/export/scratch,unlink /dev/rdsk/c0t1d0s0`
```

The `-o raw` option is used in the example to display the name of the raw device instead of the block device. By using this option, you make it easier to embed the `fssnap` command in commands that require the raw device instead, such as the `ufsdump` command.

3. **Verify that the snapshot is backed up.**

```
# ufsrestore ta /dev/rmt/0
```

▼ How to Back Up a UFS Snapshot (`tar`)

If you are using the `tar` command to back up the snapshot, mount the snapshot before backing it up.

1. **Become superuser or assume an equivalent role.**
2. **Create a mount point for the snapshot.**

   For example:

```
# mkdir /backups/home.bkup
```

3. **Mount the snapshot.**

```
# mount -F ufs -o ro /dev/fssnap/1 /backups/home.bkup
```

4. **Change to the mounted snapshot directory.**

```
# cd /backups/home.bkup
```

5. **Back up the snapshot with the `tar` command.**

```
# tar cvf /dev/rmt/0 .
```

Restoring Data From a UFS Snapshot Backup

The backup created from the virtual device is essentially just a backup of what the original file system looked like when the snapshot was taken. When you restore from the backup, restore as if you had taken the backup directly from the original file system, such as a backup that used the `ufsrestore` command. For information on using the `ufsrestore` command to restore a file or file system, see Chapter 49.

CHAPTER 49

Restoring Files and File Systems (Tasks)

This chapter describes how to use the `ufsrestore` command to restore files and file systems that were backed up by using the `ufsdump` command.

For information on the procedures associated with restoring files and file systems, see "Restoring Files and File System Backups (Task Map)" on page 659.

For information about other commands you can use to archive, restore, copy, or move files and file systems, see Chapter 51.

Restoring Files and File System Backups (Task Map)

The following task map describes the procedures associated with restoring files and file systems.

Task	Description	Instructions
Prepare to restore files and file systems	Identify the file system or files to be restored, the tape device, and how you will restore the files.	"Preparing to Restore Files and File Systems" on page 660
Determine which tapes to use	Refer to your backup tapes to find the date of the last backup that contains the file or file system that you need to restore.	"How to Determine Which Tapes to Use" on page 662
Restore files	Choose one of the following restore methods:	

Task	Description	Instructions
	Restore files interactively - Use this method when you are unsure of the file names because you can browse the media contents and select individual files and directories.	"How to Restore Files Interactively" on page 663
	Restore files non-interactively - Using this method is probably faster if you already know the few file names to be restored.	"How to Restore Specific Files Non-Interactively" on page 665
	Restore a file system – Use this method when you get a new disk drive or as part of a recovery procedure.	"How to Restore a Complete File System" on page 667
Restore the root (/) or /usr file systems	Restoring the root (/) or /usr file systems involves booting the system from a local CD or the network.	"How to Restore the root (/) and /usr File Systems" on page 670

Preparing to Restore Files and File Systems

The `ufsrestore` command copies files to disk, relative to the current working directory, from backups that were created by using the `ufsdump` command. You can use the `ufsrestore` command to reload an entire file system hierarchy from a level 0 dump and incremental dumps that follow it or to restore one or more single files from any backup tape. If the `ufsrestore` command is run as superuser, files are restored with their original owner, last modification time, and mode (permissions).

Before you start to restore files or file systems, you need to know the following:

- The tapes (or diskettes) you need
- The raw device name on which you want to restore the file system
- The type of tape device you will use
- The device name (local or remote) for the tape drive

Determining the File System Name

If you have properly labeled your backup tapes, you should be able to use the file system name (`/dev/rdsk/`*device-name*) from the tape label. For more information, see "How to Find File System Names" on page 642.

Determining the Type of Tape Device You Need

You must use a tape device that is compatible with the backup media to restore the files. The format of the backup media determines which drive you must use to restore files. For example, if your backup media is 8-mm tape, you must use an 8-mm tape drive to restore the files.

Determining the Tape Device Name

You might have specified the tape device name (`/dev/rmt/`*n*) as part of the backup tape label information. If you are using the same drive to restore a backup tape, you can use the device name from the label. For more information on media devices and device names, see Chapter 52.

Restoring Files and File Systems

When you back up files and directories, you save them relative to the file system in which they belong. When you restore files and directories, the `ufsrestore` command re-creates the file hierarchy in the current working directory.

For example, files backed up from the `/export/doc/books` directory (where `/export` is the file system), are saved relative to `/export`. In other words, the `book1` file in the `books` directory is saved as `./doc/books/book1` on the tape. Later on, if you restored the `./doc/books/book1` file to the `/var/tmp` directory, the file would be restored to `/var/tmp/doc/books/book1`.

When you restore individual files and directories, it is a good idea to restore them to a temporary location, such as the `/var/tmp` directory. After you verify them, you can move the files to their proper locations. You can restore individual files and directories to their original locations. If you do so, be sure you are not overwriting newer files with older versions from the backup tape.

To avoid conflicts with other users, you might want to create and change to a subdirectory, such as the`/var/tmp/restore` file, in which to restore the files.

If you are restoring a hierarchy, you should restore the files in a temporary directory on the same file system where the files will reside. So, you can use the `mv` command to move the entire hierarchy where it belongs after it is restored.

Note – Do not restore files in the `/tmp` directory even temporarily. The `/tmp` directory is usually mounted as a TMPFS file system and TMPFS does not support UFS file system attributes such as ACLs.

▼ How to Determine Which Tapes to Use

1. **Ask the user for the approximate date the files to be restored were last modified.**
2. **Refer to your backup plan to find the date of the last backup that contains the file or file system.**

   To retrieve the most recent version of a file, work backward through the incremental backups from highest to lowest level and most recent to least recent, unless the user requests otherwise.
3. **(Optional) If you have online archive files, identify the correct media.**

   # **ufsrestore ta** *archive-name ./path/filename ./path/filename*

`t`	Lists each file on the tape.
`a`	Reads the table of contents from the online archive file instead of the tape.
archive-name	Identifies the online archive file name.
./path/filename	Identifies the file name(s) you are looking for on the online archive. If successful, the `ufsrestore` command prints out the inode number and file name. If unsuccessful, `ufsrestore` prints an error message.

   For more information, see `ufsrestore`(1M).
4. **Insert the media that contains the files to be restored in the drive and verify the correct media.**

   # **ufsrestore tf /dev/rmt/***n ./path/filename ./path/filename*

   Be sure to use the complete path for the *filename(s)*. If a file is in the backup, its name and inode number is listed. Otherwise, a message says that the file is not on the volume.
5. **(Optional) If you have multiple dump files on the same tape, position the tape at the dump file you want to use.**

   # **ufsrestore tfs /dev/rmt/***n tape-number*

Examples—Determining Which Tapes to Use

The following example shows how to checks if `/etc/passwd` file is in the online archive.

```
# ufsrestore ta /var/tmp/root.archive ./etc/passwd
```

The following example show to verify that the `/etc/passwd` is on the backup tape.

```
# ufsrestore tf /dev/rmt/0 ./etc/passwd
```

▼ How to Restore Files Interactively

1. **Become superuser or assume an equivalent role.**
2. **(Optional) Write-protect the tapes for safety.**
3. **Insert the volume 1 tape into the tape drive.**
4. **Change to a directory that will be used to restore the files temporarily.**

   ```
   # cd /var/tmp
   ```

5. **Start the interactive restoration.**

   ```
   # ufsrestore if /dev/rmt/n
   ```

   Some informational messages and the `ufsrestore>` prompt are displayed.

6. **Create a list of files to be restored.**

   a. **List the contents of a directory.**

      ```
      ufsrestore> ls directory
      ```

   b. **Change to a directory.**

      ```
      ufsrestore> cd directory-name
      ```

   c. **Create a list of files and directories that you want to restore.**

      ```
      ufsrestore> add filename filename
      ```

   d. **(Optional) (Optional) Remove any directory or file name from the list of files to be restored, if necessary.**

      ```
      ufsrestore> delete filename
      ```

7. **(Optional) (Optional) Display the file names as they are being restored.**

   ```
   ufsrestore> verbose
   ```

8. **Restore the files.**

   ```
   ufsrestore> extract
   ```

The `ufsrestore` command asks you which volume number to use.

9. **Type the volume number and press Return. If you have only one volume, type `1` and press Return.**

   ```
   Specify next volume #: 1
   ```

   The files and directories in the list are extracted and restored to the current working directory.

10. **To keep the mode of the current directory unchanged, enter `n` at the `set owner/mode` prompt.**

    ```
    set owner/mode for '.'? [yn] n
    ```

    You must wait while the `ufsrestore` command performs its final cleanup.

11. **Quit the `ufsrestore` program.**

    ```
    ufsrestore> quit
    ```

    You then see the shell prompt.

12. **Verify the restored files.**

    a. **List the restored files and directories.**

       ```
       # ls -l
       ```

       A list of files and directories is displayed.

    b. **Check the list to be sure all the files and directories you specified in the list have been restored.**

13. **Move the files to the proper directories.**

Example—Restoring Files Interactively

The following example shows how to extract the `/etc/passwd` and `/etc/shadow` files from the backup tape.

```
# cd /var/tmp
# ufsrestore if /dev/rmt/0
ufsrestore> ls
.:
 .cpr_config   etc/          lost+found/   sbin/         usr/
 TT_DB/        export/       mnt/          sccs/         var/
 b/            home/         net/          share/        vol/
 bin           kernel/       opt/          shared/       ws/
 dev/          lib           platform/     src/          xfn/
 devices/      license/      proc/         tmp/
ufsrestore> cd etc
ufsrestore> add passwd shadow
ufsrestore> verbose
verbose mode on
```

```
ufsrestore> extract
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./etc/shadow
extract file ./etc/passwd
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
ufsrestore> quit
#
```

▼ How to Restore Specific Files Non-Interactively

1. **Become superuser or assume an equivalent role.**
2. **(Optional) Write-protect the tape for safety.**
3. **Insert the volume 1 tape into the tape drive.**
4. **Change to a directory that will be used to restore files temporarily.**

   ```
   # cd /var/tmp
   ```

5. **Restore the file or files.**

   ```
   # ufsrestore xvf /dev/rmt/n filename
   ```

`x`	Tells `ufsrestore` to copy specific files or directories in the *filename* argument.
`v`	Displays the file names as they are restored.
`f /dev/rmt/`*n*	Identifies the tape device name.
filename	Specifies one or more individual file names or directory names separated by spaces, for example: `./export/home/user1/mail` `./export/home/user2/mail`.

6. **Type the volume number where files are located and press Return.**

   ```
   Specify next volume #: 1
   ```

   The file or files are restored to the current working directory.

7. **To keep the mode of the current directory unchanged, type `n` and press Return at the `set owner/mode` prompt.**

   ```
   set owner/mode for '.'? [yn] n
   ```

8. **Verify the restored files.**

   a. **List the restored files and directories.**

      ```
      # ls -l
      ```

      A list of files and directories is displayed.

   b. **Check the list to be sure all the files and directories you specified in the list have been restored.**

9. **Move the files to the proper directories.**

Example—Restoring Specific Files Non-Interactively

The following example shows how to restore the `passwd` and `shadow` files to the `/var/tmp` directory.

```
# cd /var/tmp
# ufsrestore xvf /dev/rmt/0 ./etc/passwd ./etc/shadow
Verify volume and initialize maps
Media block size is 126
Dump   date: Wed Dec 12 10:54:45 2001
Dumped from: the epoch
Level 0 dump of / on earth:/dev/dsk/c0t1d0s0
Label: none
Extract directories from tape
Initialize symbol table.
Make node ./etc
Extract requested files
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
extract file ./etc/passwd
extract file ./etc/shadow
Add links
Set directory mode, owner, and times.
set owner/mode for '.'? [yn] n
# cd etc
# mv passwd /etc
# mv shadow /etc
# ls -l /etc
```

Example—Restoring Files From a Remote Tape Device

You can restore files from a remote tape drive by adding *remote-host* : to the front of the tape device name, when using the `ufsrestore` command.

The following example shows how to restore files using a remote tape drive `/dev/rmt/0` on the system `venus`.

```
# ufsrestore xf venus:/dev/rmt/0 ./etc/hosts
```

▼ How to Restore a Complete File System

Occasionally, a file system becomes so damaged that you must completely restore it. Typically, you need to restore a complete file system after a disk failure. You might need to replace the hardware before you can restore the software. For information on how to replace a disk, see Chapter 34 or Chapter 35. Full restoration of a file system such as `/export/home` can take a lot of time. If you have consistently backed up file systems, you can restore them to their state from the time of the last incremental backup.

Note – You cannot use this procedure to restore the root (`/`) or `/usr` file systems. For instructions on restoring these file systems, see "How to Restore the root (/) and /usr File Systems" on page 670.

---

1. **Become superuser or assume an equivalent role.**
2. **If necessary, unmount the file system.**

   ```
   # umount /dev/rdsk/device-name
   ```

3. **Create the new file system.**

   ```
   # newfs /dev/rdsk/device-name
   ```

   You are asked if you want to construct a new file system on the raw device. Verify that the *device-name* is correct so you don't destroy the wrong file system.

   For more information, see `newfs`(1M).

4. **Confirm that the new file system should be created.**

   ```
   newfs: construct a new file system /dev/rdsk/cwtxdysz:(y/n)? y
   ```

   The new file system is created.

5. **Mount the new file system on a temporary mount point.**

   ```
   # mount /dev/dsk/device-name /mnt
   ```

6. **Change to the mount point directory.**

   ```
   # cd /mnt
   ```

7. **(Optional) Write-protect the tapes for safety.**
8. **Insert the first volume of the level 0 tape into the tape drive.**
9. **Restore the files.**

```
# ufsrestore rvf /dev/rmt/n
```

The level 0 dump is restored. If the dump required multiple tapes, you are prompted to load each tape in numeric order.

10. **Remove the tape and load the next level tape in the drive.**

    Always restore tapes starting with 0 and continuing until you reach the highest level.

11. **Repeat Step 8 through Step 10 for each dump level, from the lowest to the highest level.**

12. **Verify that the file system is restored.**

    ```
    # ls
    ```

13. **Remove the `restoresymtable` file.**

    ```
    # rm restoresymtable
    ```

    The `restoresymtable` file that is created and used by the `ufsrestore` command to check-point the restore is removed.

14. **Change to another directory.**

    ```
    # cd /
    ```

15. **Unmount the newly restored file system.**

    ```
    # umount /mnt
    ```

16. **Remove the last tape and insert a new tape that is not write-protected in the tape drive.**

17. **Make a level 0 backup of the newly restored file system.**

    ```
    # ufsdump 0uf /dev/rmt/n /dev/rdsk/device-name
    ```

    A level 0 backup is performed. Always do an immediate backup of a newly created file system because `ufsrestore` repositions the files and changes the inode allocation.

18. **Mount the restored file system.**

    ```
    # mount /dev/dsk/device-name mount-point
    ```

    The restored file system is mounted and available for use.

19. **Verify that the restored and mounted file system is available.**

    ```
    # ls mount-point
    ```

Example—Restoring a Complete File System

The following example shows how to restore the `/export/home` file system.

```
# umount /export/home
# newfs /dev/rdsk/c0t3d0s7
newfs: construct a new file system /dev/rdsk/c0t3d0s7: (y/n)? y
/dev/rdsk/c0t3d0s7:    410400 sectors in 270 cylinders of 19 tracks,
80 sectors
200.4MB in 17 cyl groups (16 c/g, 11.88MB/g, 5696 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 24432, 48832, 73232, 97632, 122032, 146432, 170832, 195232, 219632,
 244032, 268432, 292832, 317232, 341632, 366032, 390432,
# mount /dev/dsk/c0t3d0s7 /mnt
# cd /mnt
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump   date: Sat Jul 14 08:49:33 2001
Dumped from: the epoch
Level 0 dump of /export/home on earth:/dev/dsk/c0t3d0s7
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Warning: ./lost+found: File exists
Make node ./kryten
Make node ./kryten/letters
Make node ./kryten/reports
Extract new leaves.
Check pointing the restore
extract file ./kryten/.cshrc
extract file ./kryten/.login
extract file ./kryten/b
extract file ./kryten/memos
extract file ./kryten/letters/b
extract file ./kryten/letters/letter1
extract file ./kryten/letters/letter2
extract file ./kryten/letters/letter3
extract file ./kryten/reports/reportA
extract file ./kryten/reports/reportB
extract file ./kryten/reports/reportC
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
# ls
# rm restoresymtable
# cd /
# umount /mnt
# ufsdump 0ucf /dev/rmt/0 /export/home
                    .
                    .
                    .
# mount /dev/dsk/c0t3d0s7 /export/home
# ls /export/home
```

▼ How to Restore the root (/) and /usr File Systems

1. **Become superuser or assume an equivalent role.**
2. **Add a new system disk to the system where the root (/) and /usr file systems will be restored.**

   For a detailed description about adding a system disk, refer to Chapter 34 or Chapter 35.
3. **Mount the new file system on a temporary mount point.**

   ```
   # mount /dev/dsk/device-name /mnt
   ```
4. **Change to the /mnt directory.**

   ```
   # cd /mnt
   ```
5. **(Optional) Write-protect the tapes for safety.**
6. **Restore the root file system.**

   ```
   # ufsrestore rvf /dev/rmt/n
   ```

   The level 0 tape is restored.
7. **Remove the tape and load the next level tape in the drive.**

   Always restore tapes starting with dump level 0 and continuing from lowest to highest level.
8. **Continue restoring as needed.**

   ```
   # ufsrestore rvf /dev/rmt/n
   ```

   The next level tape is restored.
9. **Repeat Step 7 and Step 8 for each additional tape.**
10. **Verify the file system is restored.**

    ```
    # ls
    ```
11. **Remove the restoresymtable file.**

    ```
    # rm restoresymtable
    ```

    The restoresymtable file that is created and used by the ufsrestore command to check-point the restore is removed.
12. **Change to the root (/) directory.**

    ```
    # cd /
    ```
13. **Unmount the newly created file system.**

    ```
    # umount /mnt
    ```

14. **Check the new file system.**

    ```
    # fsck /dev/rdsk/device-name
    ```

    The restored file system is checked for consistency.

15. **Create the boot blocks on the root partition.**

    ```
    # installboot  /usr/platform/`uname-i`/lib/fs/ufs/bootblk
    /dev/rdsk/device-name
    ```

    For more information, see `installboot`(1M).

    For an example of using the `installboot` command on a SPARC based system, see "SPARC: Example—Restoring the root (/) File System" on page 671. For an example of using the `installboot` command on an x86 based system, see "x86: Example—Restoring the root (/) File System" on page 672.

16. **Insert a new tape in the tape drive.**

17. **Back up the new file system.**

    ```
    # ufsdump 0uf /dev/rmt/n /dev/rdsk/device-name
    ```

    A level 0 backup is performed. Always do an immediate backup of a newly created file system because `ufsrestore` repositions the files and changes the inode allocation.

18. **Repeat steps 5 through 16 for the `/usr` file system, if necessary.**

19. **Reboot the system.**

    ```
    # init 6
    ```

    The system is rebooted.

SPARC: Example—Restoring the root (/) File System

This example shows how to restore the root (/) file system on a SPARC system. This example assumes that the system is booted from a local CD or from the network.

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdsk/c0t3d0s0
# installboot /usr/platform/sun4m/lib/fs/ufs/bootblk /dev/rdsk/c0t3d0s0
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t3d0s0
# init 6
```

x86: Example—Restoring the root (/) File System

This example shows how to restore the root (/) file system on an x86 system. This example assumes that the system is booted from a local CD or from the network.

```
# mount /dev/dsk/c0t3d0s0 /mnt
# cd /mnt
# tapes
# ufsrestore rvf /dev/rmt/0
# ls
# rm restoresymtable
# cd /
# umount /mnt
# fsck /dev/rdsk/c0t3d0s0
# installboot /usr/platform/`uname -i`/lib/fs/ufs/pboot /usr/platform/`uname -i`/lib/fs/
ufs/bootblk /dev/rdsk/c0t3d0s2
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t3d0s0
# init 6
```

CHAPTER 50

UFS Backup and Restore Commands (Reference)

This chapter contains reference information on the `ufsdump` and `ufsrestore` commands.

This is a list of information in this chapter.

- "How the `ufsdump` Command Works" on page 673
- "Options and Arguments for the `ufsdump` Command" on page 678
- "The `ufsdump` Command and Security Issues" on page 680
- "Options and Arguments for the `ufsrestore` Command" on page 681

How the `ufsdump` Command Works

The `ufsdump` command makes two passes when it backs up a file system. On the first pass, this command scans the raw device file for the file system and builds a table of directories and files in memory. Then, this command writes the table to the backup media. In the second pass, the `ufsdump` command goes through the inodes in numerical order, reading the file contents and writing the data to the media.

Determining Device Characteristics

The `ufsdump` command needs to know only an appropriate block size and how to detect the end of media.

Detecting the End of Media

The `ufsdump` command writes a sequence of fixed-size records. When the `ufsdump` command receives notification that a record was only partially written, it assumes that it has reached the physical end of the media. This method works for most devices. If a device is not able to notify the `ufsdump` command that only a partial record has been written, a media error occurs as the `ufsdump` command tries to write another record.

Note – DAT devices and 8-mm tape devices detect end-of-media. Cartridge tape devices and 1/2–inch tape devices do not detect end-of-media.

The `ufsdump` command automatically detects the end-of-media for most devices. Therefore, you do not usually need to use the `-c`, `-d`, `-s`, and `-t` options to perform multivolume backups.

The only time you need to use the end-of-media options is when the `ufsdump` command does not understand the way the device detects the end-of-media or you are going to restore the files on a SunOS 4. 1 system with an the `restore` command. To ensure compatibility with the `restore` command, the size option can still force the `ufsdump` command to go to the next tape or diskette before reaching the end of the current tape or diskette.

Copying Data With `ufsdump`

The `ufsdump` command copies data only from the raw disk slice. If the file system is still active, anything in memory buffers is probably not copied. The backup done by `ufsdump` does not copy free blocks, nor does it make an image of the disk slice. If symbolic links point to files on other slices, the link itself is copied.

Role of the `/etc/dumpdates` File

The `ufsdump` command, when used with the `-u` option, maintains and updates the `/etc/dumpdates` file. Each line in the `/etc/dumpdates` file shows the file system backed up, the level of the last backup, and the day, date, and time of the backup. For example:

```
/dev/rdsk/c0t0d0s7              0 Mon Dec 10 16:26:10 2001
/dev/rdsk/c0t0d0s7              9 Tue Dec 11 16:45:14 2001
/dev/rdsk/c0t0d0s7              9 Wed Dec 12 16:54:47 2001
```

When you do an incremental backup, the `ufsdump` command checks the `/etc/dumpdates` file to find the date of the most recent backup of the next lower level. Then, this command copies to the media all files that were modified since the date of that lower-level backup. After the backup is complete, a new information line, which describes the backup you just completed, replaces the information line for the previous backup at that level.

Use the `/etc/dumpdates` file to verify that backups are being done. This verification is particularly important if you are having equipment problems. If a backup cannot be completed because of equipment failure, the backup is not recorded in the `/etc/dumpdates` file.

If you need to restore an entire disk, check the `/etc/dumpdates` file for a list of the most recent dates and levels of backups so that you can determine which tapes you need in order to restore the entire file system.

Note – The `/etc/dumpdates` file is a text file that can be edited, but edit it only at your own risk. If you make changes to the file that do not match your archive tapes, you might not be able to find the tapes (or files) you need.

Backup Device (*dump-file*) Argument

The *dump-file* argument (to the `-f` option) specifies the destination of the backup, which can be one of the following:

- Local tape drive or diskette drive
- Remote tape drive or diskette drive
- Standard output

Use this argument when the destination is not the default local tape drive `/dev/rmt/0`. If you use the `-f` option, then you must specify a value for *dump-file*.

Note – The *dump-file* argument can also point to a file on a local or remote disk, which, if used by mistake, can fill up a file system.

Local Tape or Diskette Drive

Typically, the *dump-file* argument specifies a raw device file for a tape device or diskette. When the `ufsdump` command writes to an output device, it creates a single backup file that might span multiple tapes or diskettes.

You specify a tape device or diskette on your system by using a device abbreviation. The first device is always 0. For example, if you have a SCSI tape controller and one QIC-24 tape drive that uses medium-density formatting, use this device name:

```
/dev/rmt/0m
```

When you specify a tape device name, you can also type the letter "n" at the end of the name to indicate that the tape drive should not rewind after the backup is completed. For example:

```
/dev/rmt/0mn
```

Use the "no-rewind" option if you want to put more than one file onto the tape. If you run out of space during a backup, the tape does not rewind before the `ufsdump` command asks for a new tape. For a complete description of device naming conventions, see "Backup Device Names" on page 710.

Remote Tape or Diskette Drive

You specify a remote tape device or diskette by using the syntax *host:device*. The `ufsdump` command writes to the remote device when root on the local system has access to the remote system. If you usually run the `ufsdump` command as root, the name of the local system must be included in the `/.rhosts` file on the remote system. If you specify the device as *user@host:device*, the `ufsdump` command tries to access the device on the remote system as the specified user. In this case, the specified user must be included in the `/.rhosts` file on the remote system.

Use the naming convention for the device that matches the operating system for the system on which the device resides, not the system from which you run the `ufsdump` command. If the drive is on a system that is running a previous SunOS release (for example, 4.1.1), use the SunOS 4.1 device name (for example, `/dev/rst0`). If the system is running Solaris software, use the SunOS 5.9 convention (for example, `/dev/rmt/0`).

Using Standard Output With the `ufsdump` Command

When you specify a dash (`-`) as the *dump-file* argument, the `ufsdump` command writes to standard output.

Note – The `-v` option (verify) does not work when the *dump-file* argument is standard output.

You can use the `ufsdump` and `ufsrestore` commands in a pipeline to copy a file system by writing to the standard output with the `ufsdump` command and reading from standard input with the `ufsrestore` command, as shown in this example:

```
# ufsdump 0f - /dev/rdsk/c0t0d0s7 | (cd /home; ufsrestore xf -)
```

Specifying Files to Back Up

You must always include *filenames* as the last argument on the command line. This argument specifies the source or contents of the backup.

For a file system, specify the raw device file as follows:

```
/dev/rdsk/c0t0d0s6
```

You can specify the file system by its mount point directory (for example, `/export/home`), as long as there is an entry for it in the `/etc/vfstab` file.

For a complete description of device naming conventions, see "Backup Device Names" on page 710.

For individual files or directories, type one or more names separated by spaces.

Note – When you use the `ufsdump` command to back up one or more directories or files (rather than a complete file system), a level 0 backup is done. Incremental backups do not apply.

Specifying Tape Characteristics

If you do not specify any tape characteristics, the `ufsdump` command uses a set of defaults. You can specify tape cartridge (`c`), density (`d`), size (`s`), and number of tracks (`t`). Note that you can specify the options in any order as long as the arguments that follow match the order of the options.

Limitations of the `ufsdump` Command

The `ufsdump` command cannot do the following:

- Automatically calculate the number of tapes or diskettes that are needed for backing up file systems. You can use the dry run mode (S option) to determine the amount of space that is needed before actually backing up file systems.
- Provide built-in error checking to minimize problems when it backs up an active file system.
- Back up files that are remotely mounted from a server. Files on the server must be backed up on the server itself. Users are denied permission to run the `ufsdump` command on files they own that are located on a server.

Options and Arguments for the `ufsdump` Command

This section describes in detail the options and arguments for the `ufsdump` command. The syntax for the `ufsdump` command is as follows:

`/usr/sbin/ufsdump` *options arguments filenames*

options	Is a single string of one-letter option names.
arguments	Identifies option arguments and might be multiple strings. The option letters and the arguments that go with them must be in the same order.
filenames	Identifies the files to back up. These arguments must always come last.

Default `ufsdump` Options

If you run the `ufsdump` command without any options, use this syntax:

`# ufsdump` *filenames*

The `ufsdump` command uses these options and arguments, by default:

`ufsdump 9uf /dev/rmt/0` *filenames*

These options do a level 9 incremental backup to the default tape drive at its preferred density.

Options for the `ufsdump` Command

The following table describes the options for the `ufsdump` command.

TABLE 50–1 Options for the `ufsdump` Command

Option	Description
`0–9`	Dump level. Level 0 is for a full backup of the complete file system that is specified by *filenames*. Levels 1–9 are for incremental backups of files that have changed since the last lower-level backup.

TABLE 50–1 Options for the `ufsdump` Command *(Continued)*

Option	Description
`a` *archive-file*	Archive file. Stores (archives) a backup table of contents in a specified file on the disk. The file can be understood only by the `ufsrestore` command, which uses it to determine whether a file to be restored is present in a backup file, and if so, on which volume of the media it resides.
`b` *factor*	Blocking factor. Specifies the number of 512-byte blocks to write to tape at a time.
`c`	Cartridge. Back up to cartridge tape. When end-of-media detection applies, this option sets the block size to 126.
`d` *bpi*	Tape density. Use this option only when the `ufsdump` command cannot detect the end of the media.
`D`	Diskette. Backs up to diskette.
`f` *dump-file*	Dump file. Writes the files to the destination that is specified by *dump-file* instead of the default device. If the file is specified as *user@system:device*, the `ufsdump` command attempts to execute as the specified user on the remote system. The specified user must have a `/.rhosts` file on the remote system that allows the user who is invoking the command on the local system to access the remote system.
`l`	Autoload. Use this option if you have an autoloading (stackloader) tape drive. When the end of a tape is reached, this option takes the drive offline and waits up to two minutes for the tape drive to be ready again. If the drive is ready within two minutes, it continues. If the drive is not ready after two minutes, it prompts the operator to load another tape.
`n`	Notify. When intervention is needed, this option sends a message to all terminals of all users in the `sys` group.
`o`	Offline. When finished with a tape or diskette, this option takes the drive offline, rewinds (if tape), and if possible removes the media (for example, ejects a diskette or removes an 8-mm autoloaded tape).
`s` *size*	Size. Specifies the length of tapes in feet or for diskettes, the number of 1024-byte blocks. Use this option only when the `ufsdump` command cannot detect the end of the media.
`S`	Estimates the size of the backup. Determines the amount of space that is needed to perform the backup, without actually doing it, and outputs a single number that indicates the estimated size of the backup in bytes.

TABLE 50–1 Options for the `ufsdump` Command *(Continued)*

Option	Description
`t` *tracks*	Tracks. Specifies the number of tracks for a 1/4-inch cartridge tape. Use this option only when the `ufsdump` command cannot detect the end of the media.
`u`	Updates the dump record. A completed backup of a file system adds an entry to the `/etc/dumpdates` file. The entry indicates the device name for the file system's disk slice, the dump level (0–9), and the date. No record is written when you do not use the `u` option or when you back up individual files or directories. If a record already exists for a backup at the same level, it is replaced.
`v`	Verify. After each tape or diskette is written, verifies the contents of the media against the source file system. If any discrepancies occur, prompts the operator to mount new media, then repeats the process. Use this option only on an unmounted file system, because any activity in the file system causes the `ufsdump` command to report discrepancies.
`w`	Warning. Lists the file systems that appear in the `/etc/dumpdates` file that have not been backed up within a day. When you use this option, all other options are ignored.
`W`	Warning with highlight. Shows all the file systems that appear in the `/etc/dumpdates` file and highlights those file systems that have not been backed up within a day. When you use this option, all other options are ignored.

Note – The `/etc/vfstab` file does not contain information about how often to back up a file system.

The `ufsdump` Command and Security Issues

If you are concerned about security, you should do the following:

- Require root access for the `ufsdump` command.
- Ensure root access entries are removed from `/.rhosts` files on clients and servers if doing centralized backups.

  For general information on security, see *System Administration Guide: Security Services*.

Options and Arguments for the `ufsrestore` Command

The syntax of the `ufsrestore` command is:

`ufsrestore` *options arguments filenames*

options	Is a single string of one-letter option names. You must choose one and only one of these options: `i`, `r`, `R`, `t`, or `x`. The additional options listed in Table 50–3 are optional.
arguments	Follows the option string with the arguments that match the options. The option letters and the arguments that go with them must be in the same order.
filenames	Specifies the files to be restored as arguments to the `x` or `t` options. These arguments must always come last.

You must use one (and only one) of the `ufsrestore` command options shown in the following table.

TABLE 50–2 One Required Option for the `ufsrestore` Command

Option	Description
`i`	Interactive. Runs the `ufsrestore` command in an interactive mode. In this mode, you can use a limited set of shell-like commands to browse the contents of the media and select individual files or directories to restore. For a list of interactive commands, see Table 50–4.
`r`	Recursive. Restores the entire contents of the media into the current working directory (which should be the top level of the file system). Information used to restore incremental dumps on top of the full dump (for example, `restoresymtable`) is also included. To completely restore a file system, use this option to restore the full (level 0) dump and each subsequent incremental dump. Although this option is intended for a new file system (that was just created with the `newfs` command), files not on the backup media are preserved.
`R`	Resume restoring. Prompts for the volume from which to resume restoring and restarts from a checkpoint. You rerun the `ufsrestore` command with this option after a full restore (`r` option) is interrupted.

TABLE 50–2 One Required Option for the `ufsrestore` Command *(Continued)*

Option	Description
`x` [*filenames*]	Extract. Selectively restores the files you specify by the *filenames* argument. *filenames* can be a list of files and directories. All files under a specified directory are restored unless you also use the `h` option. If you omit *filenames* or enter " `.` " for the root directory, all files on all volumes of the media (or from standard input) are restored. Existing files are overwritten, and warnings are displayed.
`t` [*filenames*]	Table of contents. Checks the files that are specified in the *filenames* argument against the media. For each file, lists the full file name and the inode number (if the file is found) or indicates that the file is not on the "volume" (meaning any volume in a multivolume dump). If you do not enter the *filenames* argument, all files on all volumes of the media are listed (without distinguishing on which volume files are located). If you also use the `h` option, only the directory files that are specified in *filenames*, not their contents, are checked and listed. The table of contents is read from the first volume of the media, or, if you use the `a` option, from the specified archive file. This option is mutually exclusive with the `x` and `r` options.

Additional `ufsrestore` options are described in the following table.

TABLE 50–3 Additional Options for the `ufsrestore` Command

Option	Description
`a` *archive-file* [*filenames*]	Takes the dump table of contents from the specified *archive-file* instead of from the media (first volume). You can use this option in combination with the `t`, `i`, or `x` options to see if files are on the media without having to mount any media. If you use it with the `x` and interactive extract options, you are prompted to mount the appropriate volume before extracting the file(s).
`b` *factor*	Blocking factor. Specifies number of 512-byte blocks read from tape at a time. By default, the `ufsrestore` command tries to figure out the block size that was used in writing the tape.
`d`	Debug. Turns on debugging messages.
`f` *backup-file*	Backup file. Reads the files from the source indicated by *backup-file*, instead of from the default device file `/dev/rmt/0m`. If you use the `f` option, you must specify a value for *backup-file*. When *backup-file* is of the form *system:device*, the `ufsrestore` command reads from the remote device. You can also use the *backup-file* argument to specify a file on a local or remote disk. If *backup-file* is '-', the files are read from standard input.

TABLE 50–3 Additional Options for the `ufsrestore` Command *(Continued)*

Option	Description
`h`	Turns off directory expansion. Only the directory file you specify is extracted or listed.
`m`	Restores specified files into the current directory on the disk regardless of where they are located in the backup hierarchy and renames them with their inode number. For example, if the current working directory is `/files`, a file in the backup named `./dready/fcs/test` with inode number 42, is restored as `/files/42`. This option is useful only when you are extracting a few files.
`s` *n*	Skips to the *n*th backup file on the media (first volume). This option is useful when you put more than one backup on a single tape.
`v`	Verbose. Displays the names and inode numbers of each file as it is restored.
`y`	Continues when errors occur while reading the media and tries to skip over bad blocks instead of stopping and asking whether to continue. This option tells the command to assume a yes response.

The following table describes `ufsrestore`'s interactive commands.

TABLE 50–4 Commands for Interactive Restore

Option	Description
`ls` [*directory-name*]	Lists the contents of either the current directory or the specified directory. Directories are marked by a `/` suffix. Entries in the current list to be restored (extracted) are marked by an `*` prefix. Inode numbers are shown if the verbose option is used.
`cd` *directory-name*	Changes to the specified directory in the backup hierarchy.
`add` [*filename*]	Adds the current directory or the specified file or directory to the list of files to extract (restore). If you do not use the `h` option, all files in a specified directory and its subdirectories are added to the list. All the files you want to restore to a directory might not be on a single backup tape or diskette. You might need to restore from multiple backups at different levels to get the latest versions of all the files.

TABLE 50–4 Commands for Interactive Restore *(Continued)*

Option	Description
`delete` [*filename*]	Deletes the current directory or the specified file or directory from the list of files to extract (restore). If you do not use the `h` option, all files in the specified directory and its subdirectories are deleted from the list. The files and directories are deleted only from the extract list you are building. They are not deleted from the media or the file system.
`extract`	Extracts the files in the list and restores them relative to the current working directory on the disk. Specify `1` when you are asked for a volume number for a single-volume backup. If you are doing a multitape or multidiskette restore and restoring a small number of files, start with the last tape or diskette instead.
`help`	Displays a list of commands you can use in interactive mode.
`pwd`	Displays the path name of the current working directory in the backup hierarchy.
`q`	Quits interactive mode without restoring any additional files.
`setmodes`	Lets you set the mode for files to be restored to match the mode of the root directory of the file system from which they were backed up. You are prompted with: `set owner/mode for '.' [yn]?` Type `y` (for yes) to set the mode (permissions, owner, times) of the current directory to match the root directory of the file system from which they were backed up. Use this mode when you restore a complete file system. Type `n` (for no) to leave the mode of the current directory unchanged. Use this mode when you restore part of a backup to a directory other than the directory from which the files were backed up.
`verbose`	Turns on or off the verbose option (which can also be entered as `v` on the command line outside of interactive mode). When verbose is on, the interactive `ls` command lists inode numbers and the `ufsrestore` command displays information on each file as it is extracted.
`what`	Displays the backup header from the tape or diskette.

CHAPTER 51

Copying UFS Files and File Systems (Tasks)

This chapter describes how to copy UFS files and file systems to disk, tape, and diskettes by using various backup commands.

This is a list of the step-by-step instructions in this chapter.

- "How to Copy a Disk (`dd`)" on page 688
- "How to Copy Directories Between File Systems (`cpio`)" on page 690
- "How to Copy Files to a Tape (`tar`)" on page 693
- "How to List the Files on a Tape (`tar`)" on page 694
- "How to Retrieve Files From a Tape (`tar`)" on page 694
- "Copying Files to a Tape With the `pax` Command" on page 695
- "How to Copy All Files in a Directory to a Tape (`cpio`)" on page 696
- "How to List the Files on a Tape (`cpio`)" on page 698
- "How to Retrieve All Files From a Tape (`cpio`)" on page 698
- "How to Retrieve Specific Files From a Tape (`cpio`)" on page 699
- "How to Copy Files to a Remote Tape Device (`tar` and `dd`)" on page 700
- "How to Extract Files From a Remote Tape Device" on page 702
- "How to Copy Files to a Single Formatted Diskette (`tar`)" on page 703
- "How to List the Files on a Diskette (`tar`)" on page 704
- "How to Retrieve Files From a Diskette (`tar`)" on page 704
- "How to Create an Archive for Older SunOS Releases" on page 706
- "How to Retrieve `bar` Files From a Diskette" on page 707

Commands for Copying File Systems

When you need to back up and restore complete file systems, use the `ufsdump` and `ufsrestore` commands described in Chapter 50. When you want to copy or move individual files, portions of file systems, or complete file systems, you can use the procedures described in this chapter instead of the `ufsdump` and `ufsrestore` commands.

The following table describes when to use the various backup commands.

TABLE 51–1 When to Use Various Backup Commands

Task	Command	For More Information
Back up file systems to tape	`ufsdump`	"How to Backup a File System to Tape" on page 644
Create a file system snapshot	`fssnap`	Chapter 48
Restore file systems from tape	`ufsrestore`	"How to Restore a Complete File System" on page 667
Transport files to other systems	`pax`, `tar`, or `cpio`	"Copying Files and File Systems to Tape" on page 691
Copy files or file systems between disks	`dd`	"How to Copy a Disk (`dd`)" on page 688
Copy files to diskette	`tar`	"How to Copy Files to a Single Formatted Diskette (`tar`)" on page 703

The following table describes various backup and restore commands.

TABLE 51–2 Summary of Various Backup Commands

Command Name	Aware of File System Boundaries?	Support Multi Volume Backups?	Physical or Logical Copy?
`volcopy`	Yes	Yes	Physical
`tar`	No	No	Logical
`cpio`	No	Yes	Logical
`pax`	Yes	Yes	Logical
`dd`	Yes	No	Physical
`ufsdump/ufsrestore`	Yes	Yes	Logical

The following sections describe the advantages and disadvantages of each method, and provide step-by-step instructions and examples of how to use the commands.

Copying File Systems Between Disks

Two commands are used to copy file systems between disks:

- `volcopy`
- `dd`

The next section describes how to use the `dd` command to copy file systems between disks.

Making a Literal File System Copy

The `dd` command makes a literal (block-level) copy of a complete UFS file system to another file system or to a tape. By default, the `dd` command copies standard input to standard output.

Note – Do not use the `dd` command with variable-length tape drives without first specifying an appropriate block size.

You can specify a device name in place of standard input or standard output, or both. In this example, the contents of the diskette are copied to a file in the `/tmp` directory:

```
$ dd < /floppy/floppy0 > /tmp/output.file
2400+0 records in
2400+0 records out
```

The `dd` command reports on the number of blocks it reads and writes. The number after the + is a count of the partial blocks that were copied. The default block size is 512 bytes.

The `dd` command syntax is different from most other commands. Options are specified as *keyword=value* pairs, where *keyword* is the option you want to set and *value* is the argument for that option. For example, you can replace standard input and standard output with this syntax:

```
$ dd if=input-file of=output-file
```

To use the *keyword=value* pairs instead of the redirect symbols in the previous example, you would type the following:

```
$ dd if=/floppy/floppy0 of=/tmp/output.file
```

▼ How to Copy a Disk (dd)

1. **Make sure that the source disk and destination disk have the same disk geometry.**
2. **Become superuser or assume an equivalent role.**
3. **Create the `/reconfigure` file so the system will recognize the clone disk to be added when it reboots.**

   ```
   # touch /reconfigure
   ```

4. **Shut down the system.**

   ```
   # init 0
   ```

5. **Attach the clone disk to the system.**
6. **Boot the system.**

   ```
   ok boot
   ```

7. **Copy the master disk to the clone disk.**

   ```
   # dd if=/dev/rdsk/device-name of=/dev/rdsk/device-name bs=block-size
   ```

`if=/dev/rdsk/`*device-name*	Represents the overlap slice of the master disk device, usually slice 2.
`of=/dev/rdsk/`*device-name*	Represents the overlap slice of the clone disk device, usually slice 2.
`bs=`*blocksize*	Identifies block size, such as 128 Kbytes or 256 Kbytes. A large block size value decreases the time it takes to copy.

   For more information, see `dd`(1M).

8. **Check the new file system.**

   ```
   # fsck /dev/rdsk/device-name
   ```

9. **Mount the clone disk's root (`/`) file system.**

   ```
   # mount /dev/dsk/device-name /mnt
   ```

10. **Edit the clone disk's `/etc/vfstab` to reference the correct device names.**
    For example, change all instances of `c0t3d0` with `c0t1d0`.
11. **Unmount the clone disk's root (`/`) file system.**

    ```
    # umount /mnt
    ```

12. **Shut down the system.**

```
# init 0
```

13. **Boot from the clone disk to single-user mode.**

```
# boot disk n -s
```

Note – The `installboot` command is not needed for the clone disk because the boot blocks are copied as part of the overlap slice.

---

14. **Unconfigure the clone disk.**

```
# sys-unconfig
```

The system is shut down after it is unconfigured.

15. **Boot from the clone disk again and provide its system information, such as host name, time zone, and so forth.**

```
# boot disk n
```

16. **Log in as superuser to verify the system information after the system is booted.**

hostname `console login:`

Example—Copying a Disk (`dd`)

This example shows how to copy master disk `/dev/rdsk/c0t0d0s2` to clone disk `/dev/rdsk/c0t2d0s2`.

```
# touch /reconfigure
# init 0
ok boot
# dd if=/dev/rdsk/c0t0d0s2 of=/dev/rdsk/c0t2d0s2 bs=128k
# fsck /dev/rdsk/c0t2d0s2
# mount /dev/dsk/c0t2d0s2 /mnt
# cd /mnt/etc
# vi vfstab
(Modify entries for the new disk)
# cd /
# umount /mnt
# init 0
# boot disk2 -s
# sys-unconfig
# boot disk2
```

Copying Directories Between File Systems (cpio Command)

You can use the cpio (copy in and out) command to copy individual files, groups of files, or complete file systems. This section describes how to use the cpio command to copy complete file systems.

The cpio command is an archiving program that copies a list of files into a single, large output file. This command inserts headers between the individual files to facilitate recovery. You can use the cpio command to copy complete file systems to another slice, another system, or to a media device, such as a tape or diskette.

Because the cpio command recognizes end-of-media and prompts you to insert another volume, it is the most effective command, other than ufsdump, to use to create archives that require multiple tapes or diskettes.

With the cpio command, you frequently use the ls and find commands to list and select the files you want to copy, and then pipe the output to the cpio command.

▼ How to Copy Directories Between File Systems (cpio)

1. **Become superuser or assume an equivalent role.**
2. **Change to the appropriate directory.**

   # **cd** *filesystem1*

3. **Copy the directory tree from** *filesystem1* **to** *filesystem2* **by using a combination of the** **find** **and** **cpio** **commands.**

   # **find . -print -depth** | **cpio -pdm** *filesystem2*

.	Starts in the current working directory.
-print	Prints the file names.
-depth	Descends the directory hierarchy and prints file names on the way back up.
-p	Creates a list of files.
-d	Creates directories as needed.

`-m`	Sets the correct modification times on directories.

For more information, see `cpio`(1).

The files from the directory name you specify are copied and symbolic links are preserved.

You might also specify the `-u` option. This option forces an unconditional copy. Otherwise, older files do not replace newer files. This option might be useful if you want an exact copy of a directory, and some of the files being copied might already exist in the target directory.

4. **Verify that the copy was successful by displaying the destination directory contents.**

```
# cd filesystem2
# ls
```

5. **If appropriate, remove the source directory.**

```
# rm -rf filesystem1
```

Example—Copying Directories Between File Systems (`cpio`)

```
# cd /data1
# find . -print -depth | cpio -pdm /data2
19013 blocks
# cd /data2
# ls
# rm -rf /data1
```

Copying Files and File Systems to Tape

You can use the `tar`, `pax`, and `cpio` commands to copy files and file systems to tape. The command that you choose depends on how much flexibility and precision you require for the copy. Because all three commands use the raw device, you do not need to format or make a file system on tapes before you use them.

TABLE 51–3 Advantages and Disadvantages of `tar`, `pax`, and `cpio` Commands

Command	Function	Advantages	Disadvantages
`tar`	Use to copy files and directory subtrees to a single tape	■ Available on most UNIX operating systems ■ Public domain versions are readily available	■ Is not aware of file system boundaries ■ Full pathname length cannot exceed 255 characters ■ Does not copy empty directories or special files such as device files ■ Cannot be used to create multiple tape volumes
`pax`	Use to copy files, special files, or file systems that require multiple tape volumes. Or, when you want to copy files to and from POSIX-compliant systems	■ Better portability than the `tar` or `cpio` commands for POSIX-compliant systems ■ Multi vendor support	Same disadvantages as for the `tar` command, except that the `pax` command can create multiple tape volumes
`cpio`	Use to copy files, special files, or file systems that require multiple tape volumes. Or, when you want to copy files from SunOS 5.9 systems to SunOS 4.0/4.1 systems	■ Packs data onto tape more efficiently than the `tar` command ■ Skips over any bad spots in a tape when restoring ■ Provides options for writing files with different header formats, `tar`, `ustar`, `crc`, `odc`, `bar`, for portability between different system types ■ Creates multiple tape volumes	The syntax is more difficult than the `tar` or `pax` commands

The tape drive and device name that you use depend on the hardware configuration for each system. For more information about tape device names, see "Choosing Which Media to Use" on page 709.

Copying Files to Tape (`tar` Command)

Here are things that you should know before you copy files to tape with the `tar` command:

- Copying files to a tape with the `-c` option to the `tar` command destroys any files already on the tape at or beyond the current tape position.
- You can use file-name substitution wildcards (`?` and `*`) as part of the file names you specify when copying files. For example, to copy all documents with a `.doc` suffix, type `*.doc` as the file-name argument.
- You cannot use file-name substitution wildcards when you extract files from a `tar` archive.

▼ How to Copy Files to a Tape (`tar`)

1. **Change to the directory that contains the files you want to copy.**
2. **Insert a write-enabled tape into the tape drive.**
3. **Copy the files to tape.**

   `$` **`tar cvf /dev/rmt/`***`n`* *filenames*

`c`	Indicates that you want to create an archive.
`v`	Displays the name of each file as it is archived.
`f /dev/rmt/`*`n`*	Indicates that the archive should be written to the specified device or file.
filenames	Indicates the files and directories that you want to copy. Separate multiple files with spaces.

   The file names that you specify are copied to the tape, overwriting any existing files on the tape.
4. **Remove the tape from the drive and write the names of the files on the tape label.**
5. **Verify that the files you copied are on the tape.**

   `$` **`tar tvf /dev/rmt/`***`n`*

   For more information on listing files on a `tar` tape, see "How to List the Files on a Tape (`tar`)" on page 694.

Example—Copying Files to a Tape (`tar`)

The following example shows how to copy three files to the tape in tape drive 0.

```
$ cd /export/home/kryten
$ ls reports
reportA reportB reportC
$ tar cvf /dev/rmt/0 reports
a reports/ 0 tape blocks
a reports/reportA 59 tape blocks
a reports/reportB 61 tape blocks
a reports/reportC 63 tape blocks
$ tar tvf /dev/rmt/n
```

▼ How to List the Files on a Tape (`tar`)

1. **Insert a tape into the tape drive.**
2. **Display the tape contents.**

```
$ tar tvf /dev/rmt/n
```

`t`	Lists the table of contents for the files on the tape.
`v`	Used with the `t` option, and provides detailed information about the files on the tape.
`f /dev/rmt/`*n*	Indicates the tape device.

Example—Listing the Files on a Tape (`tar`)

The following example shows a listing of files on the tape in drive 0.

```
$ tar tvf /dev/rmt/0
drwx--x--x   0/1        0 Jul 14 09:24 2001 reports/
-rw------t   0/1    30000 Jul 14 09:23 2001 reports/reportA
-rw------t   0/1    31000 Jul 14 09:24 2001 reports/reportB
-rw------t   0/1    32000 Jul 14 09:24 2001 reports/reportC
```

▼ How to Retrieve Files From a Tape (`tar`)

1. **Change to the directory where you want to put the files.**
2. **Insert the tape into the tape drive.**
3. **Retrieve the files from the tape.**

```
$ tar xvf /dev/rmt/n [filenames]
```

`x`	Indicates that the files should be extracted from the specified archive file. All files on the tape in the specified drive are copied to the current directory.
`v`	Displays the name of each file as it is retrieved.
`f /dev/rmt/`*n*	Indicates the tape device that contains the archive.
filenames	Specifies a file to retrieve. Separate multiple files with spaces.

For more information, see `tar`(1).

4. **Verify that the files are copied.**

```
$ ls -l
```

Example—Retrieving the Files on a Tape (`tar`)

The following example shows how to retrieve all the files from the tape in drive 0.

```
$ cd /var/tmp
$ tar xvf /dev/rmt/0
x reports/, 0 bytes, 0 tape blocks
x reports/reportA, 0 bytes, 0 tape blocks
x reports/reportB, 0 bytes, 0 tape blocks
x reports/reportC, 0 bytes, 0 tape blocks
x reports/reportD, 0 bytes, 0 tape blocks
$ ls -l
```

Note – The names of the files extracted from the tape must exactly match the names of the files that are stored on the archive. If you have any doubts about the names or paths of the files, first list the files on the tape. For instructions on listing the files on the tape, see "How to List the Files on a Tape (`tar`)" on page 694.

Copying Files to a Tape With the `pax` Command

▼ How to Copy Files to a Tape (`pax`)

1. **Change to the directory that contains the files you want to copy.**
2. **Insert a write-enabled tape into the tape drive.**

3. **Copy the files to tape.**

   ```
   $ pax -w -f /dev/rmt/0 filenames
   ```

`-w`	Enables the write mode.
`-f /dev/rmt/0`	Identifies the tape drive.
filenames	Indicates the files and directories that you want to copy. Separate multiple files with spaces.

   For more information, see `pax`(1).

4. **Verify that the files are copied to tape.**

   ```
   $ pax -f /dev/rmt/0
   ```

5. **Remove the tape from the drive and write the names of the files on the tape label.**

Example—Copying Files to a Tape (`pax`)

The example shows how to use the `pax` command to copy all the files in the current directory.

```
$ pax -w -f /dev/rmt/0 .
$ pax -f /dev/rmt/0
filea fileb filec
```

Copying Files to Tape With the `cpio` Command

▼ How to Copy All Files in a Directory to a Tape (`cpio`)

1. **Change to the directory that contains the files you want to copy.**
2. **Insert a tape that is not write-protected into the tape drive.**
3. **Copy the files to a tape.**

   ```
   $ ls | cpio -oc > /dev/rmt/n
   ```

`ls`	Provides the `cpio` command with a list of file names.
`cpio -oc`	Specifies that the `cpio` command should operate in copy-out mode (`-o`) and write header information in ASCII character format (`-c`). This option ensures portability to other vendor's systems.
`> /dev/rmt/`*n*	Specifies the output file.

All files in the directory are copied to the tape in the drive you specify, overwriting any existing files on the tape. The total number of blocks that are copied is shown.

4. **Verify that the files are copied to tape.**

   ```
   $ cpio -civt < /dev/rmt/n
   ```

5. **Remove the tape from the drive and write the names of the files on the tape label.**

Example—Copying All Files in a Directory to a Tape (`cpio`)

The following example shows how to copy all of the files in the `/export/home/kryten` directory to the tape in tape drive 0.

```
$ cd /export/home/kryten
$ ls | cpio -oc > /dev/rmt/0
92 blocks
$ cpio -civt < /dev/rmt/0
-rw------t    1 kryten     users        400 Jul 14 09:28 2001, b
drwx--x--x    2 kryten     users          0 Jul 14 09:26 2001, letters
-rw------t    1 kryten     users      10000 Jul 14 09:26 2001, letter1
-rw------t    1 kryten     users      10100 Jul 14 09:26 2001, letter2
-rw------t    1 kryten     users      11100 Jul 14 09:27 2001, letter3
-rw------t    1 kryten     users      12300 Jul 14 09:27 2001, letter4
drwx--x--x    2 kryten     users          0 Jul 14 09:27 2001, memos
-rw------t    1 kryten     users        400 Jul 14 09:28 2001, memosmemoU
-rw------t    1 kryten     users        500 Jul 14 09:28 2001, memosmemoW
-rw------t    1 kryten     users        100 Jul 14 09:27 2001, memosmemoX
-rw------t    1 kryten     users        200 Jul 14 09:28 2001, memosmemoY
-rw------t    1 kryten     users        150 Jul 14 09:28 2001, memosmemoZ
drwx--x--x    2 kryten     users          0 Jul 14 09:24 2001, reports
92 blocks
$
```

▼ How to List the Files on a Tape (cpio)

Note – Listing the table of contents takes a long time because the `cpio` command must process the entire archive.

1. **Insert an archive tape into the tape drive.**
2. **List the files on the tape.**

```
$ cpio -civt < /dev/rmt/n
```

`-c`	Specifies that the `cpio` command should read files in ASCII character format.
`-i`	Specifies that the `cpio` command should operate in copy-in mode even though it's only listing files at this point.
`-v`	Displays the output in a format that is similar to the output from the `ls -l` command.
`-t`	Lists the table of contents for the files on the tape in the tape drive that you specify.
`< /dev/rmt/`*n*	Specifies the input file of an existing `cpio` archive.

Example—Listing the Files on a Tape (cpio)

The following example shows how to list the files on the tape in drive 0.

```
$ cpio -civt < /dev/rmt/0
drwx--x--x    2 kryten    users        0 Jul 14 09:34 2001, answers
-rw------t    1 kryten    users      800 Jul 14 09:36 2001, b
drwx--x--x    2 kryten    users        0 Jul 14 09:32 2001, sc.directives
-rw------t    1 kryten    users   200000 Jul 14 09:35 2001, direct241
drwx--x--x    2 kryten    users        0 Jul 14 09:32 2001, tests
-rw------t    1 kryten    users      800 Jul 14 09:36 2001, test13times
396 blocks
```

▼ How to Retrieve All Files From a Tape (cpio)

If the archive was created using relative path names, the input files are built as a directory within the current directory when you retrieve the files. If, however, the archive was created with absolute path names, the same absolute paths are used to re-create the file on your system.

Caution – The use of absolute path names can be dangerous because you might overwrite existing files on your system.

1. **Change to the directory where you want to put the files.**
2. **Insert the tape into the tape drive.**
3. **Extract all files from the tape.**

```
$ cpio -icvd < /dev/rmt/n
```

`-i`	Extracts files from standard input.
`-c`	Specifies that `cpio` should read files in ASCII character format.
`-v`	Displays the files as they are retrieved in a format that is similar to the output from the `ls` command.
`-d`	Creates directories as needed.
`< /dev/rmt/n`	Specifies the output file.

4. **Verify that the files are copied.**

```
$ ls -l
```

Example—Retrieving All Files From a Tape (`cpio`)

The following example shows how to retrieve all files from the tape in drive 0.

```
$ cd /var/tmp
cpio -icvd < /dev/rmt/0
answers
sc.directives
tests
8 blocks
$ ls -l
```

▼ How to Retrieve Specific Files From a Tape (`cpio`)

1. **Change to the directory where you want to put the files.**
2. **Insert the tape into the tape drive.**
3. **Retrieve a subset of files from the tape.**

```
$ cpio -icv "*file" < /dev/rmt/n
```

`-i`	Extracts files from standard input.
`-c`	Specifies that the `cpio` command should read headers in ASCII character format.
`-v`	Displays the files as they are retrieved in a format that is similar to the output from the `ls` command.
`"*`*file*`"`	Specifies that all files that match the pattern are copied to the current directory. You can specify multiple patterns, but each pattern must be enclosed in double quotation marks.
`< /dev/rmt/`*n*	Specifies the input file.

For more information, see `cpio`(1).

4. **Verify that the files are copied.**

```
$ ls -l
```

Example—Retrieving Specific Files From a Tape (`cpio`)

The following example shows how to retrieve all files with the `chapter` suffix from the tape in drive 0.

```
$ cd /home/smith/Book
$ cpio -icv "*chapter" < /dev/rmt/0
Boot.chapter
Directory.chapter
Install.chapter
Intro.chapter
31 blocks
$ ls -l
```

Copying Files to a Remote Tape Device

▼ How to Copy Files to a Remote Tape Device (`tar` and `dd`)

1. **The following prerequisites must be met to use a remote tape drive:**

   a. **The local hostname and optionally, the username of the user doing the copy, must appear in the remote system's `/etc/hosts.equiv` file. Or, the user doing the copy must have his or her home directory accessible on the remote machine,**

and have the local machine name in `$HOME/.rhosts`.

For more information, see `hosts.equiv`(4).

b. An entry for the remote system must be in the local system's `/etc/inet/hosts` file or in the name service `hosts` file.

2. To test whether you have the appropriate permission to execute a remote command, try the following:

```
$ rsh remotehost echo test
```

If `test` is echoed back to you, you have permission to execute remote commands. If `Permission denied` is echoed, check your setup as described in step 1.

3. Change to the directory where you want to put the files.

4. Insert the tape into the tape drive.

5. Copy the files to a remote tape drive.

```
$ tar cvf - filenames | rsh remote-host dd of=/dev/rmt/n obs=block-size
```

`tar cf`	Creates a tape archive, lists the files as they are archived, and specifies the tape device.
`-` (Hyphen)	Represents a place holder for the tape device.
filenames	Identifies the files to be copied.
`|` *rsh remote-host*	Pipes the `tar` command's output to a remote shell.
`dd of=`*/dev/rmt/n*	Represents the output device.
`obs=`*block-size*	Represents the blocking factor.

6. Remove the tape from the drive and write the names of the files on the tape label.

Example—Copying Files to a Remote Tape Drive (`tar` and `dd`)

```
# tar cvf - * | rsh mercury dd of=/dev/rmt/0 obs=126b
a answers/ 0 tape blocks
a answers/test129 1 tape blocks
a sc.directives/ 0 tape blocks
a sc.directives/sc.190089 1 tape blocks
a tests/ 0 tape blocks
a tests/test131 1 tape blocks
6+9 records in
0+1 records out
```

▼ How to Extract Files From a Remote Tape Device

1. **Insert the tape into the tape drive.**
2. **Change to a temporary directory.**

   ```
   $ cd /var/tmp
   ```

3. **Extract the files from a remote tape device.**

   ```
   $ rsh remote-host dd if=/dev/rmt/n | tar xvBpf -
   ```

`rsh` *remote-host*	Indicates a remote shell that is started to extract the files from the tape device by using the `dd` command.
`dd if=/dev/rmt/`*n*	Indicates the input device.
`| tar xvBpf -`	Pipes the output of the `dd` command to the `tar` command that is used to restored the files.

4. **Verify that the files have been extracted.**

   ```
   $ ls -l /var/tmp
   ```

Example—Extracting Files From a Remote Tape Drive

```
$ cd /var/tmp
$ rsh mercury dd if=/dev/rmt/0 | tar xvBpf -
x answers/, 0 bytes, 0 tape blocks
x answers/test129, 48 bytes, 1 tape blocks
20+0 records in
20+0 records out
x sc.directives/, 0 bytes, 0 tape blocks
x sc.directives/sc.190089, 77 bytes, 1 tape blocks
x tests/, 0 bytes, 0 tape blocks
x tests/test131, 84 bytes, 1 tape blocks
$ ls -l
```

Copying Files and File Systems to Diskette

Before you can copy files or file systems to diskette, you must format the diskette. For information on how to format a diskette, see Chapter 19.

Use the `tar` command to copy UFS files to a single formatted diskette.

Use the `cpio` command if you need to copy UFS files to multiple formatted diskettes. The `cpio` command recognizes end-of-media and prompts you to insert the next volume.

Note – The use of the `cpio` command to copy UFS files to multiple formatted diskettes is not a straightforward procedure because of volume management.

Things You Should Know When Copying Files to Diskettes

- Copying files to a formatted diskette by using the `tar -c` command destroys any files that are already on the diskette.
- A diskette that contains a `tar` image is not mountable.
- If you need a multiple-volume interchange utility, use the `cpio` command. The `tar` command is only a single-volume utility.

For more information, see `tar`(1).

▼ How to Copy Files to a Single Formatted Diskette (`tar`)

1. **Change to the directory that contains the files you want to copy.**
2. **Insert a formatted diskette that is not write-protected into the drive.**
3. **Make the diskette available.**

   ```
   $ volcheck
   ```

4. **Reformat the diskette if necessary.**

   ```
   $ rmformat -U /dev/rdiskette
   Formatting will erase all the data on disk.
   Do you want to continue? (y/n)y
   ```

5. **Copy the files to diskette.**

   ```
   $ tar cvf /vol/dev/aliases/floppy0 filename ...
   ```

   The file names that you specify are copied to the diskette, overwriting any existing files on the diskette.

6. **Verify that the files are copied.**

   ```
   $ tar tvf /vol/dev/aliases/floppy0
   ```

For more information on listing files, see "How to List the Files on a Diskette (tar)" on page 704.

7. **Remove the diskette from the drive.**
8. **Write the names of the files on the diskette label.**

Example—Copying Files to a Single Formatted Diskette (tar)

The following example shows how to copy two files to a diskette.

```
$ volcheck
$ cd /home/smith
$ ls evaluation*
evaluation.doc   evaluation.doc.backup
$ tar cvf /vol/dev/aliases/floppy0 evaluation*
a evaluation.doc 86 blocks
a evaluation.doc.backup 84 blocks
$ tar tvf /vol/dev/aliases/floppy0
```

▼ How to List the Files on a Diskette (tar)

1. **Insert a diskette into the drive.**
2. **Make the diskette available.**

   ```
   $ volcheck
   ```

3. **List the files on a diskette.**

   ```
   $ tar tvf /vol/dev/aliases/floppy0
   ```

Example—Listing the Files on a Diskette (tar)

The following example shows how to list the files on a diskette.

```
$ volcheck
tar tvf /vol/dev/aliases/floppy0
rw-rw-rw-6693/10  44032 Jun  9 15:45 evaluation.doc
rw-rw-rw-6693/10  43008 Jun  9 15:55 evaluation.doc.backup
$
```

▼ How to Retrieve Files From a Diskette (tar)

1. **Change to the directory where you want to put the files.**
2. **Insert the diskette into the drive.**

3. **Make the diskette available.**

   ```
   $ volcheck
   ```

4. **Retrieve files from the diskette.**

   ```
   $ tar xvf /vol/dev/aliases/floppy0
   ```

   All files on the diskette are copied to the current directory.

5. **Verify that the files have been retrieved.**

   ```
   $ ls -l
   ```

6. **Remove the diskette from the drive.**

Examples—Retrieving Files From a Diskette (`tar`)

The following example shows how to retrieve all the files from a diskette.

```
$ volcheck
$ cd /home/smith/Evaluations
$ tar xvf /vol/dev/aliases/floppy0
x evaluation.doc, 44032 bytes, 86 tape blocks
x evaluation.doc.backup, 43008 bytes, 84 tape blocks
$ ls -l
```

The following example shows how to retrieve an individual file from a diskette.

```
$ volcheck
$ tar xvf /vol/dev/aliases/floppy0 evaluation.doc
x evaluation.doc, 44032 bytes, 86 tape blocks
$ ls -l
```

The file names that you specify are extracted from the diskette and placed in the current working directory.

▼ How to Archive Files to Multiple Diskettes

If you are copying large files onto diskettes, you want to be prompted to replace a full diskette with another formatted diskette. The `cpio` command provides this capability. The `cpio` commands you use are the same as you would use to copy files to tape, except you would specify `/vol/dev/aliases/floppy0` as the device instead of the tape device name.

For information on how to use the `cpio` command, see "How to Copy All Files in a Directory to a Tape (`cpio`)" on page 696.

Copying Files With a Different Header Format

Archives that are created with the SunOS 5.9 `cpio` command might not be compatible with older SunOS releases. The `cpio` command allows you to create archives that can be read with several other formats. You specify these formats by using the `-H` option and one of these arguments:

- `crc` or `CRC` – ASCII header with checksum
- `ustar` or `USTAR` – IEEE/P1003 Data Interchange
- `tar` or `TAR` – `tar` header and format
- `odc` – ASCII header with small device numbers
- `bar` – `bar` header and format

The syntax for using the header options is as follows:

`cpio -o -H` *header-option* `<` *file-list* `>` *output-archive*

▼ How to Create an Archive for Older SunOS Releases

Use the `cpio` command to create the archive.

`$` **`cpio -oH odc <`** *file-list* **`>`** `/dev/rmt/`*n*

The `-H` arguments have the same meaning for input as they do for output. If the archive was created with the `-H` option, you must use the same option when the archive is read back in or the `cpio` command will fail. The following example includes the `cpio` error message.

Example—Creating an Archive for Older SunOS Releases

```
$ find . -print | cpio -oH tar > /tmp/test
 113 blocks
$ cpio -iH bar < /tmp/test
cpio: Invalid header "bar" specified
USAGE:
        cpio -i[bcdfkmrstuvBSV6] [-C size] [-E file] [-H hdr]
            [-I file [-M msg]] [-R id] [patterns]
        cpio -o[acvABLV] [-C size] [-H hdr] [-O file [-M msg]]
        cpio -p[adlmuvLV] [-R id] directory
```

When you create an archive with different options, always write the command syntax on the media label along with the names of the files or file system on the archive.

If you do not know which `cpio` options were used when an archive was created, all you can do is experiment with different option combinations to see which ones allow the archive to be read.

For a complete list of options, see `cpio`(1).

Retrieving Files Created With the `bar` Command

To retrieve files from diskettes that were archived by using the SunOS 4.0 or 4.1 `bar` command, use the `cpio -H bar` command.

Note – You can use only the `-H bar` option with the `-i` option to retrieve files. You cannot create files with the `bar` header option.

▼ How to Retrieve `bar` Files From a Diskette

1. **Change to the directory where you want to put the files.**
2. **Insert the diskette into the drive.**
3. **Make the diskette available.**

   ```
   $ volcheck
   ```

4. **Retrieve `bar` files from a diskette.**

   All files on the diskette are copied to the current directory.

   ```
   $ cpio -ivH bar < /vol/dev/aliases/floppy0
   ```

CHAPTER 52

Managing Tape Drives (Tasks)

This chapter describes how to manage tape drives.

This is a list of the step-by-step instructions in this chapter.

- "How to Display Tape Drive Status" on page 712
- "How to Retension a Magnetic Tape Cartridge" on page 713
- "How to Rewind a Magnetic Tape Cartridge" on page 714

Choosing Which Media to Use

You typically back up Solaris systems by using the following tape media:

- 1/2-inch reel tape
- 1/4-inch streaming cartridge tape
- 8-mm cartridge tape
- 4-mm cartridge tape (DAT)

You can perform backups with diskettes, but doing so is time-consuming and cumbersome.

The media that you choose depends on the availability of the equipment that supports it and of the media (usually tape) that you use to store the files. Although you must do the backup from a local system, you can write the files to a remote device.

The following table shows typical media that is used for backing up file systems and shows the storage capacity for each media. Capacity depends on the type of drive and the data being written to the tape.

TABLE 52–1 Media Storage Capacities

Media	Capacity
1/2-inch reel tape	140 Mbytes (6250 bpi)
2.5-Gbyte 1/4 inch cartridge (QIC) tape	2.5 Gbytes
DDS3 4-mm cartridge tape (DAT)	12–24 Gbytes
14-Gbyte 8-mm cartridge tape	14 Gbytes
DLT™ 7000 1/2-inch cartridge tape	35–70 Gbytes

Backup Device Names

You specify a tape or diskette to use for backup by supplying a logical device name. This name points to the subdirectory that contains the "raw" device file and includes the logical unit number of the drive. Tape drive naming conventions use a logical, not a physical, device name. The following table shows this naming scheme.

TABLE 52–2 Basic Device Names for Backup Devices

Device Type	Name
Tape	`/dev/rmt/`*n*
Diskette	`/vol/dev/rdiskette0/unlabeled`

In general, you specify a tape drive device as shown in the following figure.

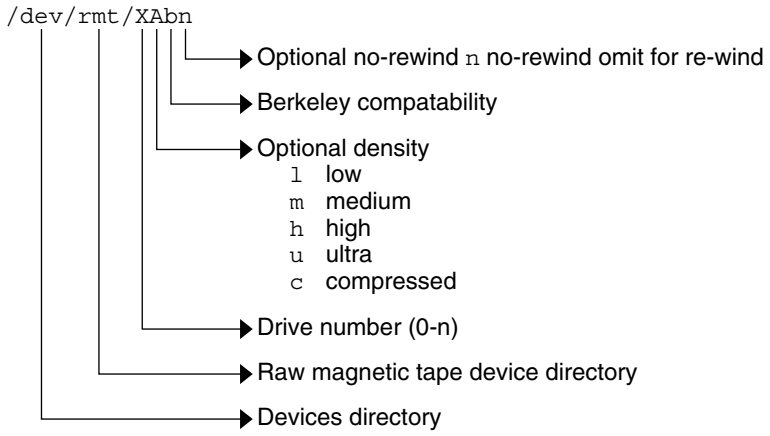


FIGURE 52–1 Tape Drive Device Names

If you don't specify the density, a tape drive typicallys write at its "preferred" density, which usually means the highest density the tape drive supports. Most SCSI drives can automatically sense the density or format on the tape and read it accordingly. To determine the different densities that are supported for a drive, look at the `/dev/rmt` subdirectory. This subdirectory includes the set of tape device files that support different output densities for each tape.

Also, a SCSI controller can have a maximum of seven SCSI tape drives.

Specifying the Rewind Option for a Tape Drive

Normally, you specify a tape drive by its logical unit number, which can run from 0 to *n*. The following table describes how to specify tape device names with a rewind or no rewind option.

TABLE 52–3 Specifying Rewind or No-Rewind for a Tape Drive

Drive and Rewind Value	Use This Option
First drive, rewind	`/dev/rmt/0`
First drive, no rewind	`/dev/rmt/0n`
Second drive, rewind	`/dev/rmt/1`
Second drive, no rewind	`/dev/rmt/1n`

Specifying Different Densities for a Tape Drive

By default, the drive writes at its "preferred" density, which is usually the highest density the tape drive supports. If you do not specify a tape device, the command writes to drive number 0 at the default density the device supports.

To transport a tape to a system whose tape drive supports only a certain density, specify a device name that writes at the desired density. The following table describes how to specify different densities for a tape drive.

TABLE 52–4 Specifying Different Densities for a Tape Drive

Drive, Density, and Rewind Value	Use this Option
First drive, low density, rewind	`/dev/rmt/0l`
First drive, low density, no rewind	`/dev/rmt/0ln`
Second drive, medium density, rewind	`/dev/rmt/1m`
Second drive, medium density, no rewind	`/dev/rmt/1mn`

The unit and density characters are shown in "Backup Device Names" on page 710.

Displaying Tape Drive Status

You can use the `status` option with the `mt` command to get status information about tape drives. The `mt` command reports information about any tape drives that are described in the `/kernel/drv/st.conf` file.

▼ How to Display Tape Drive Status

1. **Load a tape into the drive you want information about.**
2. **Display tape drive status with the `mt` command.**

   ```
   # mt -f /dev/rmt/n status
   ```

3. **Repeat steps 1-2, substituting tape drive numbers 1, 2, 3, and so on to display information about all available tape drives.**

Example—Displaying Tape Drive Status

The following example shows the status for a QIC-150 tape drive (`/dev/rmt/0`) and an Exabyte tape drive (`/dev/rmt/1`).

```
$ mt -f /dev/rmt/0 status
Archive QIC-150 tape drive:
   sense key(0x0)= No Additional Sense   residual= 0   retries= 0
   file no= 0   block no= 0
$ mt -f /dev/rmt/1 status
Exabyte EXB-8200 8mm tape drive:
sense key(0x0)= NO Additional Sense residual= 0  retries= 0
file no= 0   block no= 0
```

The following example shows a quick way to poll a system and locate all of its tape drives.

```
$ for drive in 0 1 2 3 4 5 6 7
> do
> mt -f /dev/rmt/$drive status
> done
Archive QIC-150 tape drive:
   sense key(0x0)= No Additional Sense   residual= 0   retries= 0
   file no= 0   block no= 0
/dev/rmt/1: No such file or directory
/dev/rmt/2: No such file or directory
/dev/rmt/3: No such file or directory
/dev/rmt/4: No such file or directory
/dev/rmt/5: No such file or directory
/dev/rmt/6: No such file or directory
/dev/rmt/7: No such file or directory
$
```

Handling Magnetic Tape Cartridges

If errors occur when a tape is being read, retension the tape, clean the tape drive, and then try again.

▼ How to Retension a Magnetic Tape Cartridge

Retension a magnetic tape cartridge with the `mt` command.

For example:

```
$ mt -f /dev/rmt/1 retension
$
```

Note – Do not retension non-QIC tape drives.

▼ How to Rewind a Magnetic Tape Cartridge

To rewind a magnetic tape cartridge, use the `mt` command.

For example:

```
$ mt -f /dev/rmt/1 rewind
$
```

Guidelines for Drive Maintenance and Media Handling

A backup tape that cannot be read is useless. So, to clean and check your tape drives periodically to ensure correct operation. See your hardware manuals for instructions on procedures for cleaning a tape drive. You can check your tape hardware by doing either of the following:

- Copying some files to the tape, reading the files back, and then comparing the original with the copy.
- Using the `-v` option of the `ufsdump` command to verify the contents of the media with the source file system. The file system must be unmounted or completely idle for the `-v` option to be effective.

Be aware that hardware can fail in ways that the system does not report.

Always label your tapes after a backup. If you are using a backup strategy similar to those strategies suggested in Chapter 46, you should indicate on the label "Tape A," "Tape B," and so forth. This label should never change. Every time you do a backup, make another tape label containing the backup date, the name of the machine and file system backed up, backup level, the tape number (1 of *n*, if it spans multiple volumes), plus any information specific to your site.

Store your tapes in a dust-free safe location, away from magnetic equipment. Some sites store archived tapes in fireproof cabinets at remote locations.

You should create and maintain a log that tracks which media (tape volume) stores each job (backup) and the location of each backed-up file.

Index

Numbers and Symbols

4.3 Tahoe file system, 503
9660 CD format, 226

A

accessing
  disk devices, 407, 410
  pathnames for removable media, 222
  removable media (how to), 228
  tape devices, 410
adding
  a device driver (how to), 348
  a disk (overview)
    SPARC, 458
    x86, 468
  a package, example of, 302
  a package from a mounted CD (example of), 301
  a peripheral device (how to), 346
  a SCSI device to a SCSI bus (how to), 358
  diskless client OS services (how to), 130
  entry to `/etc/vfstab` file (how to), 535
  hot-pluggable USB camera (how to), 390
  multiple versions of a package, 270
  packages (prerequisites), 269
  packages from a spool directory (example of), 304
  packages from remote package server (example of), 302
  packages to a spool directory (example of), 306
adding (Continued)
  packages with administration files, 270
  packages with base directory, 271
  PCI adapter card (how to), 365
  preparing to add OS services for diskless clients (how to), 128
  run control script (how to), 162
  server and client support description, 115
  software with Solaris Product Registry, 276
  swap to `vfstab`, 579
  USB audio device, 394
  USB mass storage device with vold running, 386
  USB mass storage device without vold running, 387
  user initialization files, 82
Admintool
  adding and removing packages overview, 297
  adding packages (how to), 298, 299
  removing packages (how to), 299
aging user passwords, 74, 82, 83
aliases, user login names vs., 71
allocated inodes, 590
appliances, definition, 118
archiving
  files to multiple diskettes with `cpio` command (how to), 705
  for older SunOS releases with `cpio` command (how to), 706
AutoClient, definition, 118
autoconfiguration process, 341

autofs, 517
automounting
  and /home, 518
  user home directories, 76

B

backing up
  a UFS snapshot with the tar command (how to), 658
  and restoring file systems
    commands for, 624
    definition, 624
  choosing file systems to, 625
  file systems to tape (how to), 644
  full and incremental, defined, 627
  preparing for (overview), 642
  reasons for, 625
  types of, 627
  UFS snapshot (full) (how to), 657
  UFS snapshot information (incremental) (how to), 657
backup
  device names, 710
  full (level 0) backup to tape (how to), 644
  record of incremental, 675
backup schedules
  daily cumulative, weekly cumulative backups, 632
  daily cumulative, weekly incremental backups, 633
  daily incremental, weekly cumulative backups, 634
  examples, 632, 638
  for a server, 635
  guidelines for, 630
  recommendations, 638
  using dump levels for, 630
bad block numbers, 591
bad inode number, 593
bad superblock, 598
banner command (PROM), 177
bar command, retrieving files created with (how to), 707
base directory (basedir), 270, 271
basedir keyword (administration files), 270, 271
becoming superuser (root), 48
bin group, 71
block disk device interface
  defined, 407
  when to use, 408
blocks
  bad, 591
  boot, 610
  data, 612
  directory data, 592
  duplicate, 591
  free, 612
  indirect, 592
  logical size, 613
  regular data, 593
  special inodes, 590
boot block, 610
boot-from PROM setting, 179
boot process
  description (SPARC), 208
  x86, 214
boot types, description, 145
booting
  a diskless client (how to), 132
  a system, guidelines, 146
  and PC BIOS, 208
  for recovery purposes (how to)
    SPARC, 186
    x86, 198
  from the network
    SPARC, 184
    x86, 197
  interactively (how to)
    SPARC, 183
    x86, 195
  the Solaris Device Configuration Assistant (how to)
    x86, 193
  to force a crash dump and reboot (how to)
    SPARC, 189
    x86, 204
  to run level 3
    SPARC, 181
  to run level 3 (how to)
    x86, 193
  to run level S
    SPARC, 182

booting (Continued)
to run level S (how to)
x86, 194
with the kernel debugger (how to)
SPARC, 188
x86, 203
Bourne shell
basic features, 91
environment variables and, 92, 96
shell (local) variables and, 92, 94
user initialization files and, 89, 90, 91, 97, 103
Break key, 186, 188
BSD Fat Fast File system, 503
bus-oriented disk controllers, 409
bytes (number per inode), 616

C

C shell
basic features, 91
environment variables and, 92, 96
shell (local) variables and, 92, 94
user initialization files and, 89, 90, 91, 97, 103
CacheFS file systems
checking with fsck command (example of), 558
checking with fsck command (how to), 558
collecting CacheFS statistics (overview), 568
creating (how to), 549
creating a packing list (how to), 562
deleting (how to), 557
displaying information about (how to), 555
displaying packed files (example of), 561
displaying packed files (how to), 561
locating CacheFS log file, 570
mounting (how to), 550
overview, 547
packing files in the cache (how to), 563
packing with cachefspack command (how to), 560
packing with cachefspack command (overview), 559
parameters, 547
setting up CacheFS logging (how to), 570
stopping CacheFS logging, 571
CacheFS file systems (Continued)
troubleshooting cachefspack errors, 564
unpacking files (how to), 563
viewing CacheFS statistics, 572
viewing working set (cache) size, 571
cachefspack command
how to use, 560
overview, 559
cartridge tape, retensioning, 713
causes of file system damage, 586
CD-ROM devices
adding software from mounted CD
example of, 301
CDPATH environment variable, 93
cdrw command
checking CD media (how to), 255
copying a CD (how to), 261
creating a multi-session data CD (how to), 257
creating an audio CD (how to), 259
description, 251
erasing CD-RW media (how to), 261
extracting an audio track on a CD (how to), 260
identifying CD media (how to), 254
restricting access to (how to), 254
writing data and audio CDs (overview), 253
CDs
ISO 9660 format, 226
names, 224
UFS CDs
SPARC vs. x86 format, 226
cfgadm
PCI hot-plugging (overview), 349
SCSI hot-plugging (overview), 349
cfsadmin command, 549, 557
changing
default boot device
SPARC, 179
directory ownership for user accounts, 81
file ownership for user accounts, 81
primary USB audio device (how to), 395
user ID numbers, 81
user login names, 81
user passwords
by user, 73, 74
frequency of, 74, 86
Users Tool, 82

character special inodes, 590
checking
  and repairing file systems, 594
  CacheFS file systems (example of), 558
  CacheFS file systems (how to), 558
  CD media (how to), 255
  file system size, 589
  file systems interactively, 595
  format and type of inodes, 590
  free blocks, 589
  free inodes, 590
  inode list for consistency, 589
  installed packages (example of), 306
clean shutdown, 166
clri command, 507
collecting, CacheFS statistics (overview), 568
configuring
  a SCSI controller with cfgadm command (how to), 355
  a SCSI device with cfgadm command (how to), 356
  a USB device(how to), 402
connecting
  a SCSI controller (how to), 358
  a USB device(how to), 402
controlling file and directory access, 69, 96
copying
  a CD (how to), 261
  all files in a directory to tape with cpio command (how to), 696
  complete file systems (dd), 687
  directories between file systems with cpio command (how to), 690
  directories between file systems with cpio command (overview), 690
  files to diskette (overview), 702
  files to diskette with tar command (how to), 703
  files to remote tape with tar and dd commands (how to), 700
  files to tape with pax command (how to), 695
  files to tape with tar command (how to), 693
  files with different header format with cpio command (how to), 706
  groups of files with cpio command (overview), 690
copying (Continued)
  individual files with cpio command (overview), 690
  removable media information (how to), 229
copying disks, with the dd command (how to), 688
cp command, copying removable media information (how to), 229
cpio command, 696
  copying directories between file systems (how to), 690
  copying files with different header format (how to), 706
  extract all files from tape (how to), 699
  listing files on tape (how to), 698
  overview, 690
  retrieving specific files from tape (how to), 699
creating
  a data CD file system (how to), 256
  a full backup of UFS snapshot information (how to), 656
  a packing list (how to), 562
  a Solaris fdisk partition (how to), 471
  a UFS file system (how to), 522
  a UFS snapshot
    example of, 654
  a UFS snapshot (how to), 654
  an audio CD (how to), 259
  an incremental backup of UFS snapshot (how to), 657
  compatible archives with cpio command (how to), 706
  disk slices and labeling a disk (how to)
    SPARC, 460
    x86, 476
  file systems (overview), 522
  loopback file system (overview), 525
  multi–session data CD (how to), 257
  swap file, 581
  temporary file system (TMPFS) (how to), 524
.cshrc file
  customizing, 75, 91, 97
  description, 89
custom parameters for file systems, 613
customizing user initialization files (how to), 103

cylinder group, 609

D

daemon group, 71
daily cumulative backups, 631
daily discrete backups, 631
damage to file systems, 586
data block, 593, 612
data directory blocks, 592
dd command
  cloning disks (how to), 688
  copying files to remote tape with tar command (how to), 700
  overview, 687
  retrieving files from remote tape drive with tar command (how to), 702
default
  file system for /tmp (TMPFS), 506
  mount options, 537
  SunOS file system, 509
deleting
  CacheFS file systems (how to), 557
  diskless client OS services (example of), 134
  diskless client OS services (how to), 133
  UFS snapshot information example of, 656
  UFS snapshot information (how to), 655
  user home directories, 82
  user mailboxes, 82
detecting end of media
  cpio command, 690
  ufsdump command, 674
determining
  file system types, 518
  mounted file systems, 534
  system's run level (how to), 152
  tape device name, 661
  type of tape drive, 661
  who is logged in to a system, 167
/dev/dsk directory, 407
/dev/rdsk directory, 407
devfsadm command, 406
device driver
  adding, 348
  defined, 340
device instance name, 406
device names
  backup, 710
  finding a file system name, 660
  finding tape, 661
devices
  accessing, 405
  when to turn off power to, 172
df command, 408, 507
dfstab file
  configuring for shared local removable media (how to), 233
  user home directory sharing and, 105
direct I/O, 512
directories
  base directory (basedir), 270, 271
  changing ownership for user accounts, 81
  controlling access to, 69, 96
  copying between file systems with cpio command (overview), 690
  home, 75
  inodes, 590
  PATH environment variable and, 93, 94, 95
  /proc, 506
  skeleton, 75, 82
  /tmp, 506
  unallocated blocks, 593
disabling
  run control script (how to), 162
  user accounts
    passwords and, 83, 86
    Users Tool, 83
disconnecting
  a SCSI controller (how to), 357
  a USB device subtree (how to), 403
disk
  adding to a (overview)
    x86, 468
  automatic configuration of SCSI drives, 450
  connecting a system disk
    x86, 468
  creating disk slices and labeling a disk (how to)
    x86, 476
  formatting a (overview), 429
  repairing defective sectors, 452, 454
  when to format (overview), 438
disk-based file systems, 503
disk controllers, 408

disk label
creating (overview), 443
description, 430
examining with prtvtoc command (how to), 445
disk slices
defined, 421
determining which slices to use, 425
displaying information about (overview), 441
requirements for system configurations, 426
diskettes
accessing on other systems (example of), 236
archiving files to multiple with cpio command (how to), 705
copying files to with tar command (how to), 703
listing files on with tar command (how to), 704
loading
with volume management, 243
loading with volume management (how to), 242
mounting remotely (example of), 236
retrieving files from with tar command (how to), 704
diskless client management commands
smosservice
add OS services, 122
diskless clients
adding OS services for (how to), 130
booting (how to), 132
definition, 118
deleting OS services (example of), 134
deleting OS services (how to), 133
preparing to add OS services (how to), 128
disks
adding to a (overview)
SPARC, 458
connecting a secondary disk (example of)
SPARC, 463
connecting a secondary disk (how to)
SPARC, 459
x86, 469
connecting a system disk (how to)
SPARC, 459
creating a file system on a new disk (how to)
SPARC, 465
creating a file system on a new disk (how to) (Continued)
x86, 478
creating disk slices and labeling a disk (example of)
SPARC, 462
creating disk slices and labeling a disk (how to)
SPARC, 461
determining if formatted (how to), 439
displaying slice information (how to), 441
examining a disk label (how to), 445
formatting a (how to), 439
identifying on a system (how to), 436
labeling a (how to), 443
recovering a corrupted disk label (how to), 447
recovering a corrupted disk label (overview), 446
displaying
detailed information about packages (example of), 305
device information, 345
disk slice information (overview), 441
environment variables, 92
installed software information, 304
packed files (example of), 561
packed files (how to), 561
PCI device information (how to), 364
removable media user (how to), 230
SCSI device configuration information (how to), 354
swap space, 580
system configuration information, 342, 344
UFS snapshot information (how to), 654
USB device information (how to), 400
user mask, 96
dmesg command, 345
SPARC example, 345
x86 example, 346
DOS, file system, 503
driver not attached message, 342
dump levels
daily, cumulative backups, 631
daily, discrete backups, 631
defined, 630
duplicate blocks, 591
DVD-ROM, 505

dynamic reconfiguration, 349

E

eject command, removable media (how to), 231
ejecting, removable media (how to), 231
encryption, 83
end-of-media detection
  cpio command, 690
  ufsdump command, 674
env command, 92
environment variables
  description, 92, 96
  LOGNAME, 93
  LPDEST, 93
  PATH, 93, 95
  SHELL, 94
  TZ, 94
erasing, CD-RW media (how to), 261
/etc/dfs/dfstab file
  configuring for shared removable media (how to), 233
  user home directory sharing and, 105
/etc/dumpdates file, 674
/etc files
  user account information and, 70, 83
/etc/init.d directory, 162
/etc/inittab file
  entry description, 153
  example of default, 154
/etc/passwd file
  description, 83
  fields in, 83
  user ID number assignment and, 71
  recovering
    SPARC, 188
  recovering (example of)
    x86, 200
  deleting user accounts and, 82
/etc/rmmount.conf file, sharing removable media drives (how to), 233
/etc/shadow file, description, 83
/etc/skel directory, 89
/etc/vfstab file, 106
/export/home file system, 75
/export/home directory, 509
exporting shell variables, 92
extended fundamental types (UFS file system), 511
extracting, an audio track on a CD (how to), 260

F

FDFS file system, 507
ff command, 507
field replaceable unit (FRU), 118
FIFO inodes, 590
FIFOFS file system, 507
file system name, 660
file system table, virtual, 516
file systems
  /, 509
  4.3 Tahoe, 503
  BSD Fat Fast, 503
  cached (overview), 547
  checking and repairing, 594
  checking interactively, 595
  checking size, 589
  copying complete (dd), 687
  creating (how to)
    TMPFS, 524
    UFS, 522
  creating (overview)
    loopback (LOFS), 525
  custom parameters, 613
  cylinder group struct, 609
  damage to, 586
  default SunOS, 509
  description of administration commands, 507
  disk-based, 503
  DOS, 503
  /export/home, 509
  FDFS, 507
  FIFOFS, 507
  finding types, 518
  fixing, 597
  High Sierra, 503
  ISO 9660, 503
  large, 531
  making available (overview), 529
  manual pages for, 508

file systems (Continued)
MNTFS, 510
mount table, 515
mounting NFS (how to), 539
NAMEFS, 507
network-based, 504
/opt, 510
PCFS, 503
preening, 596, 597
/proc, 510
process, overview, 506
PROCFS, overview, 506
pseudo, overview, 504
reasons for inconsistencies, 588
restoring complete, 667
restoring complete (how to), 667
sharing, 517
SPECFS, 507
stopping all processes accessing (how to), 542
SWAPFS, 507
TMPFS, 505
types of, 503
UFS, 503
UNIX, 503
unmounting (how to), 543
/usr, 509
/var, 509
which to back up, 625
why you back up, 625
files
archiving to multiple diskettes with cpio command (how to), 705
changing ownership for user accounts, 81
commands for copying to media (overview), 686
controlling access to, 69, 96
copying to diskette with tar command (how to), 703
copying to tape with cpio command (how to), 696
copying to tape with pax command (how to), 695
copying to tape with tar command (how to), 693
/etc/default/fs, 518
/etc/dfs/fstypes, 518
in the /proc directory, 506
files (Continued)
listing on diskette with tar command (how to), 704
listing on tape with cpio command (how to), 698
listing on tape with tar command (how to), 694
restoring interactively (how to), 663
restoring non-interactively (how to), 665
retrieving from diskette with tar command (how to), 704
retrieving from tape with cpio command (how to), 698, 699
retrieving from tape with tar command (how to), 694
sharing, 517
verifying attributes for newly installed packages, 306
finding
file system name, 660
number of tapes for a full backup (how to), 643
PROM revision level, 177
tape device name, 661
tape drive type, 712
type of file system, 518
fixing inconsistent file systems, 597
forget root password
SPARC, 188
x86, 201
format.dat file
creating an entry (how to), 450
creating an entry (overview), 449
keywords, 489, 492
syntax rules, 489
format of inodes, 590
format utility
analyze menu, 486
automatic configuration of SCSI disk drives (how to), 452
automatic configuration of SCSI disk drives (overview), 450
creating a Solaris fdisk partition (how to), 471
creating disk slices and labeling disk (how to)
SPARC, 460
x86, 476

format utility (Continued)
defect menu, 487
determining if a disk is formatted (how to), 439
displaying disk slice information (example of), 442
displaying disk slice information (how to), 441
fdisk menu, 485
features and benefits, 426
formatting a disk (example of), 440
formatting a disk (how to), 439
guidelines for using, 428
how to enter command names, 494
how to specify block numbers, 493
identifying disks on a system (examples of), 438
identifying disks on a system (how to), 436
input to, 493, 495
labeling a disk
example of, 444
labeling a disk (how to), 443
main menu, 482
overview, 426
partition menu, 484, 485
recommendations for preserving information, 481
recovering corrupted disk label (how to), 447
using help facility, 495
when to use, 427
formatting a disk, overview, 429
fragment size, 614
free blocks, 589, 612
free hog slice
See donor slice
free inodes, 590
free space (minimum), 615
fsck command, 408, 507
checking
free blocks, 589
free inodes, 590
inode list size, 589
superblock, 589
conditions to repair, 588
FSACTIVE state flag, 586
FSBAD state flag, 586
FSCLEAN state flag, 586
fsck command (Continued)
FSSTABLE state flag, 586
preening, 596
state flags, 586
syntax and options, 600
using interactively, 594
fsdb command, 508
fssnap command
creating a UFS snapshot (how to), 654
deleting UFS snapshot information (how to), 655
displaying UFS snapshot information (how to), 654
fstyp command, 508
fstypes file, 518
full backup
creating with the ufsdump command (how to), 644
definition, 628
determine number of tapes for (how to), 643
example of, 645, 647
to a remote system
example of, 648
fuser command
finding if removable media is in use (how to), 230
killing processes accessing removable media (how to), 230

G

GECOS field (passwd file), 84
GIDs, 71
assigning, 77
definition, 76
large, 72
grep command, 518
group file
deleting user accounts and, 82
description, 83
fields in, 86
group ID numbers, 71, 76, 77
groups
changing primary, 76
default, 77
description, 69, 76
description of names, 76

groups (Continued)
displaying groups a user belongs to, 76
guidelines for managing, 76, 77
ID numbers, 71, 76, 77
name services and, 77
names
description, 76
permissions setting for, 96
primary, 76, 77
secondary, 76, 77
storage of information for, 83, 86
UNIX, 76
groups command, 76

H

halt command, 166
header format, copying files with different with cpio command (how to), 706
High Sierra file system, 503
history environment variable, 93
/home (automounted), 518
HOME environment variable, 93
/home file system, user home directories and, 75
hot-plugging
adding a SCSI device to a SCSI bus (how to), 358
adding PCI adapter card (how to), 365
configuring a SCSI controller (how to), 355
configuring a SCSI device (how to), 356
connecting a SCSI controller (how to), 358
disconnecting a SCSI controller with cfgadm command (how to), 357
overview, 349
PCI devices (overview), 364
removing a SCSI device (how to), 360
removing PCI adapter card (how to), 365
replacing an identical device on a SCSI controller (how to), 359
unconfiguring a SCSI controller (how to), 355
HSFS
See High Sierra file system

I

I/O, direct, 512
ID numbers
group, 71, 76, 77
user, 71, 81
identifying
CD media (how to), 254
devices, 343
disks on a system (how to), 436
inconsistencies in file systems, 588
incorrect . and .. entries, 593
incremental backup, 628, 675
example of, 646
indirect blocks, 592
init command
description, 166
shutting down a standalone system, 171
init states
See run levels
initialization files, system, 76
inode list size, 589
inode states, 590
inodes, 610
bad number, 593
block special, 590
character special, 590
checking format and type, 590
directory, 590
FIFO, 590
link count, 591
number of bytes per, 616
regular, 590
size, 592
symbolic link, 590
installboot command, 466, 479
installing a boot block (how to)
SPARC, 466
x86, 479
interactive
checking file systems, 595
restore (how to), 663
ISO 9660 file system, 503
ISO standards, 9660 CD format, 226

K

/kernel/drv directory, 341
killing
  all processes accessing a file system (how to), 542
  processes accessing removable media (how to), 230
Korn shell
  basic features, 91
  environment variables and, 92, 96
  shell (local) variables and, 92, 94
  user initialization files and, 89, 90, 91, 97, 103

L

L1-A keys, 186, 188
labelit command, 508
LANG environment variable, 93, 95, 96
large files option, 531
LC environment variables, 95, 96
level 0 backup, 630
link count of inodes, 591
listing
  files on a diskette with tar command (how to), 704
  files on a tape with cpio command (how to), 698
  files on a tape with tar command (how to), 694
  package information (example of), 304
LK password, 83, 86
loading
  diskettes
    with volume management, 243
  diskettes with volume management (how to), 242
local.cshrc file, 89
local.login file, 89
local.profile file, 89
locale environment variable, 93
locating, CacheFS log file, 570
log (record of dumps), 674
logical block size, 613
logical device name
  definition, 406
  disk, 407
logical device name (Continued)
  tape, 410
logical device names, removable media, 411
.login file
  customizing, 75, 91, 97
  description, 89
login names (user)
  changing, 81
  description, 70
LOGNAME environment variable, 93
loopback file system (LOFS)
  creating (overview), 525
  mounting, 535
lost+found directory, 586
LPDEST environment variable, 93

M

magnetic tape cartridge
  retensioning, 713
  rewinding, 714
mail aliases, user login names vs., 71
MAIL environment variable, 92, 93
maintaining tape drives, 714
MANPATH environment variable, 93
manual mounting, remote media (how to), 235
manual pages, for file systems, 508
maximum, USB device support, 378
maximums
  secondary groups users can belong to, 76
  user ID number, 71
  user login name length, 70
  user password length, 73
media was found message, 243
memory storage (virtual), definition, 576
minimum free space, 615
minimums
  user login name length, 70
  user password length, 73
mkfile command, 581, 582
mkfs command, 508, 522
mkisofs command, create a data CD file system (how to), 256
MNTFS file system, 510
mnttab file, 515
monitor (PROM), 207
mount command, 408

mount point, definition, 513
mount table, 515
mountall command, 508
mounting
  a file system with /etc/vfstab, 536
  all files in vfstab file, 535
  diskettes on other systems (example of), 236
  file systems automatically, 517
  loopback file systems (LOFS), 535
  NFS file systems, 535
  NFS file systems (how to), 539
  PCMCIA memory cards on other systems (example of), 237
  remote media (how to), 235
  remote removable media manually (example of), 236
  removable media
    automatic mounting compared to, 221
  UFS file systems, 535
  UFS file systems (how to)
    without large files, 538
  USB mass storage devices with vold running (how to), 390
  USB mass storage devices without vold running (how to), 390
  user home directories
    automounting, 76
    remote, 105
  user home directories (how to), 106
  using default options, 537
mt command, 713
multiple versions of software packages, 270, 271
multiuser level
  See run level 3

N

name services
  groups and, 77
  user accounts and, 70, 83
NAMEFS file system, 507
names
  group
    description, 76
  software package naming conventions, 269
  SUNW prefix, 269
names (Continued)
  user login
    changing, 81
    description, 69, 70
ncheck command, 508
network-based file systems, 504
newfs command, 408, 522, 617
newgrp command, 76
NFS
  description, 517
  server description, 517
  vfstab entry for, 535
nfsd daemon
  starting, 233
  verifying if running, 232
NIS
  user accounts and, 70, 83
NIS+
  groups and, 77
  user accounts and, 70, 83
no media was found message, 243
noaccess user/group, 71, 87
noask_pkgadd administration file, 271, 302
nobody user/group, 71, 87
notifying users of system down time, 167
NP password, 86

O

/opt directory, 510
optimization type, 615
options, for ufsdump command, 678
OS server, description, 122
other (permissions setting), 96

P

packages, software, overview, 265
parameters (file system), 613
partition (swap), definition, 576
passwd file, 83
  deleting user accounts and, 82
  fields in, 83, 84
  recovering
    SPARC, 188

passwd file (Continued)
recovering (example of)
x86, 200
restoring from tape (example of), 666
user ID number assignment and, 71
passwords (user)
aging, 74, 82, 83
changing
frequency of, 74, 86
by user, 73, 74
Users Tool, 82
choosing, 74
description, 69, 73, 74
disabling/locking user accounts and, 83, 86
encryption, 83
expiration, 86
NP password, 86
LK password, 83, 86
precautions, 73, 74
setting, 73, 82
Users Tool, 82
patchadd command, 333
patches
accessing from the world wide web, 312
availability for Sun Service customers, 312
definition, 309
finding already installed, 332
general availability, 312
installation README, 313
numbering scheme, 312
removing, 335
tools and commands (overview), 313
where to find, 312
patchrm command, 335
PATH environment variable
description, 93, 94
setting up, 94, 95
path shell variable, 92
PC BIOS (and booting), 208
PCFS file system, 503
PCI devices
adding PCI adapter card (how to), 365
displaying PCI device information (how to), 364
removing PCI adapter card (how to), 365
troubleshooting PCI configuration problems, 367
PCMCIA memory cards
accessing on other systems (example of), 237
mounting remotely (example of), 237
permissions, 96
physical device name, definition, 406
/pkg directory, 304
pkgadd command
adding packages (how to), 300
alternate base directory and, 271
bypassing user interaction, 270, 271
overview, 268, 274
-a option (administration file), 270, 271, 300, 302
-d option (device name), 300, 301, 302, 303, 304
-s option (spool directory), 303
prerequisites for using, 269
spool directories and, 303
spool directories and (example of), 304
pkgchk command
overview, 274
using (example of), 306
pkginfo command
displaying all packages installed (example of), 304
how to use, 304
overview, 269, 274
pkgparam command, overview, 274
pkgrm command
caution, 270, 306
overview, 268, 274, 306
prerequisites for using, 269
removing a package (how to), 306
rm command vs., 270, 306
playing musical CD or DVD, 229
preening file systems, 596, 597
preparing
for backing up (overview), 642
to restore files (overview), 660
Primary Administrator role
assuming (how to), 53
creating (how to), 52
creating (overview), 51
primary groups, 76, 77
/proc directory, 506, 510
process file system (PROCFS), 506
PROCFS file system, overview, 506

prodreg command
checking dependencies between software products (how to), 285
identifying damaged software (how to), 286
listing information about installed products (how to), 280
listing software attributes (how to), 283
overview, 274, 279
reinstalling damaged software (how to), 295
uninstalling damaged software (how to), 293
uninstalling software (how to), 289
Product Registry
adding software with, 276
checking dependencies between software products (how to), 285
identifying damaged software (how to), 286
installing software with (how to), 278
listing information about installed products (how to), 277, 280
listing software attributes (how to), 283
purpose, 276
reinstalling damaged software (how to), 295
removing software with, 276
uninstalling damaged software (how to), 293
uninstalling software (how to), 289
uninstalling software with (how to), 279
.profile file
customizing, 75, 91, 97
description, 89
PROM
changing boot-from setting, 179
finding revision level, 177
finding the ROM revision, 177
monitor, 207
prompt shell variable, 94
prtconf command, 343
prtvtoc command, 408
example of using, 445
PS1 environment variable, 94
pseudo file systems, overview, 504
pseudo-ttys, 71
pseudo user logins, 71

R

raw disk device interface, 407, 408
reboot command, 166
reconfiguration boot, 451
SPARC example, 460
x86 example, 469
record of
dumps, 674
incremental backup, 675
recover root password (how to)
SPARC, 188
x86, 201
regular inodes, 590
remote drive (restoring from), 666
remote mounting, 105
remote package server
adding packages to a spool directory (example of), 303
software installation from, 302
software installation from (example of), 302
removable media
accessing (examples of), 228
accessing (how to), 228
accessing media on other systems (example of), 236
accessing media on other systems (how to), 235
copying information (how to), 229
ejecting (how to), 231
finding out if media is in use (how to), 230
killing processes accessing (how to), 230
making available to other systems (how to), 232
mounting
manual compared to automatic, 221
mounting remote media (example of), 236
mounting remote media (how to), 235
musical CD or DVD, 229
names, 224
preparing for new drive (how to), 226
removef command, 270
removing
a SCSI device (how to), 360
a swap file from use, 583
packages with administration files and, 271
PCI adapter card (how to), 365
software packages
guidelines for, 270

removing (Continued)
software packages (how to), 306
software with Solaris Product Registry, 276
unused USB audio device links (how to), 397
USB mass storage device with vold running, 387
USB mass storage device without vold running, 387
repairing the /etc/passwd file
SPARC, 188
x86, 200
replacing, an identical device on a SCSI controller (how to), 359
reset command, 180
resetting
a SPARC based system, 180
a USB device (how to), 403
resolving, a failed SCSI unconfigure operation (how to), 363
restoring bad superblock, 598
restoring file systems
complete (example), 668
complete (example of), 667
determining which tapes to use (how to), 662
preparing to (overview), 660
root and /usr (how to), 670
root and /usr (SPARC) (example of), 671
root and /usr (x86) (example of), 672
type of tape drive, 661
restoring files
example of interactive restore, 664
example of non-interactive restore, 666
from remote drive (example of), 666
interactively (how to), 663
non-interactively (how to), 665
restricting, removable media access (how to), 254
retensioning magnetic tape cartridge, 713
retrieving
files created wth bar command (how to), 707
files from a tape with cpio command (how to), 698
files from diskette with tar command (how to), 704
retrieving (Continued)
files from remote tape with tar and dd commands (how to), 702
files from tape withtar command (how to), 694
specific files from tape with cpio command (how to), 699
revision level of PROM, 177
rewinding magnetic tape cartridge, 714
rmmount.conf file, sharing removable media drives (how to), 233
Rock Ridge extension (HSFS file system), 503
root (/) file system, 509
root (superuser), becoming, 48
root password, forget
SPARC, 188
x86, 201
run control scripts, 156
adding (how to), 162
disabling (how to), 162
starting and stopping services, 161
run level
0 (power-down level), 152
1 (single-user level), 152
2 (multiuser level), 152
3 (multiuser with NFS), 152
booting to, 181, 193
processes executed at, 155
what happens when system is brought to, 155
6 (reboot level), 152
default run level, 151
definition, 151
determining (how to), 152
s or S (single-user level), 152
booting to, 194
s or S (single-user state)
booting to, 182

S

/sbin/rc0 script, 157
/sbin/rc1 script, 157
/sbin/rc2 script, 157
/sbin/rc3 script, 159
/sbin/rc5 script, 160
/sbin/rc6 script, 160

/sbin/rcS script, 160
scheduling backups, 630
SCSI devices
  adding a SCSI device to a SCSI bus (how to), 358
  configuring with cfgadm command (how to), 356
  connecting with cfgadm command (how to), 358
  disconnecting with cfgadm command (how to), 357
  displaying with cfgadm command (how to), 354
  removing with cfgadm command (how to), 360
  replacing an identical device on a SCSI controller (how to), 359
  resolving a failed SCSI unconfigure operation (how to), 363
  troubleshooting SCSI configuration problem, 361
  unconfiguring with cfgadm command (how to), 355
SCSI disk drives, 450
SCSI tape drives, 711
secondary disk
  connecting to the system (how to)
    SPARC, 460
    x86, 470
  description, 425
secondary groups, 76, 77
security, user ID number reuse and, 71
servers
  description, 116
  OS server, 122
set command, 92
setenv command, 92
setting up, CacheFS logging, 570
shadow file
  description, 83
  fields in, 85, 86
share command, 517
  making removable media available to other systems (how to), 233
shareall command, 517
sharing
  files, 517
  removable media (how to), 232
sharing (Continued)
  user home directories, 105
  user home directories (how to), 104
SHELL environment variable, 94
shell variables, 93, 94
shells
  basic features, 91
  environment of, 92, 94
  environment variables and, 92, 96
  local variables, 92, 93, 94
  user initialization files and, 88, 90, 91, 97, 103
shutdown command
  description, 166
  notifying users, 167
  shutting down a server, 146
  shutting down a server (how to), 168
shutting down
  a server (how to), 167
  a standalone system (how to), 171
  a system, guidelines, 145
  a system cleanly with shutdown and init commands, 166
single-user level
  See run level s or S
site initialization files, 90
size
  checking file system, 589
  fragment, 614
  inode, 592
/skel directory, 89
skeleton directories (/etc/skel), 75, 82
slice (defined), 421
software management
  definition, 267
  naming conventions for packages, 269
  package definition, 267
  tools for, 268
software packages
  installing, 304
  installing from a spool directory (example of), 303
  overview, 265
Solaris Device Configuration Assistant, overview, 192
Solaris fdisk partition, guidelines, 470
Solaris Management Console
  description, 41

Solaris Management Console (Continued)
description of tools, 42
reasons for using, 44
starting (how to), 54
using with RBAC, 49
Solaris Product Registry
adding software with, 276
checking dependencies between software products (how to), 285
identifying damaged software (how to), 286
installing software with (how to), 278
listing information about installed products (how to), 277
listing software attributes (how to), 283
purpose, 276
reinstalling damaged software (how to), 295
removing software with, 276
uninstalling damaged software (how to), 293
uninstalling software (how to), 289
uninstalling software with (how to), 279
Solaris User Registration
See User Registration
Solaris Web Start, adding software with (how to), 275
space optimization type, 615
SPARC based systems, UFS format, 226
SPECFS file system, 507
specifying a disk slice, 408, 410
spool directories
installing software packages to (example of), 304, 306
installing software packages to (how to), 303
`staff` group, 77
standalone systems, definition, 117
starting
`nfsd` daemon, 233
volume management (how to), 227
starting and stopping services, 161
state flag
`fsck`, 586
UFS file systems, 511
`Stop-A` keys, 186, 188
stopping
a system for recovery purposes
SPARC, 186
a system for recovery purposes (how to)
x86, 198
stopping (Continued)
all processes for a file system (how to), 542
CacheFS logging, 571
killing processes accessing removable media (how to), 230
volume management (how to), 227
storage (virtual memory), definition, 576
storage capacities (media), 628, 709
structure of cylinder groups, 609
`stty` command, 95
Sun software packages
adding (example of), 301
installing, 302
SunOS default file system, 509
`SUNW` prefix, 269
superblock, 589, 598, 610
superuser (root), becoming, 48
superuser (root) password, forget
SPARC, 188
x86, 201
support for servers and clients, description, 115
`swap` command, 581
swap file
adding to `vfstab`, 579
creating, 581
displaying, 580
removing from use, 583
swap partition, definition, 576
`swapadd` command, 579
SWAPFS file system, 507
symbolic links, 590
`sync` command, 188, 190
synchronize file systems with `sync` command, 190
synchronize the file systems with `sync` command, 188
syntax
`fsck` command, 600
newfs, 617
`sysdef` command, 343
system accounts, 71
system disk
connecting (how to)
SPARC, 459
x86, 469
description, 425
installing a boot block on (how to)
SPARC, 466

installing a boot block on (how to) (Continued)
x86, 479
system initialization files, 76
system shutdown commands, 166
system types
appliance, 118
AutoClient, 118
diskless client, 118
guidelines for choosing, 118
overview, 116
server, 116
standalone system, 117

T

tape, 714
capacity, 677
characteristics, 677
copying all files in a directory with cpio command (how to), 696
listing files with tar command (how to), 694
retrieving files from with cpio command (how to), 698
retrieving files from with tar command (how to), 694
retrieving specific files from with cpio command (how to), 699
sizes, 628, 709
storage capacities, 628, 709
tape (magnetic cartridge), retensioning, 713
tape devices (naming), 410
tape drive
determining type for restore, 661
finding type, 712
maintaining, 714
maximum SCSI, 711
restoring from remote (example of), 666
rewind, 711
tar command
copying files to a single diskette (how to), 703
copying files to remote tape with dd command (how to), 701
copying files to tape (how to), 693
listing files on diskette (how to), 704
listing files on tape (how to), 694
tar command (Continued)
overview, 693
retrieving files from diskette (how to), 705
retrieving files from remote tape with dd command (how to), 702
retrieving files from tape (how to), 694
temporary file system (TMPFS), overview, 505
TERM environment variable, 94
TERMINFO environment variable, 94
time (optimization type), 615
time zone environment variable, 94
/tmp directory, 506, 510
TMPFS file system
creating (how to), 524
overview, 505
troubleshooting
cachefspack errors, 564
diskless client problems, 137
PCI configuration problems, 367
SCSI configuration problems, 361
USB audio device problems, 397
ttys (pseudo), 71
ttytype pseudo user logins, 71
turn off power to all devices, how to, 173
type of file systems, 503
type of inodes, 590
type of tape drive, 712
TZ environment variable, 94

U

UDF file system, 505
UFS CDs, SPARC vs. x86 formats, 226
UFS file system, 503, 510
creating (how to), 522
extended fundamental types, 511
large file systems, 511
mounting, 535
mounting with /etc/vfstab, 536
mounting with mount command, 537
mounting without large files (how to), 538
state flags, 511
unmounting (how to), 543
UFS logging, overview, 511
UFS snapshot
backing up with the tar command (how to), 658

UFS snapshot (Continued)
creating (how to), 654
creating a full backup of, 657
creating a full backup of (howto), 656
creating an incremental backup of (how to), 657
deleting (how to), 655
description, 652
displaying (how to), 654
ufsdump command
backing up file systems to tape (how to), 644
end-of-media detection, 674
full backup example, 645, 647
full backup to remote system example of, 648
how data is copied with, 674
how it works, 673
incremental backup example, 646
limitations, 677
options and arguments, 678
ufsdump command (overview), 644
ufsrestore command, 681
determining which tapes to use (how to), 662
interactive restore (how to), 663
non-interactive restore (how to), 665
preparing to use (overview), 660
restoring a complete file system (how to), 667
restoring from a remote tape drive (example of), 666
restoring root (/) and /usr file systems (how to), 670
UIDs, 81
assigning, 71
definition, 71
large, 72
umask command, 96
umount command, 508
umountall command, 508
unallocated directory blocks, 593
unallocated inodes, 590
unconfiguring
a SCSI controller with cfgadm command (how to), 355
a USB device (how to), 401
UNIX file system, 503
UNIX groups, 76
unmounting
file systems (how to), 543
USB mass storage devices with vold running (how to), 390
USB mass storage devices without vold running (how to), 390
unsupported devices, 341, 342
USB camera, adding (how to), 390
USB devices
acronyms, 379
adding a hot-plug a USB camera (how to), 390
adding a USB mass storage device with vold running, 386
without vold running, 387
audio
adding a, 394
changing the primary device (how to), 395
device ownership, 398
identifying primary device (how to), 394
overview of, 392
removing unused device links (how to), 397
bus description, 379
cables for, 384
composite device, 380
compound device, 380
configuring a USB device (how to), 402
connect a USB device (how to), 402
connect a USB device subtree (how to), 403
device classes, 381
device nodes, 381
displaying devcie information (how to), 400
drivers, 381
host controller and root hub, 383
hot-plugging (overview), 386
keyboards and mouse devices, 382
maximum devices supported, 378
mounting mass storage device without vold running *how to), 390
mounting mass storage with vold running (how to), 389
names of, 380
overview, 378
physical device hierarchy, 379
power management, 384

USB devices (Continued)
removing a mass storage device
with vold running, 387
removing a USB mass storage device
without vold running, 387
resetting a USB device (how to), 403
Solaris USB Architecture (USBA), 381
storage devices, 385
supported, 378
troubleshooting audio device problems, 397
unconfiguring a USB device (how to), 401
unmounting mass storage device without
`vold` running (how to), 390
unmounting mass storage with vold running
(how to), 389
user accounts, 69
description, 69
disabling/locking
passwords and, 83, 86
Users Tool, 83
guidelines for, 70, 76
ID numbers, 71, 81
login names, 69, 70, 81
name services and, 70, 83
setting up
information sheet, 102
storage of information for, 70, 83
user home directories
changing ownership of, 81
customized initialization files in, 75, 82
deleting, 82
description, 69, 75
mounting
automounting, 76
remote, 105
mounting (how to), 106
nonlocal reference to (`$HOME`), 75, 90
sharing, 105
sharing (how to), 104
user ID numbers, 71, 81
user initialization files
customizing, 88, 97
adding customized files, 82
avoiding local system references, 90
environment variables, 92, 96
overview, 75, 88, 89
shell variables, 93, 94
site initialization files, 90
customizing (Continued)
user mask setting, 96
customizing (how to), 103
default, 89
description, 69, 75, 76, 88
examples, 97
shells and, 88, 90, 91, 97
user login names
changing, 81
description, 69, 70
user logins (pseudo), 71
user mask, 96
User Registration
description, 107
disabling, 110
problems, 108
`solregis` command, 107
Users Tool
disabling accounts, 83
password administration, 82
`/usr` file system, 509
`uucp` group, 71

V

`/var` directory, 509
`/var/sadm/install/admin` directory, 270
`/var/sadm/patch`, 333
`/var/spool/pkg` directory, 302, 304
variables
environment, 92, 96
shell (local), 92, 94
verifying
`nfsd` daemon is running, 232
software installation (example of), 306
software package installation
`pkginfo` command, 303
software package installation with `pkginfo`
command, 303
`vfstab` file, 518, 579
adding entries to (how to), 535
adding swap to, 579
default, 516
entry for LOFS, 527
finding file system names in, 642
mounting all files, 535

viewing
CacheFS statistics, 572
working set (cache) size, 571
virtual file system table, 516
virtual memory storage, definition, 576
`volcopy` command, 508
`volmgt start` command, 227
volume management
benefits, 220
diskettes
loading, 243
loading diskettes (how to), 242
manual compared to automatic
mounting, 221
removable media
accessing, 222
restarting (how to), 227
stopping (how to), 227

W

when to turn off power to devices, 172
`who` command, 152, 167
world (permissions), 96
writing, data and audio CDs (overview), 253

X

x86 based systems, UFS format, 226