Ce logiciel et la documentation qui l’accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d’utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d’interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu’elles soient exemptes d’erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l’accompagne, est concédé sous licence au Gouvernement des États-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l’utilise pour le compte du Gouvernement des États-Unis, la notice suivante s’applique:

U.S. GOVERNMENT END USERS. Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d’applications de gestion des informations. Ce logiciel ou matériel n’est pas conçu ni n’est destiné à être utilisé dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d’applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de securité, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l’utilisation de ce logiciel ou matériel pour ce type d’applications.

Oracle et Java sont des marques déposées d’Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d’autres propriétaires qu’Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d’Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées d’Advanced Micro Devices. UNIX est une marque déposée d’The Open Group.

Ce logiciel ou matériel et la documentation qui l’accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l’accès à des contenus, produits ou services tiers, ou à leur utilisation.
Contents

Preface ................................................................................................................................................... 17

1 Managing Removable Media (Overview/Tasks) ............................................................................. 19

Managing Removable Media (Overview) .......................................................................................... 19
  Removable Media Features and Benefits ..................................................................................... 19
  Comparison of Manual and Automatic Mounting ......................................................................... 21
  Overview of Accessing Removable Media .................................................................................... 22
Managing Removable Media (Tasks) ................................................................................................ 23
  Removable Media Considerations ............................................................................................... 23
  Formatting Diskettes ..................................................................................................................... 24
    ▼ How to Load Removable Media ............................................................................................. 24
    ▼ How to Format a Diskette (\r\nformat) ................................................................................. 25
    ▼ How to Create a File System on Removable Media .............................................................. 26
    ▼ How to Create a File System on a DVD-RAM .................................................................... 27
    ▼ How to Check a File System on Removable Media ............................................................. 27
    ▼ How to Repair Bad Blocks on Removable Media .................................................................. 28
      Applying Read or Write Protection and Password Protection to Removable Media .......... 28
    ▼ How to Enable or Disable Write Protection on Removable Media ........................................ 29
    ▼ How to Enable or Disable Read or Write Protection and Set a Password on Removable
       Media ....................................................................................................................................... 29
Accessing Removable Media ............................................................................................................. 30
  Using Removable Media Names ..................................................................................................... 30
    Guidelines for Accessing Removable Media Data .................................................................... 31
    ▼ How to Add a New Removable Media Drive ......................................................................... 31
    ▼ How to Disable or Enable Removable Media Services ......................................................... 31
    ▼ How to Access Information on Removable Media ................................................................. 32
    ▼ How to Determine If Removable Media Is Still in Use .......................................................... 32
    ▼ How to Eject Removable Media ............................................................................................. 33
Contents

Accessing Removable Media on a Remote System ............................................................... 33
 ▼ How to Make Local Media Available to Other Systems .................................................. 33
 ▼ How to Access Removable Media on Remote Systems .................................................. 35

2 Writing CDs and DVDs (Tasks) ....................................................................................... 37
 Working With Audio CDs and Data CDs and DVDs ............................................................. 37
 CD/DVD Media Commonly Used Terms ............................................................................. 38
 Writing CD and DVD Data and Audio CDs ......................................................................... 39
 Restricting User Access to Removable Media With RBAC .................................................. 40
 ▼ How to Restrict User Access to Removable Media With RBAC ....................................... 41
 ▼ How to Identify a CD or DVD Writer ............................................................................... 41
 ▼ How to Check the CD or DVD Media ............................................................................. 42
 Creating a Data CD or DVD .................................................................................................. 42
 ▼ How to Create an ISO 9660 File System for a Data CD or DVD ....................................... 42
 ▼ How to Create a Multi-Session Data CD ......................................................................... 43
 Creating an Audio CD ......................................................................................................... 45
 ▼ How to Create an Audio CD ............................................................................................ 45
 ▼ How to Extract an Audio Track on a CD .......................................................................... 46
 ▼ How to Copy a CD ............................................................................................................ 47
 ▼ How to Erase CD-RW Media ........................................................................................... 48

3 Managing Devices (Tasks) .............................................................................................. 49
 What’s New in Device Management? .................................................................................. 49
 Support for USB 3.0 Devices .............................................................................................. 49
 Where to Find Additional Device Management Tasks .......................................................... 50
 Managing Devices in Oracle Solaris .................................................................................... 50
 x86: Identifying Device Support .......................................................................................... 50
 About Device Drivers .......................................................................................................... 51
 Automatic Configuration of Devices .................................................................................... 51
 Displaying Device Configuration Information ....................................................................... 52
 Resolving Faulty Devices .................................................................................................... 57
 Adding a Peripheral Device to a System ............................................................................. 59
 ▼ How to Add a Peripheral Device ..................................................................................... 59
 ▼ How to Add a Device Driver ............................................................................................ 60
 Accessing Devices .............................................................................................................. 61
How Device Information Is Created .......................................................................................... 61
How Devices Are Managed ......................................................................................................... 61
Device Naming Conventions ..................................................................................................... 62
Logical Disk Device Names ........................................................................................................ 62
Logical Tape Device Names ....................................................................................................... 65
Logical Removable Media Device Names ................................................................................. 65

4 Dynamically Configuring Devices (Tasks) ............................................................................ 67
Dynamic Reconfiguration and Hot-Plugging ............................................................................ 67
Attachment Points ....................................................................................................................... 69
Detaching PCI or PCIe Adapter Cards ...................................................................................... 71
Attaching PCI or PCIe Adapter Cards ....................................................................................... 71
PCIe Hot-Plugging With the (hotplug) Command ..................................................................... 71
SCSI Hot-Plugging With the cfgadm Command (Task Map) ................................................... 73
SCSI Hot-Plugging With the cfgadm Command ....................................................................... 74
▼ How to Display Information About SCSI Devices ............................................................... 74
▼ How to Unconfigure a SCSI Controller ............................................................................... 75
▼ How to Configure a SCSI Controller ................................................................................... 75
▼ How to Configure a SCSI Device ......................................................................................... 76
▼ How to Disconnect a SCSI Controller ............................................................................... 77
▼ SPARC: How to Connect a SCSI Controller ......................................................................... 78
▼ SPARC: How to Add a SCSI Device to a SCSI Bus ................................................................. 78
▼ SPARC: How to Replace an Identical Device on a SCSI Controller .................................... 79
▼ SPARC: How to Remove a SCSI Device ............................................................................... 81
Troubleshooting SCSI Configuration Problems ........................................................................ 82
▼ How to Resolve a Failed SCSI Unconfigure Operation ......................................................... 83
PCI or PCIe Hot-Plugging With the cfgadm Command (Task Map) .......................................... 83
PCI or PCIe Hot-Plugging With the cfgadm Command ............................................................... 84
PCIe LED Indicator Behavior ..................................................................................................... 84
▼ How to Display PCI Slot Configuration Information ............................................................ 85
▼ How to Remove a PCI Adapter Card ..................................................................................... 86
▼ How to Add a PCI Adapter Card .......................................................................................... 88
Troubleshooting PCI Configuration Problems .......................................................................... 89
SATA Hot-Plugging With the cfgadm Command ........................................................................ 90
▼ How to Unconfigure a SATA Device ................................................................................... 90
5 Managing USB Devices (Tasks) ................................................................. 101
What’s New in USB Devices? ................................................................. 101
   USB 3.0 Support ................................................................................. 101
About USB Support in Oracle Solaris .................................................... 102
   Supported USB Features ................................................................. 102
   USB Device Features and Compatibility Issues ............................... 103
   Bus-Powered Devices ..................................................................... 104
   USB Keyboards and Mouse Devices ................................................ 104
   USB Host Controller and Hubs ......................................................... 105
   USB Hub Devices ........................................................................... 106
   SPARC: USB Power Management .................................................. 106
   Guidelines for Connecting USB Cables ............................................ 107
Overview of USB Devices ................................................................. 107
   Commonly Used USB Acronyms ...................................................... 107
   Oracle Solaris USB Architecture .................................................... 108
   USB Bus Description ...................................................................... 109
   USB Devices and Driver Classes .................................................... 110
Managing USB Mass Storage Devices .................................................. 111
   Using USB Diskette Devices .......................................................... 112
   Hot-Plugging USB Mass Storage Devices ...................................... 113
Preparing to Use a USB Mass Storage Device ................................................................. 116
▼ How to Display USB Device Information ........................................................................ 116
▼ How to Create a File System on a USB Mass Storage Device ........................................ 117
▼ How to Modify Partitions and Create a PCFS File System on a USB Mass Storage Device ........................................................................................................ 119
▼ How to Create a Solaris Partition and Modify the Slices on a USB Mass Storage Device . 122
▼ How to Mount or Unmount a USB Mass Storage Device ............................................. 124
Troubleshooting Tips for USB Mass Storage Devices ...................................................... 126
Disabling Specific USB Drivers ...................................................................................... 126
▼ How to Disable Specific USB Drivers ........................................................................... 127
▼ How to Remove Unused USB Device Links ................................................................. 127
Using USB Audio Devices ............................................................................................. 127
Hot-Plugging Multiple USB Audio Devices ...................................................................... 128
▼ How to Add USB Audio Devices ................................................................................ 129
▼ How to Identify Your System’s Primary Audio Device ............................................... 129
▼ How to Change the Primary USB Audio Device ......................................................... 130
Troubleshooting USB Audio Device Problems .................................................................. 130
Hot-Plugging USB Devices With the cfgadm Command .................................................... 131
▼ How to Display USB Bus Information (cfgadm) .......................................................... 132
▼ How to Unconfigure a USB Device ............................................................................. 133
▼ How to Configure a USB Device ................................................................................. 134
▼ How to Logically Disconnect a USB Device .................................................................. 134
▼ How to Logically Connect a USB Device ..................................................................... 134
▼ How to Logically Disconnect a USB Device Subtree ................................................... 135
▼ How to Reset a USB Device ....................................................................................... 135
▼ How to Change the Default Configuration of a Multi-Configuration USB Device ........ 136

6 Using InfiniBand Devices (Overview/Tasks) ................................................................ 137
Overview of InfiniBand Devices ...................................................................................... 137
Dynamically Reconfiguring IB Devices (Task Map) .......................................................... 138
Dynamically Reconfiguring IB Devices (cfgadm) .............................................................. 140
▼ How to Display IB Device Information ...................................................................... 141
▼ How to Unconfigure an IOC Device ......................................................................... 142
▼ How to Configure an IOC Device .............................................................................. 143
▼ How to Unconfigure an IB Port, HCA_SVC, or a VPPA Device ............................... 143
Contents

- How to Configure a IB Port, HCA_SVC, or a VPPA Device ................................................. 144
- How to Unconfigure an IB Pseudo Device .............................................................................. 144
- How to Configure an IB Pseudo Device .................................................................................. 145
- How to Display Kernel IB Clients of an HCA ........................................................................ 145
- How to Unconfigure IB Devices Connected to an HCA ....................................................... 145
- Configuring an IB HCA ............................................................................................................ 146
- How to Update the IB p_key Tables ..................................................................................... 146
- How to Display IB Communication Services ........................................................................ 147
- How to Add a VPPA Communication Service ........................................................................ 147
- How to Remove an Existing IB Port, HCA_SVC, or a VPPA Communication Service .... 148
- How to Update an IOC Configuration .................................................................................... 148
- Using the uDAPL Application Interface With InfiniBand Devices ............................................. 149
- How to Enable uDAPL .............................................................................................................. 150
- Updating the DAT Static Registry ......................................................................................... 150

7 Managing Disks (Overview) ................................................................................................. 153
- What’s New in Disk Management? ....................................................................................... 153
- Advanced Format Disk Support ............................................................................................. 153
- Where to Find Disk Management Tasks ................................................................................ 154
- Overview of Disk Management .............................................................................................. 154
- Disk Terminology ................................................................................................................... 155
- About Disk Labels ................................................................................................................... 155
- EFI (GPT) Disk Label .............................................................................................................. 156
- About Disk Slices .................................................................................................................. 159
- format Utility ......................................................................................................................... 159
- Partitioning a Disk .................................................................................................................. 162
- Partition Table Terminology .................................................................................................. 163
- Displaying Partition Table Information ................................................................................ 163
- Using the Free Hog Slice ........................................................................................................ 165

8 Managing Disk Use (Tasks) .................................................................................................. 167
- Managing Disk Use (Task Map) .............................................................................................. 167
- Displaying Information About Files and Disk Space ............................................................... 168
- How to Display Information About Files and Disk Space .................................................... 169
- Checking the Size of Files ....................................................................................................... 171
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>▼ How to Display the Size of Files</td>
<td>171</td>
</tr>
<tr>
<td>▼ How to Find Large Files</td>
<td>172</td>
</tr>
<tr>
<td>▼ How to Find Files That Exceed a Specified Size Limit</td>
<td>173</td>
</tr>
<tr>
<td>Checking the Size of Directories</td>
<td>174</td>
</tr>
<tr>
<td>▼ How to Display the Size of Directories, Subdirectories, and Files</td>
<td>174</td>
</tr>
<tr>
<td>▼ How to Display the User Ownership of Local UFS File Systems</td>
<td>175</td>
</tr>
<tr>
<td>Finding and Removing Old or Inactive Files</td>
<td>176</td>
</tr>
<tr>
<td>▼ How to List the Newest Files</td>
<td>177</td>
</tr>
<tr>
<td>▼ How to Find and Remove Old or Inactive Files</td>
<td>177</td>
</tr>
<tr>
<td>▼ How to Clear Out Temporary Directories</td>
<td>178</td>
</tr>
<tr>
<td>▼ How to Find and Delete core Files</td>
<td>179</td>
</tr>
<tr>
<td>▼ How to Delete Crash Dump Files</td>
<td>180</td>
</tr>
<tr>
<td>▼ How to Identify the Disks on a System</td>
<td>182</td>
</tr>
<tr>
<td>Formatting a Disk</td>
<td>184</td>
</tr>
<tr>
<td>▼ How to Determine if a Disk Is Formatted</td>
<td>184</td>
</tr>
<tr>
<td>▼ How to Format a Disk</td>
<td>185</td>
</tr>
<tr>
<td>Displaying Disk Slices</td>
<td>186</td>
</tr>
<tr>
<td>▼ How to Display Disk Slice Information</td>
<td>187</td>
</tr>
<tr>
<td>Creating and Examining a Disk Label</td>
<td>189</td>
</tr>
<tr>
<td>▼ How to Label a Disk</td>
<td>189</td>
</tr>
<tr>
<td>▼ How to Examine a Disk Label</td>
<td>193</td>
</tr>
<tr>
<td>Recovering a Corrupted Disk Label</td>
<td>195</td>
</tr>
<tr>
<td>▼ How to Recover a Corrupted Disk Label</td>
<td>195</td>
</tr>
<tr>
<td>Adding a Third-Party Disk</td>
<td>197</td>
</tr>
</tbody>
</table>

9 Administering Disks (Tasks) .............................................................................................................181

Administering Disks (Task Map) .................................................................................................181
Identifying Disks on a System .......................................................................................................182
▼ How to Identify the Disks on a System .....................................................................................182
Formatting a Disk ..............................................................................................................................184
▼ How to Determine if a Disk Is Formatted ..................................................................................184
▼ How to Format a Disk .................................................................................................................185
Displaying Disk Slices .......................................................................................................................186
▼ How to Display Disk Slice Information ...................................................................................187
Creating and Examining a Disk Label ..............................................................................................189
▼ How to Label a Disk ...................................................................................................................189
▼ How to Examine a Disk Label ....................................................................................................193
Recovering a Corrupted Disk Label ..............................................................................................195
▼ How to Recover a Corrupted Disk Label ..................................................................................195
Adding a Third-Party Disk ...............................................................................................................197

10 SPARC: Setting Up Disks (Tasks) ......................................................................................................199

SPARC: Setting Up Disks for UFS File Systems (Task Map) ........................................................199
SPARC: Setting Up Disks for UFS File Systems ............................................................................200
▼ SPARC: How to Set Up a Disk for a UFS Root File System ..........................................................200
▼ SPARC: How to Set Up a Disk for a UFS File System ..................................................................201
▼ SPARC: How to Create Disk Slices for a UFS File System .........................................................202
How to Add Swap Space in an Oracle Solaris ZFS Root Environment .................................. 308

17 Checking UFS FileSystem Consistency (Tasks) ............................................................................. 311

UFS FileSystem Consistency ............................................................................................................. 312
  Enhancements to UFS FileSystem Utilities (fsck, mkfs, and newfs) ............................................. 312
How the UFS FileSystem State Is Recorded .......................................................................................... 318
What the fsck Command Checks and tries to Repair ........................................................................ 319
  Why UFS FileSystem Inconsistencies Might Occur ..................................................................... 319
  UFS Components That Are Checked for Consistency ..................................................................... 320
  fsck Summary Message .................................................................................................................. 325
Interactively Checking and Repairing a UFS FileSystem .................................................................... 325

▼ How to Check the UFS root (/), /usr, or /var File Systems From an Alternate Boot Device ......... 326

▼ How to Check Other UFS File Systems (Not root (/), /usr, or /var) ............................................ 328
  Preening UFS File Systems .......................................................................................................... 330
▼ How to Preen a UFS File System ................................................................................................. 330
  Fixing a UFS File System That the fsck Command Cannot Repair ............................................. 331
Restoring a Bad UFS Superblock ........................................................................................................ 332
▼ How to Restore a Bad UFS Superblock (Solaris 10 6/06 Release) ............................................. 332
▼ How to Restore a Bad Superblock (Solaris 8, 9, and 10 Releases) ............................................. 335
Syntax and Options for the fsck Command ......................................................................................... 337

18 UFS FileSystem (Reference) ............................................................................................................ 339

Structure of Cylinder Groups for UFS File Systems ......................................................................... 339
  Boot Block .................................................................................................................................... 340
  Superblock .................................................................................................................................... 340
  Inodes .......................................................................................................................................... 340
  Data Blocks .................................................................................................................................. 341
  Free Blocks .................................................................................................................................. 342
Customizing UFS FileSystem Parameters ......................................................................................... 342
  Logical Block Size ....................................................................................................................... 343
  Fragment Size .............................................................................................................................. 343
  Minimum Free Space .................................................................................................................. 344
  Rotational Delay .......................................................................................................................... 344
  Optimization Type ...................................................................................................................... 344
19 Backing Up and Restoring UFS File Systems (Overview/Tasks) ................................................. 347
   Introduction to Backing Up and Restoring UFS File Systems ................................................. 347
   Why You Should Back Up File Systems .................................................................................. 348
   Planning Which UFS File Systems to Back Up ................................................................. 348
   Choosing the Type of Backup ................................................................................................. 350
   Choosing a Tape Device ............................................................................................................ 351
   High-Level View of Backing Up and Restoring UFS File Systems (Task Map) ................. 351
      Considerations for Scheduling Backups ............................................................................. 352
      Guidelines for Scheduling Backups .................................................................................... 353
      Using Dump Levels to Create Incremental Backups ..................................................... 354
      Sample Backup Schedules ............................................................................................... 356
   Backing Up UFS Files and File System (Task Map) .............................................................. 362
   Preparing for File System Backups ....................................................................................... 362
   ▼ How to Find UFS File System Names ............................................................................... 362
   ▼ How to Determine the Number of Tapes Needed for a Full Backup .............................. 363
   Backing Up a UFS File System .............................................................................................. 364
   ▼ How to Back Up a UFS File System to Tape .................................................................. 364
   Restoring UFS Files and File System Backups (Task Map) .................................................... 369
   Preparing to Restore UFS Files and File Systems ............................................................... 370
      Determining the UFS File System Name ......................................................................... 370
      Determining the Type of Tape Device You Need ........................................................... 370
      Determining the Tape Device Name ................................................................................. 371
   Restoring UFS Files and File Systems .................................................................................. 371
   ▼ How to Determine Which Tapes to Use ......................................................................... 371
   ▼ How to Restore UFS Files Interactively ......................................................................... 372
   ▼ How to Restore Specific UFS Files Noninteractively ..................................................... 375
   ▼ How to Restore a Complete UFS File System ................................................................. 376
   ▼ How to Restore a UFS root (/) and /usr File System ...................................................... 378

20 Using UFS Snapshots (Tasks) ............................................................................................... 383
   Using UFS Snapshots (Task Map) ......................................................................................... 383
UFS Snapshots Overview...........................................................................................................384
   Why Use UFS Snapshots? .................................................................................................384
   UFS Snapshots Performance Issues ................................................................................385
Creating and Deleting UFS Snapshots ..................................................................................385
   Creating a Multiterabyte UFS Snapshot .............................................................................386
   ▼ How to Create a UFS Snapshot .......................................................................................387
   ▼ How to Display UFS Snapshot Information ....................................................................388
   Deleting a UFS Snapshot ......................................................................................................389
   ▼ How to Delete a UFS Snapshot .......................................................................................389
Backing Up a UFS Snapshot ....................................................................................................390
   ▼ How to Create a Full Backup of a UFS Snapshot (ufsdump) ............................................390
   ▼ How to Create an Incremental Backup of a UFS Snapshot (ufsdump) ..............................391
   ▼ How to Back Up a UFS Snapshot (tar) ...........................................................................392
Restoring Data From a UFS Snapshot Backup ........................................................................392

21 Copying Files and File Systems (Tasks) ......................................................................393
   Commands for Copying File Systems ...............................................................................393
   Copying Directories Between File Systems (cpio Command) ............................................395
      ▼ How to Copy Directories Between File Systems (cpio) ..................................................395
   Copying Files and File Systems to Tape ............................................................................396
   Copying Files to Tape (tar Command) ...............................................................................396
      ▼ How to Copy Files to a Tape (tar) ................................................................................397
      ▼ How to List the Files on a Tape (tar) ..........................................................................398
      ▼ How to Retrieve Files From a Tape (tar) ....................................................................398
   Copying Files to a Tape With the pax Command ...............................................................399
      ▼ How to Copy Files to a Tape (pax) .................................................................................399
   Copying Files to Tape With the cpio Command .................................................................400
      ▼ How to Copy All Files in a Directory to a Tape (cpio) ....................................................400
      ▼ How to List the Files on a Tape (cpio) .........................................................................401
      ▼ How to Retrieve All Files From a Tape (cpio) ..............................................................402
      ▼ How to Retrieve Specific Files From a Tape (cpio) .......................................................403
   Copying Files to a Remote Tape Device ............................................................................404
      ▼ How to Copy Files to a Remote Tape Device (tar and dd) ............................................404
      ▼ How to Extract Files From a Remote Tape Device ........................................................405
Preface

System Administration: Devices and File Systems is part of a set that includes a significant part of the Oracle Solaris system administration information. This guide contains information for both SPARC based and x86 based systems.

This book assumes you have completed the following tasks:

- Installed the Oracle Solaris software
- Set up all the networking software that you plan to use

New Oracle Solaris features of interest to system administrators are covered in sections called What’s New in ... ? in the appropriate chapters.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the Oracle Solaris Hardware Compatibility List at http://www.oracle.com/webfolder/technetwork/hcl/index.html. This document cites any implementation differences between the platform types.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems running the 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.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.
What Typographic Conventions Mean

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

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

Shell Prompts in Command Examples

The following table shows the default system prompt and root (or administrative) prompt for the C shell, Bourne shell, and Korn shell.

<table>
<thead>
<tr>
<th>Shell</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>C shell prompt</td>
<td>machine_name%</td>
</tr>
<tr>
<td>C shell root prompt</td>
<td>machine_name#</td>
</tr>
<tr>
<td>Bourne shell and Korn shell prompt</td>
<td>$</td>
</tr>
<tr>
<td>Bourne shell and Korn shell root prompt</td>
<td>#</td>
</tr>
</tbody>
</table>
Managing Removable Media (Overview/Tasks)

This chapter describes how to manage and access removable media in the Oracle Solaris OS.

This is a list of the information in this chapter:

- “Managing Removable Media (Overview)” on page 19
- “Managing Removable Media (Tasks)” on page 23
- “Accessing Removable Media” on page 30
- “Accessing Removable Media on a Remote System” on page 33

Managing Removable Media (Overview)

The following information is described in this section:

- “Removable Media Features and Benefits” on page 19
- “Comparison of Manual and Automatic Mounting” on page 21
- “Overview of Accessing Removable Media” on page 22

Removable Media Features and Benefits

The Oracle Solaris release gives users and software developers a standard interface for dealing with removable media. Removable media services provide the following benefits:

- Automatically mounts removable media. For a comparison of manual and automatic mounting, see the following section.
- Enables you to access removable media without having to become an administrator.
- Allows you to give other systems on the network automatic access to any removable media on your local system. For more information, see “Accessing Removable Media on a Remote System” on page 33.
vold is Managed by the Service Management Facility (SMF)

The volume management daemon, vold, is managed by the Service Management Facility (SMF). This means you can use the svcadm disable command to disable the following new voldfs service, if appropriate:

```
# svcadm disable volfs
```

You can identify the status of the volfs service by using this command:

```
$ svcs volfs
```

<table>
<thead>
<tr>
<th>STATE</th>
<th>STIME</th>
<th>FMRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>online</td>
<td>Sep_29</td>
<td>svc:/system/filesystem/volfs:default</td>
</tr>
</tbody>
</table>

For more information, see smf(5).

You can use the svccfg command to display and to set additional vold properties. For example, you could temporarily enable vold logging to help troubleshooting a problem. For example:

```
# svccfg
svc:/ select system/filesystem/volfs
svc:/system/filesystem/volfs> setprop vold/log_debuglevel=3
svc:/system/filesystem/volfs> exit
# svcadm disable volfs
# svcadm enable volfs
```

You can also use the svccfg command to display a listing of settable vold properties.

```
# svccfg
svc:/ select volfs
svc:/system/filesystem/volfs> listprop vold/*
vold/config_file astring
vold/log_debuglevel count 3
vold/log_file astring
vold/log_nfs_trace boolean false
vold/log_verbbose boolean false
vold/root_dir astring
vold/never_writeback_label boolean false
svc:/system/filesystem/volfs> exit
```

For a description of these properties, see the vold(1M).

Volume Management (vold) Improvements

Device links are properly created for devices that contain no media, similar to the following:

```
lrwxrwxrwx 1 root root 28 Jun 13 13:09 /vol/dev/aliases/cdrom0 -> /vol/dev/rdsk/c2t2d0/nomedia
```

You can use the cdrw and rmformat commands to list devices that have no media when vold is running.

You can revert back to the previous vold behavior by changing the following support nomedia entry in the /etc/vold.conf file as follows:
support media

Then, restart `vold`. For more information, see `vold.conf(4)`.

In addition, `vold` is now hot-plug aware. This improvement means that if you insert removable media, the media is automatically detected and mounted by `vold`. There should be no need to restart `vold` manually to recognize and mount a file system from any removable media device.

If you are using a legacy or non-USB diskette device, then you might need to issue the `volcheck` command before `vold` can recognize the media.

If the media is detected, but for some reason, is unmounted, then you’ll need to run the following commands:

```
# volrmount -i rmdisk0
```

Before you hot-remove a removable media device, eject the media first. For example:

```
# eject rmdisk0
```

## Comparison of Manual and Automatic Mounting

The following table compares the steps involved in manual mounting (without removable media services) and automatic mounting (with removable media management) of removable media.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Manual Mounting</th>
<th>Automatic Mounting</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Insert media.</td>
<td>Insert media.</td>
</tr>
<tr>
<td>2</td>
<td>Become superuser.</td>
<td>For USB diskettes, use the <code>volcheck</code> command.</td>
</tr>
<tr>
<td>3</td>
<td>Determine the location of the media device.</td>
<td>Removable media services automatically perform many of the tasks that are required to manually mount and work with removable media.</td>
</tr>
<tr>
<td>4</td>
<td>Create a mount point.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Make sure you are not in the mount point directory.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Mount the device and use the proper mount options.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Exit the superuser account.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Work with files on media.</td>
<td>Work with files on media.</td>
</tr>
</tbody>
</table>
### TABLE 1–1  Comparison of Manual and Automatic Mounting of Removable Media (Continued)

<table>
<thead>
<tr>
<th>Steps</th>
<th>Manual Mounting</th>
<th>Automatic Mounting</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Become superuser.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Unmount the media device.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Eject media.</td>
<td>Eject media.</td>
</tr>
<tr>
<td>12</td>
<td>Exit the superuser account.</td>
<td></td>
</tr>
</tbody>
</table>

### Overview of Accessing Removable Media

Essentially, removable media services enable you to access removable media just as manual mounting does, but more easily and without the need for administrative access.

If your system has more than one type of removable device, see the following table for their access points.

### TABLE 1–2  Where to Access Removable Media

<table>
<thead>
<tr>
<th>Access FileSystem With This Path</th>
<th>Access Raw Data With This Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>/floppy/floppy0</td>
<td>/vol/dev/aliases/floppy0</td>
</tr>
<tr>
<td>/floppy/floppy1</td>
<td>/vol/dev/aliases/floppy1</td>
</tr>
<tr>
<td>/cdrom/cdrom0</td>
<td>/vol/dev/aliases/cdrom0</td>
</tr>
<tr>
<td>/cdrom/cdrom1</td>
<td>/vol/dev/aliases/cdrom1</td>
</tr>
<tr>
<td>/rmdisk/rmdisk0 or /rmdisk1</td>
<td>/vol/dev/aliases/rmdisk0 or /vol/dev/aliases/rmdisk1</td>
</tr>
<tr>
<td>/pcmem/pcmem0</td>
<td>/vol/dev/aliases/pcmem0</td>
</tr>
</tbody>
</table>

You can use the `rmmount -l` command to identify mounted media on your system. For example:

```bash
# rmmount -l
/dev/dsk/c5t0d0p0  rmdisk6,/media/FD-05PUB
/dev/dsk/c4t0d3p0  rmdisk5,/media/223UHS-SD-MMC
/dev/dsk/c2t0d0s2  cdrom1,cd1,sr1,Oracle_Solaris-11-1-AI-SPARC,/media/Oracle_Solaris-11-1-AI-SPARC
/dev/dsk/c3t0d0p0  rmdisk2,/media/003B-00CRA0
```

In the above output, the mounted devices are as follows:

- `/dev/dsk/c5t0d0p0`  USB diskette
- `/dev/dsk/c4t0d3p0`  CF card in a USB card reader
- `/dev/dsk/c2t0d0s2`  DVD-ROM
Managing Removable Media (Tasks)

The following guidelines and procedures are described in this section:

- “Removable Media Considerations” on page 23
- “How to Load Removable Media” on page 24
- “How to Format a Diskette (rmformat)” on page 25
- “How to Create a File System on Removable Media” on page 26
- “How to Create a File System on a DVD-RAM” on page 27
- “How to Check a File System on Removable Media” on page 27
- “How to Repair Bad Blocks on Removable Media” on page 28
- “Applying Read or Write Protection and Password Protection to Removable Media” on page 28
- “How to Enable or Disable Write Protection on Removable Media” on page 29
- “How to Enable or Disable Read or Write Protection and Set a Password on Removable Media” on page 29

Removable Media Considerations

Keep the following considerations in mind when working with diskettes:

- File system formats in Oracle Solaris consist of the basic "bit" formatting, in addition to the structure to support an Oracle Solaris file system. A complete format for a DOS file system consists of the basic "bit" formatting in addition the structure to support either an MS-DOS or an NEC-DOS file system. The procedures required to prepare a media for each type of file system are different. Therefore, before you format a diskette, consider which procedure to follow. For more information, see “Managing Removable Media (Tasks)” on page 23.
  - For information on removable media names, see “Using Removable Media Names” on page 30.
  - Diskettes that are not named (that is, they have no “label”) are assigned the default name of unnamed_floppy.
  - Diskettes that are not named (that is, they have no “label”) are assigned the default name of floppy.

An Oracle Solaris system can format the following file system types:

- ZFS or UFS
- MS-DOS or NEC-DOS (PCFS)
- UDFS
On an Oracle Solaris system (either SPARC or x86), you can format diskettes with the following densities.

<table>
<thead>
<tr>
<th>Diskette Size</th>
<th>Diskette Density</th>
<th>Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5&quot;</td>
<td>High density (HD)</td>
<td>1.44 MB</td>
</tr>
<tr>
<td>3.5&quot;</td>
<td>Double density (DD)</td>
<td>720 KB</td>
</tr>
</tbody>
</table>

By default, the diskette drive formats a diskette to a like density. This default means that a 1.44 MB drive attempts to format a diskette for 1.44 MB, regardless of whether the diskette is, in fact, a 1.44 MB diskette, 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.

### Formatting Diskettes

You can use the `rmformat` command to format and perform other management tasks on removable media. File systems are mounted automatically. So, you might have to unmount media before you can format it, if the media contains an existing file system.

The `rmformat` command has three formatting options:

- **quick** – This option formats diskettes without certification or with limited certification of certain tracks on the media.
- **long** – This option completely formats diskettes. For some devices, the use of this option might include the certification of the whole media by the drive.
- **force** – This option formats completely without user confirmation. For diskettes with a password-protection mechanism, this option clears the password before formatting. This feature is useful when a password is forgotten. On diskettes without password protection, this option forces a long format.

### How to Load Removable Media

For information about removable media hardware considerations, see “Removable Media Considerations” on page 23.

1. **Insert the media.**

2. **Ensure that the media is formatted.**

   If you aren’t sure, insert the media and check the status messages in the system console window, as described in Step 3. If you need to format the media, go to “How to Format a Diskette (`rmformat`)” on page 25.
3 (Optional) Notify volume management if you are using a legacy, non-USB diskette device.

$ volcheck -v

Two status messages are possible:

**media was found**

Volume management detected the media and will attempt to mount it in the directory described in “Using Removable Media Names” on page 30.

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. However, error messages similar to the following appear in the system console window:

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 “How to Format a Diskette (rmformat)” on page 25.

**no media was found**

Volume management did not detect the media. Ensure that the media is inserted properly, and run volcheck again. If unsuccessful, check the media, which 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 /media/floppy

lost+found myfiles

▼ How to Format a Diskette (rmformat)

You can use the rmformat command to format a diskette. By default, this command creates two partitions on the media: partition 0 and partition 2 (the whole media).

1 Verify that removable media service is running. If so, you can use the shorter nickname for the device name.

# svcs hal dbus rmvolmgr

<table>
<thead>
<tr>
<th>STATE</th>
<th>STIME</th>
<th>FMRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>online</td>
<td>Apr 09</td>
<td>svc:/system/dbus:default</td>
</tr>
</tbody>
</table>
For information on restarting removable media services, see “How to Disable or Enable Removable Media Services” on page 31. For information on identifying media device names, see “Using Removable Media Names” on page 30.

2 Format the diskette.

```bash
$ rmformat -F [ quick | long | force ] device-name
```

See “Formatting Diskettes” on page 24 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 28.

3 (Optional) Label the diskette with an 8-character label.

```bash
$ rmformat -b label device-name
```

For information on creating a DOS label, see `mkfs_pcfs(1M)`.

---

### Example 1–1 Formatting a Diskette

This example shows how to format a diskette.

```bash
$ rmformat -F quick /dev/rdiskette
Formatting will erase all the data on disk.
Do you want to continue? (y/n) y
```

---

#### How to Create a File System on Removable Media

1 (Optional) Format a diskette, if necessary.

To format a USB diskette, use syntax similar to the following:

```bash
$ rmformat -F long /dev/rdsk/c11t0d0p0
```

2 (Optional) Create an alternate Solaris partition table.

```bash
$ rmformat -s slice-file device-name
```

A sample slice file appears similar to the following:

```bash
slices: 0 = 0, 30MB, "wm", "home" :
  1 = 30MB, 51MB :
    2 = 0, 94MB, "wm", "backup" :
    6 = 81MB, 13MB
```

3 Become superuser.
Determine the appropriate file system type and select one of the following:

- Create a PCFS file system. For example:
  
  ```bash
  # mkfs -F pcfs -o nofdisk,size=9800 /dev/rdsk/c1t0d0p0
  ```

- Create a UDFS file system. For example:
  
  ```bash
  # mkfs -F udfs /dev/rdsk/c0t1d0p0
  ```

▼ How to Create a File System on a DVD-RAM

Use this procedure to create a file system on a DVD-RAM.

1 Become superuser.

2 Create a file system on the DVD-RAM device.
   For example, create a UDFS file system, as follows:
   
   ```bash
   # mkfs -F udfs /dev/rdsk/c0t0d0s2
   ```

3 Mount the file system.
   For example, mount a UDFS file system, as follows:
   
   ```bash
   # mount -F udfs /dev/dsk/c0t0d0s2 /mnt
   ```

4 Verify that you can read or write to the file system.

5 When finished, eject the DVD-RAM.

▼ How to Check a File System on Removable Media

1 Become superuser.

2 Identify the file system type and select one of the following:
   
   - Check a UDFS file system, as follows:
     
     ```bash
     # fsck -F udfs device-name
     ```

   - Check a PCFS file system, as follows:
     
     ```bash
     # fsck -F pcfs device-name
     ```

Example 1–2 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 USB memory sticks 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 the failure to repair the media.

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 Protection and Password Protection to Removable Media

You can apply read protection or write protection, and set a password, on removable media that support this feature.
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:

- Enable write protection.
  
  ```
  rmformat -w enable device-name
  ```

- 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 Set a Password on Removable Media

You can apply a password with a maximum of 32 characters for removable media that support this feature.

You will 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 set a password. Select one of the following:

- 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:

- 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
Verify whether the media’s read protection or write protection is enabled or disabled.

$ rmformat -p device-name

Accessing Removable Media

You can access information on removable media with or without using volume management. For information on accessing information on removable media with GNOME’s File Manager, see the GNOME desktop documentation.

This following procedures are described in this section:

- “Using Removable Media Names” on page 30
- “Guidelines for Accessing Removable Media Data” on page 31
- “How to Add a New Removable Media Drive” on page 31
- “How to Disable or Enable Removable Media Services” on page 31
- “How to Access Information on Removable Media” on page 32
- “How to Determine If Removable Media Is Still in Use” on page 32
- “How to Eject Removable Media” on page 33

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>
<thead>
<tr>
<th>Media</th>
<th>Volume Management Device Name</th>
<th>Volume Management Device Alias Name</th>
<th>Device Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>First diskette drive</td>
<td>/floppy</td>
<td>/vol/dev/aliases/floppy0</td>
<td>/dev/rdiskette</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>/vol/dev/rdiskette0/</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>volume-name</td>
</tr>
<tr>
<td>First, second, third CD-ROM or</td>
<td>/cdrom0</td>
<td>/vol/dev/aliases/cdrom0</td>
<td>/vol/dev/rdsk/cntn[dn]/</td>
</tr>
<tr>
<td>DVD-ROM drives</td>
<td>/cdrom1</td>
<td>/vol/dev/aliases/cdrom1</td>
<td>volume-name</td>
</tr>
<tr>
<td></td>
<td>/cdrom2</td>
<td>/vol/dev/aliases/cdrom2</td>
<td>volume-name</td>
</tr>
<tr>
<td>USB memory stick</td>
<td>/rmdisk/hostname</td>
<td>/vol/dev/aliases/rmdisk0</td>
<td>/vol/dev/dsk/cntndn/volume-name:c</td>
</tr>
</tbody>
</table>
Guidelines for Accessing Removable Media Data

Most DVDs are formatted to the ISO 9660 standard, which is portable. So, DVDs can be mounted by volume management.

To accommodate possible different formats, a DVD is split into slices. Slices are similar in effect to partitions on hard disks. The 9660 portion is portable. If you are having trouble mounting a DVD, particularly if it is an installation DVD, make sure that its file system is appropriate for your system’s architecture. For example, you can check the label on DVD.

How to Add a New Removable Media Drive

Generally, most modern bus types support hot-plugging. This means you can insert a disk in an empty slot and the system recognizes it.

For more information about hot-plugging devices, see Chapter 4, “Dynamically Configuring Devices (Tasks).”

1 Become superuser.

2 Connect the new media drive.
   See your hardware handbook for specific instructions.

3 Confirm that the system sees the new media drive.
   # rmformat
   Looking for devices...

How to Disable or Enable Removable Media Services

Occasionally, you might want to manage media without using removable media services. This section describes how to disable and enable removable media services.

Disabling these services means that you would have to mount all media manually by using the mount command.

1 Ensure that the media is not being used.
   If you are not sure whether you have found all users of the media, use the fuser command, see “How to Determine If Removable Media Is Still in Use” on page 32.

2 Become superuser.

3 Select one of the following:
   - Disable removable media services.
### How to Access Information on Removable Media

1. **Insert the media.**
   
The media is mounted after a few seconds.

2. **List the contents of the media. For example:**
   
   ```
   # ls /media
   ```

3. **(Optional) Copy the file that you identified in the previous step.**

   **Example 1–3**

   **Accessing Information on Removable Media**

   This example shows how to access information on a USB memory stick.

   ```
   $ ls /rmdisk
   rmdisk0/    rmdisk1/
   ```

   This example shows how to access information on a DVD.

   ```
   $ ls /cdrom
   cdrom0     sol_10_811_sparc
   ```

### How to Determine If Removable Media Is Still in Use

1. **Become superuser.**

2. **Identify the processes that are accessing the media.**

   ```
   # fuser -u /media
   ```

   The `-u` displays the user of the media.

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

3. **(Optional) Kill the process accessing the media.**

   ```
   # fuser -u -k /media
   ```

   The `-k` option kills the processes accessing the media.
Caution – Killing the processes that are accessing the media should only be used in emergency situations.

4 Verify that the process is gone.
   # pgrep process-ID

▼ How to Eject Removable Media

1 Ensure that 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 DVD (for example, a shell hidden behind a desktop tool might be accessing it), use the fuser command. See “How to Determine If Removable Media Is Still in Use” on page 32.

2 Eject the media.
   # eject media
   For example, for a DVD, you would do the following:
   
   # eject cdrom
   For example, for a USB memory stick, you would do the following:
   
   # eject rmdisk0

Tip – You can view the removable device name with the eject -I command.

Accessing Removable Media on a Remote System

Removable media can be shared with remote systems by using NFS. For additional information about using NFS, see Oracle System Administration Guide: Network Services.

The following procedures are described in this section:

- “How to Make Local Media Available to Other Systems” on page 33
- “How to Access Removable Media on Remote Systems” on page 35

▼ 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. One exception is musical CDs. Once your media drives are shared,
other systems can access the media they contain simply by mounting them. For instructions, see “How to Access Removable Media on Remote Systems” on page 35.

1  Become superuser.

2  Confirm that the media is loaded.

3  Add the following entry to the /etc/dfs/dfstab file.
   For example:
   
   share -F nfs -o ro /cdrom/sol_10_811_sparc

4  Determine whether the NFS server service is running.
   # svcs *nfs*
   
   The following output is returned from the svcs command if NFS server service is running:
   
   online 14:28:43 svc:/network/nfs/server:default

5  Identify the NFS server status, and select one of the following:
   
   - If the NFS server service is running, go to Step 8.
   - If the NFS server service is not running, go to the next step.

6  Start the NFS server service.
   
   # svcadm enable network/nfs/server

7  Verify that the NFS daemons are running.
   
   For example:
   
   # svcs -p svc:/network/nfs/server:default
   
   STATE   STIME    FMRI
   online   Aug_30   svc:/network/nfs/server:default
           Aug_30   319 mountd
           Aug_30   323 nford

8  Verify that the media is available to other systems.
   
   If the media is available, its share configuration is displayed.
   
   # share
   - /cdrom/sol_10_811_sparc sec=sys,ro ""

Example 1–4  Making Local DVDs Available to Other Systems

The following example shows how to make any local DVD available to other systems on the network.
Accessing Removable Media on Remote Systems

You can access media on a remote system by manually mounting the media into your file system. Also, the remote system must have shared its media according to the instructions in “How to Make Local Media Available to Other Systems” on page 33.

1. **Select an existing directory to serve as the mount point. Or create a mount point.**
   ```shell
   $ mkdir /directory
   ```
   where `/directory` is the name of the directory that you create to serve as a mount point for the remote system’s DVD.

2. **Find the name of the media that you want to mount.**
   ```shell
   $ showmount -e system-name
   ```

3. **As an administrator, mount the media.**
   ```shell
   # mount -F nfs -o ro system-name:/media/media-name local-mount-point
   ```
   - **system-name:** Is the name of the system whose media you will mount.
   - **media-name:** Is the name of the media you want to mount.
   - **local-mount-point** Is the local directory onto which you will mount the remote media.

4. **Log out as an administrator.**

5. **Verify that the media has been mounted.**
   ```shell
   $ ls /mnt
   ```

**Example 1–5 Accessing DVDs or CDs on Remote Systems**

The following example shows how to automatically access the remote DVD named `sol_10_811_sparc` from the remote system `starbug`.

```shell
$ showmount -e starbug
export list for starbug:
/cdrom/sol_10_811_sparc (everyone)
$ ls /net/starbug/cdrom/
cdrom/sol_10_811_sparc
```
Chapter 2

Writing CDs and DVDs (Tasks)

This chapter provides step-by-step instructions for writing and copying data CDs and DVDs and audio CDs with the cd rw command.

This is a list of the information in this chapter:

- “Working With Audio CDs and Data CDs and DVDs” on page 37
- “Writing CD and DVD Data and Audio CDs” on page 39

Working With Audio CDs and Data CDs and DVDs

You can use the cd rw command to write file systems for CDs and DVDs in ISO 9660 format with Rock Ridge or Joliet extensions on CD-R, CD-RW, DVD-RW, or DVD+RW media devices.

You can use the cd rw command to perform the following tasks:

- Create data CDs and DVDs.
- Create audio CDs.
- Extract audio data from an audio CD.
- Copy CDs and DVDs.
- Erase CD-RW media.

The cd rw command is available in the following releases:

- Oracle Solaris 10 releases (SUNWcd rw)
- Oracle Solaris 11 release, media/cd rw package

# CD/DVD Media Commonly Used Terms

This section defines commonly used terms related to CD/DVD media.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD-R</td>
<td>CD read media that can be written once and after that, can only be read from.</td>
</tr>
<tr>
<td>CD-RW</td>
<td>CD re-writable media that can be written to and erased. CD-RW media can only be read by CD-RW devices.</td>
</tr>
<tr>
<td>DVD-R</td>
<td>Digital video disk (recordable) that can be written once and after that, can only be read from. These devices have much larger capacity than CD-R media.</td>
</tr>
<tr>
<td>DVD+R</td>
<td>Digital video disk (recordable) that can be written once and after that, can only be read from. DVD+R devices have more complete error management system than DVD-R, which allows for more accurate burning to media, independent of the quality of the media.</td>
</tr>
<tr>
<td>DVD-RW</td>
<td>Digital video disk (rewritable) with storage capacity equal to a DVD-R. This media can be re-recorded by first erasing the entire disk.</td>
</tr>
<tr>
<td>DVD+RW</td>
<td>Digital video disk (random-access re-writable) with storage capacity equal to a DVD+R. This medium allows overwriting of individual blocks without erasing the entire disk.</td>
</tr>
<tr>
<td>DVD-RAM</td>
<td>Digital video disk (random access memory, re-writable) with circular rather than spiral tracks and hard sectoring.</td>
</tr>
<tr>
<td>ISO 9660</td>
<td>ISO, an acronym for Industry Standards Organization, is an organization that sets standards for computer storage formats. An ISO 9660 file system is a standard CD or DVD file system that enables you to read the same CD or DVD 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 or DVD drives can read files from an ISO 9660 file system.</td>
</tr>
<tr>
<td>Joliet extensions</td>
<td>Adds Windows file system information.</td>
</tr>
<tr>
<td>Rock Ridge extensions</td>
<td>Adds UNIX file system information. (Rock Ridge is named after the town in the movie Blazing Saddles.)</td>
</tr>
</tbody>
</table>

**Note** - These extensions are not mutually exclusive. You can specify both `mkisofs -R` and `-j` options for compatibility with both systems. (See `mkisofs` for details.)
Term | Description
--- | ---
MMC-compliant recorder | Acronym for Multi Media Command, which means these recorders comply with a common command set. Programs that can write to one MMC-compliant recorder should be able to write to all other recorders.
Red Book CDDA | Acronym for Compact Disc Digital Audio, which is an industry standard method for storing digital audio on compact discs. 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 writing to CD media are listed in the following table.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blanking</td>
<td>The process of erasing data from the CD-RW media.</td>
</tr>
<tr>
<td>session</td>
<td>A complete track with lead-in and lead-out information.</td>
</tr>
<tr>
<td>track</td>
<td>A complete data or audio unit.</td>
</tr>
</tbody>
</table>

**Writing CD and DVD Data and Audio CDs**

The following procedures are described in this section:

- “Restricting User Access to Removable Media With RBAC” on page 40
- “How to Restrict User Access to Removable Media With RBAC” on page 41
- “How to Identify a CD or DVD Writer” on page 41
- “How to Check the CD or DVD Media” on page 42
- "Creating a Data CD or DVD” on page 42
- “How to Create an ISO 9660 File System for a Data CD or DVD” on page 42
- ”How to Create a Multi-Session Data CD” on page 43
- ”Creating an Audio CD” on page 45
- “How to Create an Audio CD” on page 45
- “How to Extract an Audio Track on a CD” on page 46
- “How to Erase CD-RW Media” on page 48

The process of writing to a CD or DVD cannot be interrupted and needs a constant stream of data. Consider using the `cdrw -s` option to simulate writing to the media to verify that the system can provide data at a sufficient rate for writing to the CD or DVD.
Write errors can be caused by one of the following problems:

- 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 are starving the writing process.
- The image is on a remote system, and network congestion is causing delays in reading the image.
- The source drive is slower than the destination drive.

If any of these problems occur, you can lower the writing speed of the device by using the `cd`r`w -p` option.

For example, the following command shows how to simulate writing at 4x speed:

```
$ cdrw -i5 -p 4 image.iso
```

**Note** – CD-R, CD-RW (not MRW formatted), DVD-R, and DVD-RW media support simulation mode (-5), but DVD-RAM, DVD+R, DVD+RW, and any MRW-formatted media and some others do not support simulation mode. The following message is displayed if simulation mode is not supported:

```
Media does not support simulated writing
```

For more information about media types, see "CD/DVD Media Commonly Used Terms" on page 38.

For more information, see `cd`r`w(1).

## Restricting User Access to Removable Media With RBAC

By default, all users can access removable media. 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 “Role-Based Access Control (Overview)” in System Administration Guide: Security Services.
How to Restrict User Access to Removable Media With RBAC

1 Become superuser.

2 Set up a role that includes the Device Management rights.
   For more information, see Chapter 9, “Using Role-Based Access Control (Tasks),” in System Administration Guide: Security Services.
   
   \# roleadd -m -d /export/home/muser -c "mediauser role" -A solaris.device.cdrw -P All muser

3 Add users who need to use the cdrw command to the newly created role.
   \# usermod -R muser joe

4 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 or DVD Writer

1 Identify the CD or DVD writers on the system.
   For example:
   
   \$ cdrw -l
   Looking for CD devices...
   Node | Connected Device | Device type
   --- | ----------------- | ------------
   cdrom0 | YAMAHA CRW88245 | 1.0d | CD Reader/Writer

2 Identify a specific CD or DVD writer.
   For example:
   
   \$ cdrw -a filename.wav -d cdrom2

3 Identify whether the media is blank or whether a table of contents exists on the media.
   For example:
   
   \$ cdrw -M
   Device : YAMAHA CRW88245
How to Check the CD or DVD Media

The `cdrw` command works with or without removable media services running. For more information about disabling or enabling removable media services, see "How to Disable or Enable Removable Media Services" on page 31.

1. Insert a CD or DVD into the drive.
   The CD or DVD can be any CD or DVD that the drive can read.

2. Check that the drive is connected properly by listing the drive.
   
   ```
   $ cdrw -l
   Looking for CD devices...
   Node  Connected Device  Device type
   cdrom1  | YAMAHA  CRW8824S  1.0d  | CD Reader/Writer
   ```

3. (Optional) If you do not see the drive in the list, select one of the following so that the system recognizes the drive.
   - Add the drive without rebooting the system.
     ```
     # devfsadm
     ```
     Then restart removable media services.

Creating a Data CD or DVD

Prepare the data first by using the `mkisofs` command to convert the file and file information into the High Sierra format used on CDs or DVDs.

How to Create an ISO 9660 File System for a Data CD or DVD

1. Insert a blank CD or DVD into the drive.

2. Create the ISO 9660 file system on the new CD or DVD.
   ```
   $ mkisofs -r /pathname > cd-file-system
   - r            Creates Rock Ridge information and resets file ownerships to zero.
   /pathname      Identifies the path name used to create the ISO 9660 file system.
3 Copy the file system onto the CD or DVD.

$ cdrw -i cd-file-system

The `-i cd-file-system` specifies the image file for creating a data CD or DVD.

**Example 2–1 Creating an ISO 9660 File System for a Data CD or DVD**

The following example shows how to create an ISO 9660 file system for a data CD or DVD.

$ mkisofs -r /home/dubs/dir > dubs_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 file system onto the CD.

$ cdrw -i dubs_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 a 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

-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 removable media services.
/data/infoA Identifies the ISO image directory to create.
2 Copy the ISO file system for the first session onto the CD.
   
   $ cdrw -i infoA
   Initializing device...done.
   Writing track 1...done.
   Finalizing (Can take several minutes)...done.
   -i infoA Identifies the name of the image file to write to the CD.
   -0 Keeps the CD open for writing.

3 Reinsert the CD after it is ejected.

4 Identify the path name of the CD media to include in the next write session.
   
   $ eject -n
   .
   .
   .
   cdrom0 -> /vol/dev/rdsk/c2t4d0/my_infoA
   Note the /vol/dev/... path name.

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   |60887        
        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 that you can provide this address
   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.
Creating Rock Ridge information and resets file ownerships to zero.

- Identifies the starting address of the first session and the next writable address.

- Specifies the path of the existing ISO image to be merged.

- 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 described in the following table.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sun</td>
<td>Oracle .au file with data in Red Book CDDA format</td>
</tr>
<tr>
<td>wav</td>
<td>RIFF (.wav) file with data in Red Book CDDA format</td>
</tr>
<tr>
<td>cda</td>
<td>.cda file with raw CD audio data, which is 16-bit PCM stereo at 44.1 kHz sample rate in little-endian byte order</td>
</tr>
<tr>
<td>aur</td>
<td>.aur files with raw CD data in big-endian byte order</td>
</tr>
</tbody>
</table>

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 drive.

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.
Example 2–2 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 multi-session audio CD. The CD is ejected after the first session is written. You would need to reinsert 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 the audio track to a new CD.

If you don’t use the cdrw -T option to specify the audio file type, the cdrw command 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 an audio CD into the CD-RW drive.

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.
audio-file Identifies the audio track to be extracted.

3 Copy the track to a new CD.
   $ cdw -a audio-file

Example 2–3 Extracting and Creating Audio CDs

The following example shows how to extract the first track from an audio CD and name the file song1.wav.

   $ cdw -x -T wav 1 song1.wav
   Extracting audio from track 1...done.

This example shows how to copy a track to an audio CD.

   $ cdw -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 of them onto a blank CD.

Note – By default, the cdw command copies the CD into the /tmp directory. The copying might require up to 700 MB 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 drive.

2 Create a directory for the audio files.
   $ mkdir music_dir

3 Extract the tracks from the audio CD.
   $ cdw -c -m music_dir
   An Extracting audio ... message is display for each track.
   The CD is ejected when all the tracks are extracted.

4 Insert a blank CD and press Return.
   After the tracks are extracted, the audio CD is ejected. You are prompted to insert a blank CD.
Example 2–4 _copying a cd

This example shows 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.

- Erase the entire media or just the last session on the CD by selecting one of the following:
  - 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.
  - Erase the entire media.
    $ cdrw -d cdrom0 -b all
Managing Devices (Tasks)

Device management in the Oracle Solaris release 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. This chapter provides overview information and step-by-step instructions for managing peripheral devices, such as disks, DVD drives, and tape devices, in the Oracle Solaris release.

This is a list of the information in this chapter:
- “What’s New in Device Management?” on page 49
- “Where to Find Additional Device Management Tasks” on page 50
- “Managing Devices in Oracle Solaris” on page 50
- “Adding a Peripheral Device to a System” on page 59
- “Accessing Devices” on page 61

What's New in Device Management?

This section provides information about new device management features in the Oracle Solaris release. For a complete listing of new Oracle Solaris features and a description of Oracle Solaris releases, see Oracle Solaris 10 1/13 What’s New.

Support for USB 3.0 Devices

USB 3.0 support is provided in this Oracle Solaris release. A new USB host controller driver, xhci, is introduced. For more information, see Chapter 5, “Managing USB Devices (Tasks).”
Where to Find Additional 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.

<table>
<thead>
<tr>
<th>Device Management Task</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a disk that is not hot-pluggable.</td>
<td>Chapter 10, &quot;SPARC: Setting Up Disks (Tasks),&quot; or &quot;x86: Setting Up Disks for ZFS File Systems (Task Map)&quot; on page 222</td>
</tr>
<tr>
<td>Hot-plug a SCSI or PCI device.</td>
<td>“SCSI Hot-Plugging With the cfgadm Command” on page 74 or “PCI or PCIe Hot-Plugging With the cfgadm Command” on page 84</td>
</tr>
<tr>
<td>Hot-plug a USB device.</td>
<td>“Managing USB Mass Storage Devices” on page 111</td>
</tr>
<tr>
<td>Add a modem.</td>
<td>Chapter 3, &quot;Managing Serial Ports With the Service Access Facility (Tasks),&quot; in System Administration Guide: Advanced Administration</td>
</tr>
<tr>
<td>Add a printer.</td>
<td>System Administration Guide: Printing</td>
</tr>
<tr>
<td>Secure a device.</td>
<td>Chapter 4, &quot;Controlling Access to Devices (Tasks),&quot; in System Administration Guide: Security Services</td>
</tr>
</tbody>
</table>

Managing Devices in Oracle Solaris

The following sections provide overview information about features that manage devices in Oracle Solaris:

- "About Device Drivers" on page 51
- "Automatic Configuration of Devices" on page 51
- “Displaying Device Configuration Information” on page 52
- “Resolving Faulty Devices” on page 57

For information about accessing devices, see “Accessing Devices” on page 61.

x86: Identifying Device Support

You can use the device detection tool to identify whether your x86 hardware is supported in this Oracle Solaris release. For more information, go to http://www.oracle.com/webfolder/technetwork/hcl/hcts/device_detect.jsp.
About Device Drivers

A computer typically uses a wide range of peripheral devices and mass-storage devices. Your system, for example, probably has a disk drive, a keyboard and a mouse, and some kind of DVD writer.

Other commonly used devices include the following:

- DVD drives
- Printers and plotters
- Light pens
- Touch-sensitive screens
- Digitizers
- Tablet-and-stylus pairs

The Oracle Solaris software does not directly communicate with all of these devices. Each type of device requires different data formats, protocols, and transmission rates.

A device driver is a low-level program that enables 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 consists of a small generic core with a platform-specific component and a set of modules. The kernel is configured automatically in the Oracle Solaris release.

A kernel module is a 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 kernel modules are described in the following table.

<table>
<thead>
<tr>
<th>Description of Solaris Kernel Modules</th>
<th>Location Directory Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform-specific kernel components</td>
<td>/platform/ uname -m/k kernel</td>
</tr>
<tr>
<td>Kernel components common to all platforms that are needed for booting the system</td>
<td>/kernel</td>
</tr>
<tr>
<td>Kernel components common to all platforms within a particular instruction set</td>
<td>/usr/kernel</td>
</tr>
</tbody>
</table>
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 devices 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.

Autoconfiguration is used when you add a new device (and driver) to the system. In previous releases, it was necessary to perform a reconfiguration boot if you added a device to a system that is shutdown. Device configuration enhancements make a reconfiguration boot unnecessary when a device is added to a system that is shutdown.

You can add, remove, or replace devices in the Oracle Solaris OS while the system is still running, if the system components support hot-plugging. For information about hot-plugging devices, see Chapter 4, “Dynamically Configuring Devices (Tasks).”

**What You Need for Unsupported Devices**

Device drivers that are needed to support a wide range of standard devices are included in the Oracle Solaris release. These drivers can be found in the `/kernel/drv` and `/platform/uname` directories.

However, if you have 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 because the device might be incompatible with Oracle Solaris utilities.

For more information about what you need for unsupported devices, contact your device manufacturer.

**Displaying Device Configuration Information**

The following table describes the three commands that are used to display system and device configuration information.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Man Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>prtconf</td>
<td>Displays system configuration information, including the total amount of memory and the device configuration, as described by the system’s device hierarchy. The output that is displayed by this command depends upon the type of system.</td>
<td>prtconf(1M)</td>
</tr>
<tr>
<td>sysdef</td>
<td>Displays device configuration information, including system hardware, pseudo devices, loadable modules, and selected kernel parameters.</td>
<td>sysdef(1M)</td>
</tr>
<tr>
<td>dmesg</td>
<td>Displays system diagnostic messages, as well as a list of devices that are attached to the system since the last reboot.</td>
<td>dmesg(1M)</td>
</tr>
</tbody>
</table>

For information about the device names that are used to identify devices on the system, see "Device Naming Conventions” on page 62.

**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 no device exists at this node or the device is not in use. Drivers are loaded automatically when the device is accessed. They are unloaded when the device is not in use.

**In-Use Device Error Checking**

The following utilities detect when a specified device is in use:

- dumpadm
- format
- mkfs and newfs
- swap

These enhancements mean that these utilities might detect some of the following usage scenarios:

- Device is part of a ZFS storage pool
- Device is a dump or swap device
- Mounted file system or an entry for the device exists in the `/etc/vfstab` file

For example, if you attempt to use the `format` utility to access an active device, you will see a message similar to the following:
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c1t0d0 <FUJITSU-MAY2073RCSUN72G-0401 cyl 8921 alt 2 hd 255 sec 63>
     /pci@0,0/pci1022,7450@2/pci1000,3060@3/sd0@0,0
     /dev/chassis/SYS/HD0/disk
  1. c1t1d0 <FUJITSU-MAY2073RCSUN72G-0401-68.37GB>
     /pci@0,0/pci1022,7450@2/pci1000,3060@3/sd1@0,0
     /dev/chassis/SYS/HD1/disk
Specify disk (enter its number): 0
selecting c1t0d0
[disk formatted]
/dev/dsk/c1t0d0s0 is part of active ZFS pool rpool. Please see zpool(1M).

FORMAT MENU:

▼ How to Display System Configuration Information
Use the output of the `prtconf` and `sysdef` commands to identify which disk, tape, and DVD devices are connected to the system. The output of these commands displays the `driver not attached` messages next to the device instances. Because these devices are always being monitored by some system process, the `driver not attached` message is usually a good indication that no device exists at that device instance.

Use the `sysdef` command to display system configuration information that includes pseudo devices, loadable modules, and selected kernel parameters.

● Display system and device configuration information.
  ▪ Display all the devices connected to a system.
  
  For example, the following `prtconf -v` output on a SPARC system identifies the disk devices connected to the system. The detailed disk information is described in the Device Minor Nodes section.

```
$ /usr/sbin/prtconf -v | more
Memory size: 32640 Megabytes
System Peripherals (Software Nodes):

SUNW,SPARC-Enterprise-T5220

location: /dev/chassis/SUN-Storage-J4400.0918QAKA24/SCSI_Device__2/disk
Device Minor Nodes:
  dev=(27,40)
    dev_path=/pci@0/pci@0/pci@0/pci@0/pci@a/LSILogic,sas@0/sd0@2,0:a
    spectype=blk type=minor
dev_link=/dev/dsk/c4t2d0s0
    dev_path=/pci@0/pci@0/pci@0/pci@0/pci@a/LSILogic,sas@0/sd0@2,0:a,raw
    spectype=chr type=minor
```
Display information about one specific device connected to the system.

For example, the following `prtconf` output on a SPARC system displays the sd instance number for `/dev/dsk/c4t2d0s0`.

```
# prtconf -v /dev/dsk/c4t2d0s0 | grep instance
sd, instance #5
```

Display only the devices that are attached to the system.

```
# prtconf | grep -v not
```

Display device usage information.

For example, the following `fuser` command displays which processes are accessing the `/dev/console` device.

```
# fuser -d /dev/console
/dev/console: 5742o 2269o 20322o 858o
```

Example 3–1 Displaying System Configuration Information

The following `prtconf` output is displayed on an x86 based system.

```
# prtconf
System Configuration: Oracle Corporation i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

i86pc
  scsi_vhci, instance #0
  pci, instance #0
    pci108e,4843, instance #0
    pci8086,25e2, instance #0
    pci8086,3500, instance #7
    pci8086,3510, instance #9
    pci8086,3518, instance #10
    pci108e,4843, instance #0
    pci108e,4843, instance #1
    pci8086,350c, instance #8
    pci8086,25e3 (driver not attached)
    pci8086,25f8, instance #2
    pci108e,286, instance #0
    disk, 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
```
Resolving Faulty Devices

A device retirement mechanism isolates a device as faulty by the fault management framework (FMA). This feature allows faulty devices to be safely and automatically inactivated to avoid data loss, data corruption, or panics and system down time. The retirement process is done safely, taking into account the stability of the system after the device has been retired.

Critical devices are never retired. If you need to manually replace a retired device, use the fmadm repair command after the device replacement so that system knows that the device is replaced, in addition to the manual replacement steps.

For more information, see fmadm(1M).

A general message regarding device retirement is displayed on the console and written to the /var/adm/messages file so that you aware of a retired device. For example:

Aug 9 18:14 starbug genunix: [ID 751201 kern.notice] NOTICE: One or more I/O devices have been retired

You can use the prtconf command to identify specific retired devices. For example:

# prtconf
... ...
pci, instance #2
  scsi, instance #0
    disk (driver not attached)
    tape (driver not attached)
  sd, instance #3
    sd, instance #0 (retired)
    scsi, instance #1 (retired)
      disk (retired)
      tape (retired)
  pci, instance #3
    network, instance #2 (driver not attached)
    network, instance #3 (driver not attached)
    os-io (driver not attached)
    iscsi, instance #0
    pseudo, instance #0
  ...
How to Resolve a Faulty Device

Use the steps that follow to resolve a faulty device or a device that has been retired.

You can also review ZFS device problem or failure information by using the `zpool status` or the `fmadm` command. For ZFS device problem or failure information, see Chapter 10, "Oracle Solaris ZFS Troubleshooting and Pool Recovery," in Oracle Solaris ZFS Administration Guide.

1 Identify the faulted device with the `fmadm faulty` command. For example:

```bash
# fmadm faulty
```

```
TIME       EVENT-ID              MSG-ID    SEVERITY
Jun 20 16:30:52 55c82fff-b709-62f5-b66e-b4e1bbe9dcb1 ZFS-8000-LR Major
```

Problem Status : solved
Diag Engine : zfs-diagnosis / 1.0
System Manufacturer : unknown
Name : ORCL,SPARC-T3-4
Part Number : unknown
Serial Number : 1120808CCD
Host_ID : 84a02d28

Suspect 1 of 1:
Fault class : fault.fs.zfs.open_failed
Certainty : 100%
Affects : zfs://pool=86124fa573cad84e/vdev=25d366e0a7f49/
pool_name=pond/vdev_name=id1, sd@n5000c500335dc0f/a
Status : faulted and taken out of service

FRU Name : "zfs://pool=86124fa573cad84e/vdev=25d366e0a7f49/
pool_name=pond/vdev_name=id1, sd@n5000c500335dc0f/a"
Status : faulty
Description : ZFS device 'id1, sd@n5000c500335dc0f/a' in pool 'pond' failed to open.
Response : An attempt will be made to activate a hot spare if available.
Impact : Fault tolerance of the pool may be compromised.
Action : Use 'fmadm faulty' to provide a more detailed view of this event. Run 'zpool status -lx' for more information. Please refer to the associated reference document at http://support.oracle.com/msg/ZFS-8000-LR for the latest service procedures and policies regarding this diagnosis.

2 Replace the faulty or retired device or clear the device error. For example:

```bash
# zpool clear pond c0t5000c500335dc0f0d0
```

If an intermittent device error occurred but the device was not replaced, you can attempt to clear the previous error.
3 Clear the FMA fault. For example:

```
# fmadm repaired zfs://pool=86124fa573cad84e/vdev=25d36cd46e0a7f49/
  pool_name=pond/vdev_name=id1,sd@n5000c500335dc60f/a
```

4 Confirm that the fault is cleared.

```
# fmadm faulty
```

If the error is cleared, the `fmadm faulty` command returns nothing.

---

### Adding a Peripheral Device to a System

Adding a new peripheral device that is not hot-pluggable usually involves the following:

- Shutting down the system
- Connecting the device to the system
- Rebooting the system

Use "How to Add a Peripheral Device" on page 59 to add the following devices that are not hot-pluggable to a system:

- DVD drive
- Secondary disk drive
- Tape drive

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 4, "Dynamically Configuring Devices (Tasks)."

---

### How to Add a Peripheral Device

1 Become superuser.

2 (Optional) If you need to add a device driver to support the device, complete the procedure "How to Add a Device Driver" on page 60.

3 Shut down the system.

```
# shutdown -i 0 -g 30 -y
```

- `-i 0` Brings the system to the 0 init state, which is the appropriate state for turning the system power off for adding and removing devices.
- `-g 30` Shuts the system down in 30 seconds. The default is 60 seconds.
Continues the system shutdown without user intervention. Otherwise, you are prompted to continue the shutdown process.

4 Select one of the following to turn off power to the system after it is shut down:
   ■ For SPARC platforms, it is safe to turn off power if the ok prompt is displayed.
   ■ For x86 platforms, it is safe to turn off power if the type any key to continue prompt is displayed.

5 Turn off power to all peripheral devices.
   For the location of power switches on any peripheral devices, refer to the hardware installation guides that accompany your peripheral devices.

6 Install the peripheral device, making sure that the hardware configuration adheres to manufacturer specifications.
   Refer to the hardware installation guide that accompanies the peripheral device for information on installing and connecting the device.

7 Turn on the power to the system.
   The system boots to multiuser mode, and the login prompt is displayed.

8 Verify that the peripheral device has been added by attempting to access the device.
   For information on accessing the device, see “Accessing Devices” on page 61.

▼ How to Add a Device Driver

This procedure assumes that the device has already been added to the system. If not, see “How to Add a Peripheral Device” on page 59.

1 Become superuser.

2 Place the media into the drive.

3 Install the driver.
   
   # pkgadd [-d] device package-name

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.
Adding a Device Driver

The following example shows how to install and verify a package called XYZdrv.

```bash
# pkgadd XYZdrv
(licensing messages displayed)
```

```
Installing XYZ Company driver as <XYZdrv>
```

```
Installation of <XYZdrv> was successful.
```

```bash
# pkgchk XYZdrv
```

## 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. The kernel also provides a set of pointers to the drivers that perform specific operations.

### How Devices Are Managed

The devfs file system manages the /devices directory, which is the name space of all devices on the system. This directory represents the physical devices that consists of actual bus and device addresses.

The dev file system manages the /dev directory, which is the name space of logical device names.

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. The devfsadm command also maintains the path_to_inst instance database.
Updates to the /dev and /devices directories in response to dynamic reconfiguration events or file system accesses are handled by devfsadm, the daemon version of the devfsadm command. This daemon is started by the service management facility when a system is booted.

Because the devfsadm daemon automatically detects device configuration changes generated by any reconfiguration event, there is no need to run this command interactively.

For more information, see the following references:

- devfsadm(1M)
- devfs(7FS)
- path_to_inst(4)

### Device Naming Conventions

Devices are referenced in one of three ways in Oracle Solaris:

- **Physical device name** – Represents the full device path name in the device information hierarchy. The physical device name is created by when the device is first added to the system. 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.

- **Logical device name** – The logical device name is created by when the device is first added to the system. Logical device names are used with most file system commands to refer to devices. For a list of file commands that use logical device names, see Table 3-3. Logical device files in the /dev directory are symbolically linked to physical device files in the /devices directory.

The preceding device name information is displayed with the following commands:

- dmesg
- format
- sysdef
- prtconf

### Logical Disk Device Names

Logical device names are used to access disk devices when you perform the following tasks:

- 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/rdisk, followed by a string identifying the particular controller, disk, and slice.

**FIGURE 3–1  Description of Logical Device Names**

`/dev/[r]disk/cvtwx[psyz]`

- **Slice number** (s0 to s7) or fdisk **partition number** (p0 to p4)
- **Drive number**
- **Physical bus target number**
- **Logical controller number**
- **Raw disk device subdirectory**
- **Devices directory**

**Specifying the Disk Subdirectory**

Some 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 rdisk 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/risk`, 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 3-3  Device Interface Type Required by Some Frequently Used Commands

<table>
<thead>
<tr>
<th>Command Reference</th>
<th>Interface Type</th>
<th>Example of Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dumpadm(1M)</code></td>
<td>Block</td>
<td><code>dumpadm -d /dev/zvol/dsk/rpool/dump</code></td>
</tr>
<tr>
<td><code>prtvtoc(1M)</code></td>
<td>Raw</td>
<td><code>prtvtoc /dev/rdsk/c0t0d0s0</code></td>
</tr>
<tr>
<td><code>swap(1M)</code></td>
<td>Block</td>
<td><code>swap -a /dev/zvol/dsk/rpool/swap</code></td>
</tr>
</tbody>
</table>

### Direct and Bus-Oriented Controllers

You might access disk partitions or slices differently depending upon whether the disk device is connected to a direct or bus-oriented controller. Generally, direct controllers do not include a `target` identifier in the logical device name.

The conventions for both types of controllers are explained in the following subsections.

**Note** – Controller numbers are assigned automatically during system initialization. The numbers are strictly logical and imply no direct mapping to physical controllers.

### Disks With Direct Controllers

To specify a slice on a disk with an IDE controller, follow the naming convention that is shown in the following figure.

**FIGURE 3-2**  Disks With Direct Controllers

```
cwx [s0, s7] [p0, p4]
```

Slice number (s0 to s7) or `fdisk` partition number (p0 to p4)

Drive number

Logical controller number

To indicate the entire `fdisk` partition, specify slice 2 (s2).

### Disks With Bus-Oriented Controllers

To specify a slice on a disk with a bus-oriented controller, SCSI for instance, follow the naming convention that is shown in the following figure.
To indicate the whole disk, 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.

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 22, “Managing Tape Drives (Tasks).”

**Logical Removable Media Device Names**

Since removable media is managed by removable media management services, 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 “Accessing Removable Media” on page 30.
This chapter provides instructions for dynamically configuring devices in the Oracle Solaris OS. You can add, remove, or replace devices in the Oracle Solaris OS while the system is still running, if the system components support hot-plugging. If the system components do not support hot-plugging, new devices will be configured at boot time, after the new components are installed in the system.

For information about the procedures that are associated with dynamically configuring devices, see the following sections:

- “SCSI Hot-Plugging With the \texttt{cfgadm} Command (Task Map)” on page 73
- “PCI or PCIe Hot-Plugging With the \texttt{cfgadm} Command (Task Map)” on page 83
- “SATA Hot-Plugging With the \texttt{cfgadm} Command” on page 90
- “Application Developer RCM Script (Task Map)” on page 93
- “System Administrator RCM Script (Task Map)” on page 94

For information on hot-plugging USB devices with the \texttt{cfgadm} command, see “Hot-Plugging USB Devices With the \texttt{cfgadm} Command” on page 131.

For information on hot-plugging InfiniBand devices with \texttt{cfgadm} command, see Chapter 6, “Using InfiniBand Devices (Overview/Tasks).”

For information about accessing devices, see “Accessing Devices” on page 61.

**Dynamic Reconfiguration and Hot-Plugging**

The following information is described in this section:

- “Attachment Points” on page 69
- “Detaching PCI or PCIe Adapter Cards” on page 71
- “Attaching PCI or PCIe Adapter Cards” on page 71
- “PCIe Hot-Plugging With the \texttt{(hotplug)} Command” on page 71
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.

Generally, you can hot-plug the following bus types:

- USB
- Fibre Channel
- 1394
- ATA
- SCSI
- Infiniband

In addition, you can hot-plug the following devices with the `cfgadm` command on both SPARC and x86 platforms:

- USB devices
- SCSI or SAS devices
- PCI devices
- PCIe devices
- SATA devices
- InfiniBand devices

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 components, see the following:

- “SCSI Hot-Plugging With the `cfgadm` Command” on page 74
- “PCI or PCIe Hot-Plugging With the `cfgadm` Command” on page 84
- “SATA Hot-Plugging With the `cfgadm` Command” on page 90
- `cfgadm(1M)`

**Note** – Not all SCSI and PCI controllers support hot-plugging with the `cfgadm` command.
As part of Oracle’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** – Some systems have slots that hot-pluggable and slots that are not hot-pluggable. For information about hot-plugging devices on your specific hardware configuration, such as on enterprise-level systems, 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_Ids). The physical Ap_Id is the physical path name of the attachment point. The logical Ap_Id is a user-friendly alternative for the physical Ap_Id. For more information on Ap_Ids, 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 has this format:

`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 and unconfigured) and operations (such as connect, configure, unconfigure, and so on). For more information on these common states and operations, see cfgadm(1M).

The following table shows the receptacle and occupant states for the SCSI HBA attachment points.

<table>
<thead>
<tr>
<th>Receptacle State</th>
<th>Description</th>
<th>Occupant State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty</td>
<td>N/A for SCSI HBA</td>
<td>configured</td>
<td>One or more devices is configured on the bus</td>
</tr>
<tr>
<td>disconnected</td>
<td>Bus quiesced</td>
<td>unconfigured</td>
<td>No devices are configured</td>
</tr>
<tr>
<td>connected</td>
<td>Bus active</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following table shows the receptacle and occupant states for SCSI device attachment points.

<table>
<thead>
<tr>
<th>Receptacle State</th>
<th>Description</th>
<th>Occupant State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty</td>
<td>N/A for SCSI devices</td>
<td>configured</td>
<td>Device is configured</td>
</tr>
<tr>
<td>disconnected</td>
<td>Bus quiesced</td>
<td>unconfigured</td>
<td>Device is not configured</td>
</tr>
<tr>
<td>connected</td>
<td>Bus active</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The state of SCSI attachment points is unknown unless special hardware indicates otherwise. For instructions on displaying SCSI component information, see "How to Display Information About SCSI Devices" on page 74.
Detaching PCI or PCIe 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.

Attaching PCI or PCIe 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 “PCI or PCIe Hot-Plugging With the cfgadm Command” on page 84.

PCIe Hot-Plugging With the (hotplug) Command

You can use the hotplug command to manage hot pluggable connections, where a connection can be a connector or port, on PCI Express (PCIe) and PCI SHPC devices only. A hotplug connector is a physical representation in the system where a component is inserted or removed. A hotplug port is a logical representation in the system device tree where the connection of a device to the system is managed.

You can use the hotplug features to online or offline a device, even an on-board device, without physically adding or removing the device from the system.

You must enable the hotplug service to manage devices with the hotplug command.

# svcadm enable svc:/system/hotplug:default

The following examples show how to use the hotplug command:

Display all the PCI/PCIe hot-pluggable connectors/ports (virtual and physical) in the system as follows:

# hotplug list -lv
Configure an Ethernet card in a PCIe slot. For example:

```
# hotplug enable /pci0,0 pcie0
```

Unconfigure an Ethernet card in a PCIe slot. For example:

```
# hotplug disable /pci0,0 pcie0
```

Offline a PCI device node, which means detach the device driver for that node. For example:

```
# hotplug offline /pci0,0/pci1 pci.0,2
```

Online a PCI device node, which means attach the driver for that node. For example:

```
# hotplug online /pci0,0/pci1 pci.0,2
```

Install dependent ports of an IOV physical function. For example:

```
# hotplug install /pci@400/pci@1/pci@0/pci@4 pci.0,1
```

Then, display the resulting IOV virtual functions that were probed. For example:

```
# hotplug list -v /pci@400/pci@1/pci@0/pci@4 pci.0,1
<pci.0,1> (ONLINE)
  { IOV physical function }
  { IOV virtual function 'pci.0,81' }
  { IOV virtual function 'pci.0,83' }
  { IOV virtual function 'pci.0,85' }
  { IOV virtual function 'pci.0,87' }
<pci.0,81> (OFFLINE)
  ethernet@0,81
<pci.0,83> (OFFLINE)
  ethernet@0,83
<pci.0,85> (OFFLINE)
  ethernet@0,85
<pci.0,87> (OFFLINE)
  ethernet@0,87
```

Uninstall the dependent ports of an IOV physical function. For example:

```
# hotplug uninstall /pci@400/pci@1/pci@0/pci@4 pci.0,0
```

This operation fails if a dependent IOV virtual function is busy. For example:

```
# hotplug uninstall /pci@400/pci@1/pci@0/pci@4 pci.0,0
ERROR: devices or resources are busy. 
ethernet@0,81:
  { Network interface igbvfl }
  { igbvfl: hosts IP addresses: 10.0.0.1 }
  { Plumbed IP Address }
```
Troubleshooting PCI Hot Plug Operations (**hotplug**)

The following service must be running to use the **hotplug** command.

```
svc:/system/hotplug:default
```

Otherwise, you will see the following message:

**ERROR:** **hotplug** service is not available.

The following error message is displayed on systems that do not have any supported I/O buses:

**ERROR:** there are no connections to display.
(See **hotplug(1m)** for more information.)

The above message could mean that the system might have other hot-pluggable I/O devices, but you need to use the **cfgadm** command rather than the **hotplug** command to manage these devices.

---

### SCSI Hot-Plugging With the **cfgadm** Command (Task Map)

#### Task Description For Instructions

<table>
<thead>
<tr>
<th>Task Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display information about SCSI devices.</td>
<td>&quot;How to Display Information About SCSI Devices&quot; on page 74</td>
</tr>
<tr>
<td>Unconfigure a SCSI controller.</td>
<td>&quot;How to Unconfigure a SCSI Controller&quot; on page 75</td>
</tr>
<tr>
<td>Configure a SCSI controller.</td>
<td>&quot;How to Configure a SCSI Controller&quot; on page 75</td>
</tr>
<tr>
<td>Configure a SCSI device.</td>
<td>&quot;How to Configure a SCSI Device&quot; on page 76</td>
</tr>
<tr>
<td>Disconnect a SCSI controller.</td>
<td>&quot;How to Disconnect a SCSI Controller&quot; on page 77</td>
</tr>
<tr>
<td>Connect a SCSI controller.</td>
<td>&quot;SPARC: How to Connect a SCSI Controller&quot; on page 78</td>
</tr>
<tr>
<td>Add a SCSI device to a SCSI bus.</td>
<td>&quot;SPARC: How to Add a SCSI Device to a SCSI Bus&quot; on page 78</td>
</tr>
<tr>
<td>Replace an identical device on a SCSI controller.</td>
<td>&quot;SPARC: How to Replace an Identical Device on a SCSI Controller&quot; on page 79</td>
</tr>
<tr>
<td>Remove a SCSI device.</td>
<td>&quot;SPARC: How to Remove a SCSI Device&quot; on page 81</td>
</tr>
<tr>
<td>Task</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>Troubleshoot SCSI configuration problems.</td>
<td>Resolve a failed SCSI unconfigure operation.</td>
</tr>
</tbody>
</table>

### SCSI Hot-Plugging With the `cfgadm` Command

This section describes various SCSI hot-plugging procedures that you can perform with the `cfgadm` command.

**Note** – The SCSI framework generally supports hot-plugging of SCSI devices. However, you should consult your hardware documentation to confirm whether hot-plugging is supported for your SCSI devices.

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 `c2` and `c3` and the devices that are attached to them in the 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, the device does not display in the `cfgadm` command output.

1. **Become superuser.**

2. **Display information about attachment points on the system.**

   ```bash
   # cfgadm -l
   Ap_Id | Type   | Receptacle | Occupant | Condition
   c2    | scsi-bus | connected  | configured | unknown
   c3    | scsi-sas | connected  | configured | unknown
   ```

   In this example, `c2` and `c3` represent two SCSI controllers.

3. **Display information about a system’s SCSI controllers and their attached devices.**

   ```bash
   # cfgadm -al
   Ap_Id | Type   | Receptacle | Occupant | Condition
   c2    | scsi-bus | connected  | configured | unknown
   ```
How to Unconfigure a SCSI Controller

The following procedure uses SCSI controller c2 in the example of unconfiguring a SCSI controller.

1 Become superuser.

2 Unconfigure a SCSI controller.
   
   # cfgadm -c unconfigure c2

3 Verify that the SCSI controller is unconfigured.
   
   # cfgadm -al

```
<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Type</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2</td>
<td>scsi-bus</td>
<td>connected</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
<tr>
<td>c2::dsk/c2t0d0</td>
<td>unavailable</td>
<td>connected</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
<tr>
<td>c3</td>
<td>scsi-sas</td>
<td>connected</td>
<td>configured</td>
<td>unknown</td>
</tr>
<tr>
<td>c3::dsk/c3t0d0</td>
<td>disk</td>
<td>connected</td>
<td>configured</td>
<td>unknown</td>
</tr>
<tr>
<td>c3::dsk/c3t1d0</td>
<td>disk</td>
<td>connected</td>
<td>configured</td>
<td>unknown</td>
</tr>
<tr>
<td>c3::dsk/c3t2d0</td>
<td>disk</td>
<td>connected</td>
<td>configured</td>
<td>unknown</td>
</tr>
<tr>
<td>c3::dsk/c3t3d0</td>
<td>disk</td>
<td>connected</td>
<td>configured</td>
<td>unknown</td>
</tr>
</tbody>
</table>
```

Notice that the Occupant column for c2 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 83.

How to Configure a SCSI Controller

The following procedure uses SCSI controller c2 in the example of configuring a SCSI controller.

1 Become superuser.

2 Configure a SCSI controller.
   
   # cfgadm -c configure c2
3 Verify that the SCSI controller is configured.

```
# cfgadm -al
Ap_Id       Type     Receptacle    Occupant   Condition
----------  --------  ------------  ---------  ---------------
c2          scsi-bus connected configured unknown
c2::dsk/c2t0d0 unavailable connected unconfigured unknown
c3          scsi-sas connected configured unknown
c3::dsk/c3t0d0 disk    connected configured unknown
c3::dsk/c3t1d0 disk    connected configured unknown
c3::dsk/c3t2d0 disk    connected configured unknown
c3::dsk/c3t3d0 disk    connected configured 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 c2t0d0 in the example of configuring a SCSI device.

1 Become superuser.

2 Identify the device to be configured.

```
# cfgadm -al
Ap_Id       Type     Receptacle    Occupant   Condition
----------  --------  ------------  ---------  ---------------
c2          scsi-bus connected configured unknown
c2::dsk/c2t0d0 unavailable connected unconfigured unknown
c3          scsi-sas connected configured unknown
c3::dsk/c3t0d0 disk    connected configured unknown
c3::dsk/c3t1d0 disk    connected configured unknown
c3::dsk/c3t2d0 disk    connected configured unknown
c3::dsk/c3t3d0 disk    connected configured unknown
```

3 Configure the SCSI device.

```
# cfgadm -c configure c2::dsk/c2t0d0
```

4 Verify that the SCSI device is configured.

```
# cfgadm -al
Ap_Id       Type     Receptacle    Occupant   Condition
----------  --------  ------------  ---------  ---------------
c2          scsi-bus connected configured unknown
c2::dsk/c2t0d0 CD-ROM    connected configured unknown
c3          scsi-sas connected configured unknown
c3::dsk/c3t0d0 disk    connected configured unknown
c3::dsk/c3t1d0 disk    connected configured unknown
c3::dsk/c3t2d0 disk    connected configured unknown
c3::dsk/c3t3d0 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 components of the root file system. 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 c2 in the 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
   c2      scsi-bus  connected   configured    unknown
   c2::dsk/c2t0d0  CD-ROM    connected   configured    unknown
   c3      scsi-sas  connected   configured    unknown
   c3::dsk/c3t0d0  disk      connected   configured    unknown
   c3::dsk/c3t1d0
   c3::dsk/c3t2d0
   c3::dsk/c3t3d0

3 Disconnect the SCSI controller.

   # cfgadm -c disconnect c2
   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
   c2      unavailable disconnected configured unknown
   c2::dsk/c2t0d0 unavailable disconnected configured unknown
   c3      scsi-sas  connected   configured    unknown
   c3::dsk/c3t0d0  disk      connected   configured    unknown
   c3::dsk/c3t1d0
   c3::dsk/c3t2d0
   c3::dsk/c3t3d0

The controller and all the devices that are attached to it are disconnected from the system.
SPARC: How to Connect a SCSI Controller

The following procedure uses SCSI controller c2 in the 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
   c2     unavailable disconnected configured unknown
   c2::dsk/c2t0d0 unavailable disconnected configured unknown
   c3     scsi-sas connected configured unknown
   c3::dsk/c3t0d0 disk connected configured unknown
   c3::dsk/c3t1d0 disk connected configured unknown
   c3::dsk/c3t2d0 disk connected configured unknown
   c3::dsk/c3t3d0 disk connected configured unknown

3 Connect the SCSI controller.

   # cfgadm -c connect c2

4 Verify that the SCSI controller is connected.

   # cfgadm -al
   Ap_Id  Type     Receptacle    Occupant Condition
   c2     scsi-bus connected configured unknown
   c2::dsk/c2t0d0 CD-ROM connected configured unknown
   c3     scsi-sas connected configured unknown
   c3::dsk/c3t0d0 disk connected configured unknown
   c3::dsk/c3t1d0 disk connected configured unknown
   c3::dsk/c3t2d0 disk connected configured unknown
   c3::dsk/c3t3d0 disk connected configured unknown

SPARC: How to Add a SCSI Device to a SCSI Bus

SCSI controller c3 is used in the 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
   c2     scsi-bus connected configured unknown
   c2::dsk/c2t0d0 CD-ROM connected configured unknown
   c3     scsi-sas connected configured unknown
3 Add the SCSI device to the SCSI bus.

a. Type the following `cfgadm` command.
   For example:
   ```
   # cfgadm -x insert_device c3
   Adding device to SCSI HBA: /devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2
   This operation will suspend activity on SCSI bus: c3
   ```

b. 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.

c. Connect the device and then power it on.

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 added.

   ```
   # cfgadm -al
   Ap_Id Type Receptacle Occupant Condition
   c2 scsi-bus connected configured unknown
   c2::dsk/c2t0d0 CD-ROM connected configured unknown
   c3 scsi-sas connected configured unknown
   c3::dsk/c3t0d0 disk connected configured unknown
   c3::dsk/c3t1d0 disk connected configured unknown
   c3::dsk/c3t2d0 disk connected configured unknown
   c3::dsk/c3t3d0 disk connected configured unknown
   ```
   A new disk has been added to controller c3.

**SPARC: How to Replace an Identical Device on a SCSI Controller**

The following procedure uses SCSI disk c3t3d0 in the example of replacing an identical device on a SCSI controller.
Review the following conditions when attempting to replace an identical device on a SCSI controller:

- If you are replacing a mirrored or unmirrored boot device that is part of a ZFS root pool, see the following reference:
  

- If the device is controlled by legacy volume management software, consult your volume management documentation for the specific steps to replace a device. If you have an active Oracle support plan, see the following references:
  - If you are using Solaris Volume Manager (SVM), see MOS document 1010753.1.
  - If you are using Veritas Volume Manager (VxVM), see MOS document 1003122.1, 1011782.1, and 1002285.1.

1. Become superuser.

2. Identify the current SCSI configuration.

   ```bash
   # cfgadm -al
   c2::dsk/c2t0d0         scsi-bus   connected  configured  unknown
   c2::dsk/c2t1d0         CD-ROM     connected  configured  unknown
   c3::dsk/c3t0d0         scsi-sas   connected  configured  unknown
   c3::dsk/c3t1d0         disk       connected  configured  unknown
   c3::dsk/c3t2d0         disk       connected  configured  unknown
   c3::dsk/c3t3d0         disk       connected  configured  unknown
   ```

3. Replace a device on the SCSI bus with another device of the same type.

   a. Type the following `cfgadm` command.

      ```bash
      # cfgadm -x replace_device c3::dsk/c3t3d0
      Replacing SCSI device: /devices/pci@7c0/pci@0/pci@1/pci@0,2/LSILogic,sas@2/sd@3,0
      This operation will suspend activity on SCSI bus: c3
      ```

   b. 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.

      ```bash
      Continue (yes/no)? y
      SCSI bus quiesced successfully.
      It is now safe to proceed with hotplug operation.
      ```

   c. Power off the device to be removed and remove it.

   d. 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.
e. 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
  c2      scsi-bus    connected   configured unknown
  c2::dsk/c2t0d0  CD-ROM     connected   configured unknown
  c3      scsi-sas    connected   configured unknown
  c3::dsk/c3t0d0  disk       connected   configured unknown
  c3::dsk/c3t1d0  disk       connected   configured unknown
  c3::dsk/c3t2d0  disk       connected   configured unknown
  c3::dsk/c3t3d0  disk       connected   configured unknown
```

▼ SPARC: How to Remove a SCSI Device

The following procedure uses SCSI disk c3t3d0 in the 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
  c2      scsi-bus    connected   configured unknown
  c2::dsk/c2t0d0  CD-ROM     connected   configured unknown
  c3      scsi-sas    connected   configured unknown
  c3::dsk/c3t0d0  disk       connected   configured unknown
  c3::dsk/c3t1d0  disk       connected   configured unknown
  c3::dsk/c3t2d0  disk       connected   configured unknown
  c3::dsk/c3t3d0  disk       connected   configured unknown
```

3 Remove the SCSI device from the system.

a. Type the following `cfgadm` command.

For example:

```
# cfgadm -x remove_device c3::dsk/c3t3d0
Removing SCSI device: /devices/pci@7c0/pci@0/pci@0/pci@0,2/LSILogic,sas@2/sd@3,0
This operation will suspend activity on SCSI bus: c3
```

b. 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.

c. Power off the device to be removed and remove it.
d. Type \textit{y} at the \texttt{Enter y if operation is complete or n to abort (yes/no)}? prompt.

\texttt{Enter y if operation is complete or n to abort (yes/no)? y}

\textbf{Note} – This step must be performed if you are removing a SCSI RAID device from a SCSI RAID array.

4 Verify that the device has been removed from the system.

\texttt{# cfgadm -al}

\begin{tabular}{lllll}
\text{Ap Id} & \text{Type} & \text{Receptacle} & \text{Occuptant} & \text{Condition} \\
c2 & scsi-bus & connected & configured & unknown \\
c2::dsk/c2t0d0 & CD-ROM & connected & configured & unknown \\
c3 & scsi-sas & connected & configured & unknown \\
c3::dsk/c3t0d0 & disk & connected & configured & unknown \\
c3::dsk/c3t1d0 & disk & connected & configured & unknown \\
c3::dsk/c3t2d0 & disk & connected & configured & unknown \\
\end{tabular}

\section*{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 \texttt{cfgadm(1M)}.

\textbf{Error Message}

\texttt{cfgadm: Component system is busy, try again: failed to offline: }

\begin{tabular}{ll}
\textit{device-path} \\
Resource \\
\hline
\end{tabular}

\begin{tabular}{ll}
\textit{Information} \\
/mdisk/ /mnt/c2t0d0s0 mounted filesystem "/file-system" \\
\hline
\end{tabular}

\textbf{Cause}

You attempted to remove or replace a device with a mounted file system.

\textbf{Solution}

Unmount the file system that is listed in the error message and retry the \texttt{cfgadm} operation.

\textbf{Error Message}

\texttt{cfgadm: Component system is busy, try again: failed to offline: }

\begin{tabular}{ll}
\textit{device-path} \\
Resource \\
\hline
\end{tabular}

\begin{tabular}{ll}
\textit{Information} \\
/mdisk/ /mnt/c2t0d0s0 swap area \\
\hline
\end{tabular}

\textbf{Cause}

If you use the \texttt{cfgadm} command to remove a system resource, such as a swap device or a dedicated dump device, a similar error message is displayed if the system resource is still active.

\textbf{Solution}

Unconfigure the swap areas on the device that is specified and retry the \texttt{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 dedicated 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.

2 Reconfigure the controller.
   # cfgadm -c configure device-name

PCI or PCIe Hot-Plugging With the `cfgadm` Command (Task Map)

The following task map describes the tasks for managing PCI or PCIe devices on your system.
### PCI or PCIe Hot-Plugging With the `cfgadm` Command

This section provides step-by-step instructions for hot-plugging PCI or PCIe adapter cards on SPARC and x86 based systems.

In addition to the `cfgadm` command, the `prtconf` command is helpful during hot-plug operations. The `prtconf` command displays additional configuration information that pertains to the hardware.

After adding hardware, you can use the `prtconf` command to verify that the hardware is configured correctly. For example, after a configure operation, use the `prtconf -d` command to verify that the driver is attached to the newly installed hardware device. If the device driver has not been added to the system prior to hardware configuration, it may be manually added by using the `add_drv` command.

For more information, see `prtconf(1M)` and `add_drv(1M)`.

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.

### PCIe LED Indicator Behavior

You might observe the LED indicators on the system to get a visual indication about the status of the slot’s hot-plug operation. The LED behavior, in case of PCI Express, matches that defined in the PCI Express specification or the behavior might otherwise be platform dependent.

Please refer to your platform guide for specific details. In case of PCI Express, when the Attention Button is pressed, the power indicator blinks, which indicates the beginning of a state transition. The blinking ends when the state transition has ended.
How to Display PCI Slot Configuration Information

This procedure has been updated to include PCIe configuration information.

The `cfgadm` 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 configuration information.

   - Display PCI slot configuration information.
     
     For example:
     ```
     # cfgadm
     Ap_Id: Type: Receptacle: Occupant: Condition
     pci1:0:0: slot0: unknown: empty: unconfigured: unknown
     pci1:0:0: slot1: unknown: empty: unconfigured: unknown
     pci1:0:0: slot2: unknown: empty: unconfigured: unknown
     pci1:0:0: slot3: ethernet/hp: connected: configured: ok
     pci1:0:0: slot4: unknown: empty: unconfigured: unknown
     ```

   - Display specific PCI device information.
     
     For example:
     ```
     # cfgadm -s "cols=ap_id:type:info" pci
     Ap_Id: Type: Information
     pci1:0:0: slot0: unknown: Slot 7
     pci1:0:0: slot1: unknown: Slot 8
     pci1:0:0: slot2: unknown: Slot 9
     pci1:0:0: slot3: ethernet/hp: Slot 10
     pci1:0:0: slot4: unknown: Slot 11
     ```

     The logical `Ap_Id`, `pci1:0:0:slot0`, is the logical `Ap_Id` for hot-pluggable slot, Slot 7. The component `0:0` 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.

   - Display PCIe slot configuration information.
     
     For example:
     ```
     # cfgadm pci
     Ap_Id: Type: Receptacle: Occupant: Condition
     pcie1: unknown: empty: unconfigured: unknown
     pcie2: unknown: empty: unconfigured: unknown
     pcie3: unknown: empty: unconfigured: unknown
     pcie4: ethernet/hp: connected: configured: ok
     pcie5: pci-pci/hp: connected: configured: ok
     pcie6: unknown: disconnected: unconfigured: unknown
     ```

PCI or PCIe Hot-Plugging With the `cfgadm` Command
Display specific PCIe device information.

For example:

```
# cfgadm -s "cols=ap_id:busy:o_state" pci
```

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Busy</th>
<th>Occupant</th>
</tr>
</thead>
<tbody>
<tr>
<td>pcie1</td>
<td>n</td>
<td>unconfigured</td>
</tr>
<tr>
<td>pcie2</td>
<td>n</td>
<td>unconfigured</td>
</tr>
<tr>
<td>pcie3</td>
<td>n</td>
<td>unconfigured</td>
</tr>
<tr>
<td>pcie4</td>
<td>n</td>
<td>configured</td>
</tr>
<tr>
<td>pcie5</td>
<td>n</td>
<td>configured</td>
</tr>
<tr>
<td>pcie6</td>
<td>n</td>
<td>configured</td>
</tr>
</tbody>
</table>

Note – The logical Ap_Id in most cases should match the slot label that is silk-screened on the system chassis. Refer to your platform guide for the cfgadm output of the hot-pluggable slots. The Busy field can be displayed to ensure that the Ap_Id is not transitioning to another state before a hot-plug operation is attempted.

**How to Remove a PCI Adapter Card**

The following procedure has been updated for removing a PCIe adapter card. However, the procedure to remove an adapter card is the same whether you are using PCI or PCIe.

1. Become superuser.

2. Determine which slot the PCI adapter card is in.

   For example:

   ```
   # cfgadm pci
   ```

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Type</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>pcie1</td>
<td>unknown</td>
<td>empty</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
<tr>
<td>pcie2</td>
<td>unknown</td>
<td>empty</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
<tr>
<td>pcie3</td>
<td>unknown</td>
<td>empty</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
<tr>
<td>pcie4</td>
<td>etherne/hp</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>pcie5</td>
<td>pci-pci/hp</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>pcie6</td>
<td>unknown</td>
<td>disconnected</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
</tbody>
</table>

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 unplug the interface. For example:

   ```
   # ifconfig unplumb bge1
   ```

4. Unconfigure the device manually by using the `cfgadm` command as shown below. Or, if you have a PCIe adapter card, use the auto-configuration method, such as pressing the slot’s Attention Button as defined by your platform guide.

   ```
   # cfgadm -c unconfigure pcie4
   ```
5 Confirm that the device has been unconfigured.

For example:

```
# cfgadm pci
Ap_Id    Type    Receptacle Occupant Condition
pcie1    unknown empty unconfigured unknown
pcie2    unknown empty unconfigured unknown
pcie3    unknown empty unconfigured unknown
pcie4    unknown connected unconfigured unknown
pcie5    pci-pci/hp connected configured ok
pcie6    unknown disconnected unconfigured unknown
```

Note – The Type and Condition also become unknown when the device is unconfigured.

6 Disconnect the power to the slot manually. If the auto-configuration method is used, this step is not necessary.

```
# cfgadm -c disconnect pcie4
```

Refer to your platform guide for more information.

7 Confirm that the device has been disconnected.

For example:

```
# cfgadm pci
Ap_Id    Type    Receptacle Occupant Condition
pcie1    unknown empty unconfigured unknown
pcie2    unknown empty unconfigured unknown
pcie3    unknown empty unconfigured unknown
pcie4    unknown disconnected unconfigured unknown
pcie5    pci-pci/hp connected configured ok
pcie6    unknown disconnected unconfigured unknown
```

8 Follow appropriate instructions in your platform guide to remove the PCI adapter card. After the card is removed, the Receptacle state is empty.

For example:

```
# cfgadm pci
Ap_Id    Type    Receptacle Occupant Condition
pcie1    unknown empty unconfigured unknown
pcie2    unknown empty unconfigured unknown
pcie3    unknown empty unconfigured unknown
pcie4    unknown disconnected unconfigured unknown
pcie5    pci-pci/hp connected configured ok
pcie6    unknown disconnected unconfigured unknown
```

Note – The auto-configuration method can be enabled or disabled at boot, depending on the platform implementation. Set the auto-configuration method as appropriate for your environment.
How to Add a PCI Adapter Card

The following procedure has been updated for adding a PCIe adapter card. However, the procedure to add an adapter card is the same whether you are using PCI or PCIe.

1. Become superuser.

2. Identify the hot-pluggable slot and open latches.
   For example, pcie3.

   ```
   # cfgadm pci
   Ap_Id  Type       Receptacle  Occupant Condition
   pcie1  unknown   empty       unconfigured unknown
   pcie2  unknown   empty       unconfigured unknown
   pcie3  unknown   empty       unconfigured unknown
   pcie4  unknown   empty       unconfigured unknown
   pcie5 pci-pci/hp connected configured ok
   pcie6  unknown   disconnected unconfigured unknown
   ```

3. Follow the appropriate instructions in your platform guide to insert a PCI adapter card into the slot.

4. Determine which slot the PCI adapter card is in after it is inserted.
   For example:

   ```
   # cfgadm pci
   Ap_Id  Type       Receptacle  Occupant Condition
   pcie1  unknown   empty       unconfigured unknown
   pcie2  unknown   empty       unconfigured unknown
   pcie3  unknown   disconnected unconfigured unknown
   pcie4  unknown   empty       unconfigured unknown
   pcie5 pci-pci/hp connected configured ok
   pcie6  unknown   disconnected unconfigured unknown
   ```

5. Connect the power to the slot manually using the `cfgadm` command. Or, if you have a PCIe adapter card, use the auto-configuration method, such as pressing the slot’s Attention Button as defined by your platform guide.
   For example:
   ```
   # cfgadm -c connect pcie3
   ```

6. Confirm that the attachment point is connected.
   For example:

   ```
   # cfgadm pci
   Ap_Id  Type       Receptacle  Occupant Condition
   pcie1  unknown   empty       unconfigured unknown
   pcie2  unknown   empty       unconfigured unknown
   pcie3  unknown   connected configured unknown
   pcie4  unknown   empty       unconfigured unknown
   pcie5 pci-pci/hp connected configured ok
   pcie6  unknown   disconnected unconfigured unknown
   ```
7 Configure the PCI adapter card manually by using the `cfgadm` command as shown below. If using the auto-configuration method, this step should not be necessary. Refer to your platform guide for more information.

For example:

```
# cfgadm -c configure pcie3
```

8 Verify the configuration of the PCI adapter card in the slot.

For example:

```
# cfgadm pci
```

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Type</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>pcie1</td>
<td>unknown</td>
<td>empty</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
<tr>
<td>pcie2</td>
<td>unknown</td>
<td>empty</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
<tr>
<td>pcie3</td>
<td>etherne/hp</td>
<td>connected</td>
<td>configured</td>
<td>unknown</td>
</tr>
<tr>
<td>pcie5</td>
<td>pci-pci/hp</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>pcie6</td>
<td>unknown</td>
<td>disconnected</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
</tbody>
</table>

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. For example:

```
# ifconfig plumb bge1
```

**Note** - The auto-configuration method can be enabled or disabled at boot, depending on the platform implementation. Set the auto-configuration method as appropriate for your environment.

### 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.
SATA Hot-Plugging With the `cfgadm` Command

SATA controller and port multiplier device ports are represented by attachment points in the device tree. SATA devices that are connected and configured on the system are shown as attachment point name extension. The terms attachment point and SATA port can be used interchangeably.

The `cfgadm` syntax that is used with SATA devices is slightly different than `cfgadm` syntax for SCSI or PCI devices.

You can display SATA device information as follows:

```bash
% cfgadm -al
Ap_Id  Type     Receptacle Occupant Condition
sata0/0::dsk/c7t0d0 disk connected configured ok
sata0/1::dsk/c7t1d0 disk connected configured ok
sata0/2::dsk/c7t2d0 disk connected configured ok
sata0/3::dsk/c7t3d0 disk connected configured ok
```

### How to Unconfigure a SATA Device

In general, SATA devices must be unconfigured before they can be removed and replaced. If you attempt to unconfigure a device that is part of an active ZFS storage pool, you will see an error message similar to the following:

```
# cfgadm -c unconfigure sata5/7
Unconfigure the device at: /devices/pci@2,0/pci1022...
This operation will suspend activity on the SATA device
Continue (yes/no)? y
cfgadm: Hardware specific failure: Failed to unconfig device at ap_id: /devices/pci@2,0/pci10...
```

1. Become superuser.

2. Identify the device to be unconfigured.

   ```bash
   # cfgadm -al | grep c7t0d0
   sata0/0::dsk/c7t0d0  disk connected configured ok
   ```

3. Unconfigure the device.

   ```bash
   # cfgadm -c unconfigure sata0/0
   ```
If you attempt to unconfigure the device by specifying the individual device, you will see a message similar to the following:

```
# cfgadm -c unconfigure sata0/0::dsk/c7t0d0
do_control_ioctl: open failed: errno:2
cfgadm: Library error: Cannot open ap_id: /devices/pci@0,0/pci10...
No such file or directory
```

4 Confirm that the device is unconfigured.
```
# cfgadm | grep sata0/0
sata0/0 disk connected unconfigured ok
```

▼ How to Configure a SATA Device

After a disk is physically removed or replaced, it can be configured.

1 Become superuser.

2 Configure the device.
```
# cfgadm -c configure sata0/0
```

3 Confirm that the device is configured.
```
# cfgadm | grep sata0/0
sata0/0::dsk/c7t0d0 disk connected configured ok
```

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 dynamically remove the resource. Or, you can use the `cfgadm` command with the `-f` option to force a reconfiguration operation. However, 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 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 an RCM script as follows:

```
$ script-name command [args ...]
```

An RCM 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 `rcmsh(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` and `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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Identify the resources your application uses.</td>
<td>Identify the resources (device names) your application uses that you could potentially dynamically remove.</td>
<td><code>cfgadm(1M)</code></td>
</tr>
</tbody>
</table>
## RCM Script Tasks

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>Identify the commands to release the resource.</td>
<td>Identify the commands for notifying the application to cleanly release the resource from the application.</td>
</tr>
<tr>
<td>3.</td>
<td>Identify the commands for post-removal of the resource.</td>
<td>Include the commands for notifying the application of the resource removal.</td>
</tr>
<tr>
<td>4.</td>
<td>Identify the commands if the resource removal fails.</td>
<td>Include the commands for notifying the application of the available resource.</td>
</tr>
<tr>
<td>5.</td>
<td>Write the RCM script.</td>
<td>Write the RCM script based on the information identified in tasks 1-4.</td>
</tr>
<tr>
<td>6.</td>
<td>Install the RCM script.</td>
<td>Add the script to the appropriate script directory.</td>
</tr>
<tr>
<td>7.</td>
<td>Test the RCM script</td>
<td>Test the script by running the script commands manually and by initiating a dynamic reconfiguration operation.</td>
</tr>
</tbody>
</table>

## 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.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Identify the resources to be dynamically removed.</td>
<td>Identify the resources (device names) to be potentially removed by using the cfgadm -l command.</td>
</tr>
<tr>
<td>2.</td>
<td>Identify the applications to be stopped.</td>
<td>Identify the commands for stopping the applications cleanly.</td>
</tr>
<tr>
<td>3.</td>
<td>Identify the commands for pre-removal and post-removal of the resource.</td>
<td>Identify the actions to be taken before and after the resource is removed.</td>
</tr>
<tr>
<td>4.</td>
<td>Write the RCM script.</td>
<td>Write the RCM script based on the information identified in tasks 1-3.</td>
</tr>
<tr>
<td>5.</td>
<td>Install the RCM script.</td>
<td>Add the script to the appropriate script directory.</td>
</tr>
</tbody>
</table>
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 96

### 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 an administrator to install or remove an RCM script. Use this table to determine where you should install your RCM script.

<table>
<thead>
<tr>
<th>Directory Location</th>
<th>Script Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/etc/rcm/scripts</code></td>
<td>Scripts for specific systems</td>
</tr>
<tr>
<td><code>/usr/platform/uname -i</code>/<code>lib/rcm/scripts</code></td>
<td>Scripts for a specific hardware implementation</td>
</tr>
<tr>
<td><code>/usr/platform/uname -m</code>/<code>lib/rcm/scripts</code></td>
<td>Scripts for a specific hardware class</td>
</tr>
<tr>
<td><code>/usr/lib/rcm/scripts</code></td>
<td>Scripts for any hardware</td>
</tr>
</tbody>
</table>

**How to Install an RCM Script**

1. Become an administrator.

2. Copy the script to the appropriate directory. See Table 4–1.

   For example:
   ```
   # cp ABC,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/ABC,sample.pl
   ```
4 Send SIGHUP to the RCM daemon.
   # pkill -HUP -x -u root rcm_daemon

▼ How to Remove an RCM Script

1 Become an administrator.

2 Remove the script from the RCM script directory.
   For example:
   # rm /usr/lib/rcm/scripts/ABC,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 that 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 95.

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 cl::dsk/clt0d0
   $ cfgadm -f -c unconfigure cl::dsk/clt0d0
   $ cfgadm -c configure cl::dsk/clt0d0
Caution – Make sure that you are familiar with these commands because they can alter the state of the system and 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, the script 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

```perl
#!/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_queryremove,
  "preremove" => \&do_preremove
};

if (defined($dispatch($cmd))) {
  \$dispatch($cmd);
} else {
```
RCM Script Tasks

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);
#}
#
Chapter 4 • Dynamically Configuring Devices (Tasks)

99


# 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);
}
Managing USB Devices (Tasks)

This chapter provides overview information and step-by-step instructions for using USB devices in the Oracle Solaris OS. This chapter specifically covers how to use USB devices with HAL services.

The following is a list of the information in this chapter:

- “What’s New in USB Devices?” on page 101
- “About USB Support in Oracle Solaris” on page 102
- “Overview of USB Devices” on page 107
- “Managing USB Mass Storage Devices” on page 111
- “Using USB Audio Devices” on page 127
- “Hot-Plugging USB Devices With the cfgadm Command” on page 131


For general information about dynamic reconfiguration and hot-plugging, see Chapter 4, “Dynamically Configuring Devices (Tasks).”

For information on configuring USB printers, see System Administration Guide: Printing.

What’s New in USB Devices?

The following features are new or changed in this release.

USB 3.0 Support

USB 3.0 support is provided by the introduction of a new USB host controller driver, xhci, the enhancement of the hub driver, and the Oracle Solaris USB Architecture (USBA) framework. USB 3.0 hub and mass-storage devices can now work in USB 3.0 mode, when they are inserted.
into USB 3.0 ports. In addition, the legacy USB devices continue to work when they are connected to USB 3.0 ports, with the exception of USB audio devices. See “Supported USB Features” on page 102.

About USB Support in Oracle Solaris

The following information is described in this section:

- “Supported USB Features” on page 102
- “USB Device Features and Compatibility Issues” on page 103
- “Bus-Powered Devices” on page 104
- “USB Keyboards and Mouse Devices” on page 104
- “USB Host Controller and Hubs” on page 105
- “USB Hub Devices” on page 106
- “SPARC: USB Power Management” on page 106
- “Guidelines for Connecting USB Cables” on page 107

Supported USB Features

This Oracle Solaris release includes expanded supported for USB 3.0 devices (“SuperSpeed” devices) that use the USB 3.0 xhci host controller driver. The xhci host controller driver supports all USB devices, with the exception of audio devices. The xhci host controller driver supports interrupt, control, and bulk transfers for USB hosts, including super speed and non-super speed bus interfaces. Compatibility with USB 2.0, 1.1, and 1.0 devices and drivers is supported so that you can use the same cables, connectors, and software interfaces.

Support for the following USB devices is provided on both SPARC and x86 based systems, except where indicated:

- USB 3.0 devices work in USB 3.0 mode when they are inserted into USB 3.0 ports. Legacy USB devices (2.0, 1.1, and 1.0) continue to work when they are connected to USB 3.0 ports, with the exception of USB audio devices.
- Audio device support is provided for USB 2.0, 1.1, and 1.0 audio devices only.
  Devices that are not supported by a USB driver might include libusb applications, such as gphoto2, gtkam, and pilotlink. For more information, refer to the user-space USB library documentation in /usr/share/doc/libusb/libusb.txt.
- Generic USB driver support – See ugen(7D).
- Human Interface Device (HID) support (keyboard and mouse devices) – See hid(7D))
- Hub support – See hubd(7D).
- Printer support
- USB CDC-ECM (USB Communication Device Class- Ethernet Control Model) device support
Serial device support, including the following:

- Edgeport – See usbser_edge(7D).
- Prolific – See usbprl(7D).
- Keyspan – See usbsksp(7D).
- Storage device support – See scsa2usb(7D).
- User-space USB device management library support. See libusb(3LIB).

Increased USB 3.0 bus speed from 480 Mbits/sec to 5 Gbits/sec.

A USB 3.0 or USB 2.0 port might be one of the following possibilities:

- A port on a USB 3.0 or USB 2.0 PCI card
- A port on a USB 3.0 or USB 2.0 hub that is connected to USB 3.0 or USB 2.0 port
- A port on a SPARC or x86 computer motherboard

**Note** – A USB 2.0 PCI card might be needed for older SPARC platforms.

USB 2.0 and USB 1.1 devices work as they have in the past, even if you have USB 3.0, USB 2.0, and USB 1.0 devices on the same system. A USB 2.0, and a USB 1.0 host controller supports all four transfer types: control, bulk, interrupt, and isochronous. A USB 3.0 host controller supports three transfer types: control, bulk, and interrupt. The isochronous transfer type is not supported under a USB 3.0 host controller.

While USB 3.0 and USB 2.0 devices can operate on a USB 1.1 or 1.0 port, their performance is significantly better when they are connected to a USB 2.0 or a USB 3.0 port.

An XHCI host controller does not have a companion controllers, meaning a single controller supports all speeds of the USB devices. A USB 2.0 host controller has one high-speed EHCI and one or more OHCI or UHCI embedded controllers. Devices that are connected to a USB 2.0 port are dynamically assigned to either a EHCI or OHCI controller, depending on whether they support USB 2.0.

For more information about USB 3.0 device support, see xhci(7D).

For more information on USB 2.0 device support, see ehci(7D) and usba(7D).

For more information about USB 1.1 device support, see ohci(7D).

For a description of USB devices and terminology, see "Overview of USB Devices" on page 107.

**USB Device Features and Compatibility Issues**

USB 3.0 devices are defined as super speed ("SuperSpeed") devices that follow the USB 3.0 specification. USB 2.0 devices are defined as high-speed ("Hi-Speed") devices that follow the USB 2.0 specification. You can refer to the USB 3.0 and the USB 2.0 specifications at http://www.usb.org/home.
To identify the speed of your USB device, check the `/var/adm/messages` file for messages similar to the following:

```
Dec 13 17:05:57 mysystem usba: [ID 912658 kern.info] USB 2.0 device (usb50d,249) operating at hi speed (USB 2.x) on USB 2.0 external hub: storage@4, scsaa2usb0 at bus address 4
```

Here are some of the USB devices that are supported in this Oracle Solaris release:

- Mass storage devices, such as CD-RWs, hard disks, DVDs, digital cameras, diskettes, tape drives, memory sticks, and multi-format card readers
- Keyboards and mouse devices
- Audio devices, such as speakers and microphones

**Note** – Audio device support is not available in USB 3.0.

Additional storage devices might work by modifying the `scsaa2usb.conf` file. For more information, see `scsaa2usb(7D)`.

### Bus-Powered Devices

Bus-powered hubs use power from the USB bus to which they are connected, to power devices connected to them. Special care must be taken to not overload these hubs, because the power these hubs offer to their downstream devices is limited.

Power budgeting is implemented for USB devices. This feature has the following limitations:

- Cascading two bus-powered hubs is not recommended.
- Each bus-powered hub is allowed a maximum of 100 mA only for each port.
- Only self-powered or low bus-powered devices are allowed to connect to a bus-powered hub. High bus-powered devices are denied the connection. Some hubs or devices can report a false power source, such that the connection might be unpredictable.

### USB Keyboards and Mouse Devices

Keep the following issues in mind when using USB keyboards and mouse devices:

- *Do not* move the keyboard and mouse during a reboot or at the `ok` prompt on a SPARC system. You can move the keyboard and mouse to another hub at any time after a system reboot. After you plug in a keyboard and mouse, they are fully functional again.
- The keys just to the left of the keypad might not function on some third-party USB keyboards.
SPARC – Keep the following issues in mind when using USB keyboards and mouse devices on SPARC systems:

- The power key on a USB keyboard behaves differently than the power key on the type 5 keyboard. On a USB keyboard, you can suspend or shut down the system by using the SUSPEND/SHUTDOWN key. However, you cannot use that key to power up the system.
- Before the boot process finishes, the OpenBoot PROM (OBP) limits keyboard and mouse devices to the motherboard root hub ports only.
- USB keyboard and mouse devices cannot be used simultaneously with Type 3, 4, or 5 keyboards on legacy SPARC systems.
- For information about multiple keyboard and mouse device support, see virtualkm(7D).

USB Wheel Mouse Support

The following wheel mouse features are supported:

- Support for more than 3 buttons is available on USB or PS/2 mouse devices.
- Wheel mouse scrolling is available on a USB or PS/2 mouse device. This support means that rolling the wheel on a USB or a PS/2 mouse results in a scroll in the application or window under mouse focus. StarOffice, Firefox, and GNOME applications support wheel mouse scrolling. However, other applications might not support this functionality.

USB Host Controller and Hubs

A USB hub is responsible for the following tasks:

- 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 tasks:

- 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

Keeping the following key points in mind when using USB hub devices:

- Do not cascade hubs beyond four levels on either SPARC based systems or x86 based systems. On SPARC systems, the OpenBoot PROM cannot reliably probe beyond four levels of devices.
- Do not plug a bus-powered hub into another bus-powered hub in a cascading style. A bus-powered hub does not have its own power supply.
- Do not connect a device that requires a large amount of power to a bus-powered hub. These devices might be denied connection to bus-powered hubs or might drain the hub of power for other devices. An example of such a device is a USB diskette device.

SPARC: USB Power Management

Suspending and resuming USB devices is fully supported on SPARC systems. However, do not suspend a device that is busy and never remove a device when the system is powered off under a suspend shutdown.

The USB framework makes a best effort to power manage all devices on SPARC based systems with power management enabled. Power managing a USB device means that the hub driver suspends the port to which the device is connected. Devices that support remote wake up can notify the system to wake up everything in the device’s path so that the device can be used. The host system could also wake up the device if an application sends an I/O to the device.

All HID devices (keyboard, mouse, hub, and storage devices), hub devices, and storage devices are power managed by default if they support remote wake-up capability. A USB printer is power managed only between two print jobs. Devices that are managed by the generic USB driver (UGEN) are power managed only when they are closed.

When power management is running to reduce power consumption, USB leaf devices are powered down first. After all devices that are connected to a 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.

For information about using the SUSPEND/SHUTDOWN key on SPARC systems, see “USB Keyboards and Mouse Devices” on page 104.
Guidelines for Connecting USB Cables

Keep the following guidelines in mind when connecting USB cables:

- USB 3.0 specification does not specify a maximum cable length. However, a USB 3.0 cable is distinctly different from a USB 2.0 cable. It is estimated that shorter cables achieve better performance and three meter cables are recommended to achieve multi-gigabit transfer rates.
- For USB 2.0 devices, always use compliant, fully rated (480 Mbit/sec) 20/28 AWG cables for connecting these devices.
- The maximum cable length that is supported is 5 meters.
- Do not use cable extenders. For best results, use a self-powered hub to extend cable length.

For more information, go to http://www.usb.org/about/faq.

Overview of USB Devices

The following overview information is provided in this section:

- “Commonly Used USB Acronyms” on page 107
- “Oracle Solaris USB Architecture” on page 108
- “USB Bus Description” on page 109
- “USB Devices and Driver Classes” on page 110

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, in one way. 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.

Devices connect to USB ports on external USB hubs, or on a root hub that is located on the computer itself. Since hubs have several ports, several branches of a device tree can stem from a hub.

For more information, see usba(7D) or go to http://www.usb.org/home.

Commonly Used USB Acronyms

The following table describes the USB acronyms that are used in the Oracle Solaris OS. For a complete description of USB components and acronyms, go to http://www.usb.org/home.
Overview of USB Devices

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>UGEN</td>
<td>USB generic driver</td>
<td>ugen(7D)</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
<td>usb(7D)</td>
</tr>
<tr>
<td>USBA</td>
<td>Universal Serial Bus Architecture (Solaris)</td>
<td>usba(7D)</td>
</tr>
<tr>
<td>USBAI</td>
<td>USBA Client Driver Interface (Solaris)</td>
<td>N/A</td>
</tr>
<tr>
<td>HCD</td>
<td>USB host controller driver</td>
<td>N/A</td>
</tr>
<tr>
<td>EHCI</td>
<td>Enhanced Host Controller Interface</td>
<td>ehci(7D)</td>
</tr>
<tr>
<td>OHCI</td>
<td>Open Host Controller Interface</td>
<td>ohci(7D)</td>
</tr>
<tr>
<td>UHCI</td>
<td>Universal Host Controller Interface</td>
<td>uhci(7D)</td>
</tr>
<tr>
<td>XHCI</td>
<td>Extensible Host Controller Interface</td>
<td>xhci(7D)</td>
</tr>
</tbody>
</table>

Oracle Solaris USB Architecture

USB devices can be 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.

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 can bind to just one interface. 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 IEEE 1275 binding specification.

The Oracle Solaris USB Architecture (USBA) adheres to the USB 1.1, USB 2.0, and USB 3.0 specifications and is part of the Oracle Solaris Device Driver Interface (DDI). The USBA model is similar to Oracle Common SCSI Architecture (SCSA). As the following figure shows, the USBA is a thin layer that provides a generic USB transport-layer abstraction to client drivers, providing them with services that implement core generic USB functionality.
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 Hubs” on page 105.
Figure 5–2 shows a system with three active USB ports. The first USB port connects a USB memory stick. The second USB port connects an external hub, which in turn, connects a cdrw device and a composite keyboard/mouse device. As a composite device, this keyboard contains a USB controller, which operates both the keyboard and an attached mouse. The keyboard and the mouse share a common USB bus address because they are directed by the same USB controller.

Figure 5–2 also shows an example of a hub and a printer as a compound device. The hub is an external hub that is enclosed in the same casing as the printer. The printer is permanently connected to the hub. The hub and printer have separate USB bus addresses.

The device tree path name for some of the devices that are displayed in Figure 5–2 are listed here.

Memory stick /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
cdrw device /pci@1f,4000/usb@5/hub@2/storage@3
Printer /pci@1f,4000/usb@5/hub@3/printer@1

**USB Devices and Driver Classes**

USB devices with similar attributes and services are grouped into device classes. Each device class has a corresponding driver. Devices within a class are managed by the same device driver pair. However, the USB specification also allows for vendor-specific devices that are not part of a specific class.

The HID class contains devices that are user-controlled, such as the following devices:

- Keyboards
- Mouse devices
- Joysticks

The Communication Device class includes the following devices:

- Modems
- Ethernet adapters

Other device classes include the following classes:

- Audio
- Monitor
- Printer
- Storage Device
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 by going to http://www.usb.org/home/.

For more information about USB devices supported in the Oracle Solaris release, see usb(7D).

Managing USB Mass Storage Devices

The following tasks are described in this section:

- “Using USB Diskette Devices” on page 112
- “Hot-Plugging USB Mass Storage Devices” on page 113
- “Preparing to Use a USB Mass Storage Device” on page 116
- “How to Display USB Device Information” on page 116
- “How to Create a File System on a USB Mass Storage Device” on page 117
- “How to Modify Partitions and Create a PCFS File System on a USB Mass Storage Device” on page 119
- “How to Create a Solaris Partition and Modify the Slices on a USB Mass Storage Device” on page 122
- “How to Mount or Unmount a USB Mass Storage Device” on page 124
- “Troubleshooting Tips for USB Mass Storage Devices” on page 126
- “Disabling Specific USB Drivers” on page 126
- “How to Disable Specific USB Drivers” on page 127
- “How to Remove Unused USB Device Links” on page 127

The following USB removable mass storage devices are supported:

- CD-RWs
- Hard disks
- DVDs
- Digital cameras
- Diskette devices
- SmartMedia and CompactFlash devices

In previous Oracle Solaris releases, all USB storage devices were identified as removable media devices, which provides many of the following advantages, including automatic mounting. In the release, USB mass storage devices are identified as hot-pluggable devices but also enjoy the advantages of USB removable devices that are described in the following sections. For more information about the hot-pluggable behavior, see “Hot-Plugging USB Mass Storage Devices” on page 113.
Guidelines for managing USB mass storage device are as follows:

- A hot-pluggable device is automatically mounted.
- USB storage devices with standard MS-DOS or Windows (FAT) file systems are supported.
- You can use the user-friendly \texttt{rmformat} command to create slices. You can also use the \texttt{fdisk} command to partition a USB device, but never use the \texttt{format} utility or the \texttt{rmformat -F} command to physically format a USB drive.
- Use the \texttt{rmformat} command to display all USB devices with media inserted. For example, see “How to Display USB Device Information” on page 116.
- Non-root users can now access USB storage devices, since the \texttt{mount} command is no longer needed. The device is automatically mounted and is available under the \texttt{/rmdisk} directory.
- These devices can be managed with or without removable media services.
- Disks with FAT file systems can be mounted and accessed. For example:
  
  \begin{verbatim}
  mount -F pcfs /dev/dsk/c2t0d0s0:c /mnt
  \end{verbatim}

- All USB storage devices are now power managed, except for those that support LOG SENSE pages. Devices with LOG SENSE pages are usually SCSI drives connected through a USB-to-SCSI bridge device.
- Applications might work differently with USB mass storage devices. Keep the following issues in mind when using applications with USB storage devices:
  - Applications might make incorrect assumptions about the size of the media since only smaller devices like diskettes were removable previously.
  - Requests by applications to eject media on devices where this would be inapplicable, such as a hard drive, will succeed and do nothing.
  - If you prefer the behavior in previous Oracle Solaris releases where all USB mass storage were treated as removable media devices, then you can force the old behavior by updating the \texttt{/etc/driver/drv/scsa2usb.conf} file.

For more information on using USB mass storage devices, see \texttt{scsa2usb(7D)}.

### Using USB Diskette Devices

USB diskette devices appear as removable media devices. USB diskette devices are not managed by the \texttt{fd} (floppy) driver. Applications that issue \texttt{ioctl(2)} calls intended for the \texttt{fd} (native floppy) driver will fail. Applications that issue only \texttt{read(2)} and \texttt{write(2)} calls will succeed. Other applications, such as SunPCI and \texttt{rmformat}, also succeed.

The USB diskette device is identified as a SCSI removable media device. The device is available for access under the \texttt{/rmdisk} directory.
For more information on how to use USB diskette devices, see “Overview of USB Devices” on page 107.

**Hot-Plugging USB Mass Storage Devices**

Hot-plugging a device means the device is added or removed without shutting down the operating system or powering off the system. All USB devices are hot-pluggable.

The *hot-pluggable* device attribute identifies those devices that can be connected or disconnected without rebooting the system and configured or unconfigured automatically without user intervention. All USB devices are identified as hot-pluggable devices to gain those benefits. In addition, non-removable media USB devices are no longer identified as removable-media devices and no longer have a *removable-media* attribute.

Non-removable USB storage devices are identified as hot-pluggable devices at the driver level. This behavior means that these devices can be connected or disconnected without rebooting the system and configured or unconfigured automatically without intervention. These changes are made at the kernel level and do not impact the use of these devices. For example, the responsibility of mounting and unmounting these devices is controlled by the removable media management services.

For more information about using these devices, see *scsa2usb*(7D).

The removable media manager is now aware of hot-plugged devices. You can just plug in the device, which is mounted in a few seconds. If nothing happens, check to see if it is mounted.

Make sure that removable media services are running.

```
# svc -a volfs
STATE STIME FMRI
online Sep 11 svc:/system/filesystem/volfs:default
```

The file system can be mounted from the device if it is valid and it is recognized.

If it fails to mount, stop *vold*.

```
# svcadm disable volfs
```

If the file system on the device is not automatically mounted, try a manual mount.

Before hot-removing the device, find the name of the device in the *eject* -l command’s alias name. Then eject the device’s media. If you don’t do this, the device is released and the port is usable again, but the file system on the device might have been damaged.
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 you are using the device.

If you are using a device 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 returns an error.

In this situation, the system prompts you to plug in the original device. If the device is no longer available, stop the applications. After a few seconds, the port becomes available again.

**Note** – Data integrity might be impaired if you remove an active or open device. Always close the device before removing, except the attached keyboard and mouse, which can be moved while active.

### How to Add a USB Mass Storage Device

1. **Become superuser.**
2. **Connect the USB mass storage device.**
3. **Verify that the USB device has been added.**
   
   For example:
   ```bash
   $ rmformat
   Looking for devices...
   1. Logical Node: /dev/rdsk/c3t0d0p0
      Physical Node: /pci@0,0/pci108e,534a@2,1/storage@3/disk@0,0
      Connected Device: SanDisk Cruzer Micro 0.3
      Device Type: Removable
      Bus: USB
      Size: 245.0 MB
      Label: <None>
      Access permissions: Medium is not write protected.
   ```
4. **Verify that the device is automatically mounted under the `/rmdisk` directory.**
   
   For example:
   ```bash
   $ ls /rmdisk/noname
   aa  bb
   ```

### How to Add a USB Camera

If the camera’s media uses a PCFS file system, it will be automatically mounted. If the device does not bind to the `scsa2usb` driver, use `libusb` applications for transferring the pictures. For more information, refer to the `/usr/share/doc/libusb/libusb.txt` file.
1 **Plug in and turn on the USB camera.**

The system creates a logical device for the camera. After the camera is plugged in, output is written to the `/var/adm/messages` file to acknowledge the device's connection. The system treats the camera as a storage device.

2 **Examine the output that is written to the `/var/adm/messages` file.**

   

   ```bash
   # more /var/adm/messages
   
   Examining this output enables you to determine which logical device was created so that you can then use that device to access your images. The output looks similar to the following:
   
   Jul 15 09:53:35 buffy usba: [ID 349649 kern.info] OLYMPUS, C-3040ZOOM, 00153719068
   Jul 15 09:53:35 buffy generic: [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
   
   Match the device with a mountable `/dev/dsk` link entry, by doing the following:
   
   ```bash
   # ls -l /dev/dsk/c*t | grep /pci@0,0/pci925,1234@7,2/storage@2
   lrwxrwxrwx 1 root root 58 Jun 14 2010 c3t0d0p0 -> ../../devices/pci@0,0/pci925,1234@7,2/storage@2/disk@0,0:a
   
   Mount the USB camera filesystem.
   The camera's filesystem is most likely a PCFS filesystem. If file system is PCFS, then it should be automatically mounted.

   - To manually mount the file system on an x86 system, you would use syntax similar to the following:
     ```bash
     # mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
     ```

   - To manually mount the file system on a SPARC system, you would use syntax similar to the following:
     ```bash
     # mount -F pcfs /dev/dsk/c3t0d0s0:c /mnt
     ```

   For information on mounting file systems, see “Mounting and Unmounting Oracle Solaris File Systems” on page 286.

   For information on mounting different PCFS file systems, see `mount_pcfs(1M)`.

3 **Mount the USB camera file system.**

4 **(Optional) Verify that the image files are available.**

   For example:

   ```bash
   # ls /mnt/DCIM/100OLYMP/
   P7220001.JPG* P7220003.JPG* P7220005.JPG*
   P7220002.JPG* P7220004.JPG* P7220006.JPG*
   ```
5 (Optional) View and manipulate the image files created by the USB camera.
   For example:
   
   # /usr/dt/bin/sdtimage P7220001.JPG &

6 Unmount the file system before disconnecting the camera.
   For example:
   
   # umount /mnt

7 (Optional) Turn off and disconnect the camera.

Preparing to Use a USB Mass Storage Device

You can access information on removable media with or without using removable media services. For information on accessing information on removable media with GNOME’s File Manager, see the GNOME desktop documentation.

After the USB device is formatted, it is usually mounted under the /rmdisk/label directory.

The device nodes are created under the /dev/rdsk directory for character devices and under the /dev/dsk directory for block devices. Device links are created when the devices are hot-plugged. For more information, see scsa2usb(7D).

If the device cannot be identified by removable media services, then try to manually mount the device. First, use the rmformat command to identify the device path and then use the mount command to manually mount the device as an administrator.

If a device is mounted by removable media services then you can unmount it by using the rmumount command. If the device is manually mounted, then you would unmount it using the umount command as an administrator.

For more information about mounting and unmounting USB devices, see “How to Mount or Unmount a USB Mass Storage Device” on page 124.

▼ How to Display USB Device Information

Display information about USB devices.

The prtconf output in this example has been truncated to only display USB device information.

$ 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 `rmformat` command to display USB storage device information:

```
$ rmformat
Looking for devices...
  1. Logical Node: /dev/rdsk/c3t0d0p0
     Physical Node: /pci@0,0/pci108e,534a@2,1/storage@3/disk@0,0
     Connected Device: SanDisk Cruzer Micro 0.3
     Device Type: Removable
     Bus: USB
     Size: 245.0 MB
     Label: <None>
     Access permissions: Medium is not write protected.
```

▼ **How to Create a File System on a USB Mass Storage Device**

A USB diskette must be formatted before you can add a file system to it. All other USB mass storage devices just need a file system before they can be used.

Keep the following key points in mind when formatting a USB device:

- Do not use the `rmformat -F` except on a USB diskette.

- If the default slices are not acceptable, use the `rmformat -s` command to create slices. Use the `fdisk` utility to partition a USB device, if needed. For step-by-step instructions, see:
  - "How to Modify Partitions and Create a PCFS File System on a USB Mass Storage Device" on page 119
  - "How to Create a Solaris Partition and Modify the Slices on a USB Mass Storage Device" on page 122

- If the USB device is automatically mounted, you will have to unmount it before you can create a file system on the USB device.
Note – Perform Steps 4-5 only if you need to format a USB diskette.

1  **Become superuser.**

2  **Add the USB device to your system.** For information on hot-plugging USB devices, see:
   - “Hot-Plugging USB Mass Storage Devices” on page 113
   - “Hot-Plugging USB Devices With the \texttt{cfgadm} Command” on page 131

3  **(Optional) Identify the USB device.**
   
   ```
   # rmformat
   Looking for devices...
   1. Logical Node: /dev/rdsk/c2t0d0p0
      Physical Node: /pci@0,0/pci108e,534a@2,1/hub@7/floppy@1/disk@0,0
      Connected Device: MITSUMI USB FDD 1039
      Device Type: Floppy drive
      Bus: USB
      Size: 1.4 MB
      Label: <None>
      Access permissions: Medium is not write protected.
   ```

   In this example, the diskette device is `c2t0d0p0`.

4  **Insert a diskette into the diskette drive, if necessary.**

5  **Format the diskette, if necessary.**
   
   ```
   # rmformat -F long raw-device
   ```

6  **Determine the file system type and make sure the device is unmounted.** Then, select one of the following:

   For more information about unmounting a USB device, see “How to Mount or Unmount a USB Mass Storage Device” on page 124.

   - Create a ZFS pool and file system on a USB stick if you need to transfer data to another system.
     ```
     # zpool create c5t0d0 temp-pool
     # zfs create temp-pool/data
     ```

   - Create a PCFS file system.
     ```
     # mkfs -F pcfs -onofdisk,size=size raw-device
     ```

     Specify the -size option in 512-byte blocks.

     The following example shows how to create a PCFS file system on a 1.4-MB diskette on a SPARC system:
# mkfs -F pcfs /dev/rdsk/c2t0d0p0
Construct a new FAT file system on /dev/rdsk/c2t0d0p0: (y/n)? y

The following example shows how to create a PCFS file system on a 1.4-MB diskette on an x86 system:

# mkfs -F pcfs /dev/rdsk/c2t0d0s2
Construct a new FAT file system on /dev/rdsk/c2t0d0s2: (y/n)? y

The following example shows how to create a PCFS file system on a 100-MB USB memory stick on a SPARC system:

# mkfs -F pcfs /dev/rdsk/c5t0d0s2:c

The following example shows how to create a PCFS file system on a 100-MB USB memory stick on an x86 system:

# mkfs -F pcfs /dev/rdsk/c5t0d0p0:c

This command can take several minutes to complete.

- Create a legacy UFS file system.
  
  # newfs raw-device

  Consider using the newfs -f 4096 option or newfs -T option for large USB hard disks.

---

**Note** – UFS file system overhead consumes a significant portion of space on a diskette, due to a diskette’s limited storage capacity.

See “How to Modify Partitions and Create a PCFS File System on a USB Mass Storage Device” on page 119 and “How to Create a Solaris Partition and Modify the Slices on a USB Mass Storage Device” on page 122 for detailed examples of creating a PCFS file system and modifying slices on a USB mass storage device.

▼ **How to Modify Partitions and Create a PCFS File System on a USB Mass Storage Device**

The following steps describe how to delete an existing partition, create a new partition, and then create a PCFS file system on the USB device. Make sure you backup any data before you perform this task. Note that this procedure includes specific examples to serve as a guideline only. Information that you provide should apply to your particular system.

1  **Become superuser.**

2  **Start the fdisk utility.**
   
   # fdisk /dev/rdsk/c3t0d0p0
3 Delete the partition by selecting option 3.

Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

<table>
<thead>
<tr>
<th>Cylinders</th>
<th>Partition</th>
<th>Status</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Active</td>
<td>Solaris2</td>
<td>1</td>
<td>28</td>
<td>28</td>
<td>97</td>
<td></td>
</tr>
</tbody>
</table>

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Enter Selection: 3

4 Choose the partition number to delete.

Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

<table>
<thead>
<tr>
<th>Cylinders</th>
<th>Partition</th>
<th>Status</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Active</td>
<td>Solaris2</td>
<td>1</td>
<td>28</td>
<td>28</td>
<td>97</td>
<td></td>
</tr>
</tbody>
</table>

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Specify the partition number to delete (or enter 0 to exit): 1

Partition deleted.

5 Create a partition.

Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

<table>
<thead>
<tr>
<th>Cylinders</th>
<th>Partition</th>
<th>Status</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
<th>%</th>
</tr>
</thead>
</table>

WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)
Enter Selection: 1

### Select the FAT32 partition type.

Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

<table>
<thead>
<tr>
<th>Cylinder</th>
<th>Partition Status Type</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
<th>%</th>
</tr>
</thead>
</table>

**WARNING:** no partitions are defined!

**SELECT ONE OF THE FOLLOWING:**

1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Select the partition type to create:
- 1=SOLARIS2
- 2=UNIX
- 3=PCIXOS
- 4=Other
- 5=DOS12
- 6=DOS16
- 7=DOSEXT
- 8=DOSBIG
- 9=DOS16LBA
- A=x86 Boot
- B=Diagnostic
- C=FAT32
- D=FAT32LBA
- E=DOSEXTLBA
- F=EFI
- 0=Exit

7. Specify the percentage of disk to use for this partition.

Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

<table>
<thead>
<tr>
<th>Partition Status Type</th>
<th>Cylinders</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
<th>%</th>
</tr>
</thead>
</table>

**WARNING:** no partitions are defined!

**SELECT ONE OF THE FOLLOWING:**

1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Select the partition type to create:
Specify the percentage of disk to use for this partition (or type "c" to specify the size in cylinders). 100

8. Select whether the new partition should be the active partition or an inactive partition.

Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

<table>
<thead>
<tr>
<th>Partition Status Type</th>
<th>Cylinders</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
<th>%</th>
</tr>
</thead>
</table>
WARNING: no partitions are defined!

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Select the partition type to create:
Should this become the active partition? If yes, it will be activated each time the computer is reset or turned on.
Please type "y" or "n": n

9 Update the disk configuration and exit.
Total disk size is 29 cylinders
Cylinder size is 2048 (512 byte) blocks

<table>
<thead>
<tr>
<th>Partition</th>
<th>Status</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Win95 FAT32</td>
<td>1</td>
<td>28</td>
<td>28</td>
<td>97</td>
</tr>
</tbody>
</table>

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Exit (update disk configuration and exit)
6. Cancel (exit without updating disk configuration)

Enter Selection: 5

10 Create the PCFS file system on this partition.
Make sure the device is unmounted before creating the new file system. For more information about unmounting a USB device, see “How to Mount or Unmount a USB Mass Storage Device” on page 124.

```
# mkfs -F pcfs -o fat=32 /dev/rdsk/c3t0d0p0:c
```
Construct a new FAT file system on /dev/rdsk/c3t0d0p0:c: (y/n)? y

▼ How to Create a Solaris Partition and Modify the Slices on a USB Mass Storage Device

The following steps illustrate how to create a Solaris partition and modify the slices. Note that this procedure includes specific examples to serve as a guideline only. Information that you provide should apply to your particular system.

Make sure you back up any data before you perform this task.

1 Become superuser.
2 **Start the fdisk utility.**

```bash
# fdisk /dev/rdsk/c5t0d0s2
```
No fdisk table exists. The default partition for the disk is:

- a 100% "SOLARIS System" partition

Type 'y' to accept the default partition, otherwise type "n" to edit the partition table.

```
y
```

3 **Display the current slices.**

For example:

```bash
# prtvtoc /dev/rdsk/c5t0d0s2
```

```
* /dev/rdsk/c5t0d0s2 partition map
* Dimensions:
  * 512 bytes/sector
  * 63 sectors/track
  * 255 tracks/cylinder
  * 16065 sectors/cylinder
  * 5836 cylinders
  * 5836 accessible cylinders
* Flags:
  * 1: unmountable
  * 10: read-only
* Partition Tag Flags First Sector Last Sector Mount Directory
  0 0 0 0 93755340 93755339
  2 0 0 0 93755340 93755339
```

4 **Create a text file with the slice information.**

For example:

```
slices: 0 = 0, 5GB, "wm", "home":
  1 = 8225280000, 6GB :
  2 = 0, 44GB, "wm", "backup" :
  6 = 1645056000, 15GB
```

Make sure each slice starts on a cylinder boundary. For example, slice 1 starts at 822280000 bytes, which is the cylinder size in bytes multiplied by 1000.

For more information, see the `-s` option description in `rmformat(1)`.

5 **Create the slices by including the slice file created above.**

For example:

```bash
# rmformat -s slice_file /dev/rdsk/c5t0d0s2
```
6 View the new slice information.

For example:

```
# prtvtoc /dev/rdsk/c5t0d0s2
* /dev/rdsk/c5t0d0s2 partition map
*
* Dimensions:
*  512 bytes/sector
*  63 sectors/track
*  255 tracks/cylinder
*  16065 sectors/cylinder
*  5836 cylinders
*  5836 accessible cylinders
*
* Flags:
*  1: unmountable
* 10: read-only
*
* Unallocated space:
*  First  Sector  Last  Sector  Count  Sector
*  10485760  5579240  16064999
*  28647912  3482088  32129999
*  63587280  30168060  93755339
*
*  First  Sector  Last  Sector  Count  Sector  Mount Directory
  0   8   00   0  10485760  10485759
  1   3   01  12582912  28647911
  2   5   00  92274688  92274687
  6   4   00  31457280  63587279
```

▼ How to Mount or Unmount a USB Mass Storage Device

The following steps illustrate how to mount and unmount a USB mass storage device.

1 Become superuser.

2 (Optional) Identify the device.

For example:

```
$ rformat
Looking for devices...
  1. Logical Node: /dev/rdsk/c3t0d0p0
     Physical Node: /pci@0,0/pci108e,534a@2,1/storage@3/disk@0,0
     Connected Device: Sandisk Cruzer Micro 0.3
     Device Type: Removable
     Bus: USB
     Size: 245.0 MB
     Label: <None>
     Access permissions: Medium is not write protected.
```

In this example, the physical diskette device is c2t0d0p0.
3 Select one of the following to mount or unmount a USB mass storage device:

- Mount a USB mass storage device as a console user.
  You can use the `rmmount` command with device nicknames, mount points, or device paths, similar to the following:
  
  ```
  $ rmmount rmdisk0
  $ rmmount NONAME
  $ rmmount /dev/dsk/c3t0d0p0:1
  ```
  
  For example:
  
  ```
  $ rmmount NONAME
  NONAME /dev/dsk/c2t0d0p0 mounted
  $ ls /media/NONAME
  AA.TXT
  ```

- Unmount a USB mass storage device as a console user.
  For example:
  
  ```
  $ rmmount NONAME
  NONAME /dev/dsk/c2t0d0p0 unmounted
  ```

- Mount a USB mass storage device as superuser.
  This example shows how to mount a device with a UFS file system:
  
  ```
  $ mount /dev/dsk/c1t0d0s2 /mnt
  ```
  
  This example shows how to mount a device with a PCFS file system on a SPARC system:
  
  ```
  $ mount -F pcfs /dev/dsk/c3t0d0s2:c /mnt
  ```
  
  This example shows how to mount a device with a PCFS file system on an x86 system:
  
  ```
  $ mount -F pcfs /dev/dsk/c3t0d0p0:c /mnt
  ```
  
  This example shows how to mount a CD with a read-only HSFS file system:
  
  ```
  $ mount -F hsfs -o ro /dev/dsk/clt0d0s2 /mnt
  ```

- Unmount a USB mass storage device as superuser.
  First, be sure no one is using the file system on the device.
  For example:
  
  ```
  $ fuser -c -u /mnt
  $ umount /mnt
  ```

4 Eject the device, which is optional for DVD, CD, or diskette devices.

For example:

```
$ eject /dev/rdsk/c1t0d0s2
```
Troubleshooting Tips for USB Mass Storage Devices

Keep the following tips in mind if you have problems adding or removing a USB mass storage device.

Check the /var/adm/messages file for failures to enumerate the device. For enumeration failures, possibly, insert the USB hub or remove a hub and connect it directly to a root USB hub.

- If you have problems accessing a device that was connected while the system is running, try the following command:

  \# devfsadm

- Do not move devices around if the system has been powered down by a suspend operation. For more information, see "SPARC: USB Power Management" on page 106.

- If a device has been hot removed while in use by applications and is no longer available, then stop the applications. Use the prtconf command to see whether the device node has been removed.

Disabling Specific USB Drivers

You can disable specific types of USB devices by disabling their client driver. For example, USB printers can be disabled by disabling the usbpri driver that directs them. Disabling usbpri does not affect other kinds of devices, such as USB storage devices.

The following table identifies some USB device types and their corresponding drivers.

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Driver to Disable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>usb_ac and usb_as</td>
</tr>
<tr>
<td>HID (usually keyboard and mouse)</td>
<td>hid</td>
</tr>
<tr>
<td>Storage</td>
<td>scsa2usb</td>
</tr>
<tr>
<td>Printer</td>
<td>usbpri</td>
</tr>
<tr>
<td>Serial</td>
<td>usbser_edge</td>
</tr>
</tbody>
</table>

If you disable a driver for a USB device that is still connected to the system, you see a console message similar to the following:

usba10: WARNING: usba: no driver found for device name
How to Disable Specific USB Drivers

1 Become superuser.

2 Exclude the driver alias entry from the /etc/system file.
   For example, include the following exclude statement to exclude the usbprn driver:
   `exclude: usbprn`

3 Reboot the system.
   `# init 6`

How to Remove Unused USB Device Links

Use this procedure if a USB device is removed while the system is powered off. Removing the USB device while the system is powered off can leave device links for devices that do not exist.

1 Become superuser.

2 Close all applications that might be accessing the device.

3 Remove the unused links for a specific USB class.
   For example:
   `# devfsadm -C -c audio`
   Or, just remove the dangling links:
   `# devfsadm -C`

Using USB Audio Devices

The following tasks are provided in this section:

- “Hot-Plugging Multiple USB Audio Devices” on page 128
- “How to Add USB Audio Devices” on page 129
- “How to Identify Your System’s Primary Audio Device” on page 129
- “How to Change the Primary USB Audio Device” on page 130
- “Troubleshooting USB Audio Device Problems” on page 130

Note – USB 3.0 does not include support for audio devices.

For information about USB audio support in specific Oracle Solaris releases, see “About USB Support in Oracle Solaris” on page 102.
Oracle Solaris USB audio support is implemented by a pair of cooperating drivers, usb_ac and usb_as. The audio control driver, usb_ac, is a Solaris USB Architecture compliant client driver that provides the controlling interface to user applications. The audio streaming driver, usb_as, processes audio data messages during play and record. It sets sample frequency and precision, and encodes requests from the usb_ac driver. Both drivers comply with the USB audio class 1.0 specification.

Some audio devices can set volume under software control. A STREAMS module, usb_ah, is pushed on top of the HID driver for managing this function.

Oracle Solaris supports USB audio devices that are play-only, record-only, or record and play.

Hot-plugging of USB audio devices is supported, as follows:

- For fully supported audio data format information, see usb_ac(7D).

The primary audio device is /dev/audio. You can verify that /dev/audio is pointing to USB audio by using the following command:

```bash
% 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 audierecord command through the /dev/sound/N device links.

Note that the /dev/audio and /dev/sound/N devices can refer to speakers, microphones, or combination devices. If you refer to the incorrect device type, the command fails. For example, the audioplay command fails if you try to use it with a microphone.

You can select a specific default audio device for most Oracle audio applications, such as audioplay and audierecord, by setting the AUDIODEV shell variable or by specifying the -d option for these commands. However, setting AUDIODEV does not work for third-party 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 on-board audio to USB audio and vice versa, refer to “How to Change the Primary USB Audio Device” on page 130, 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 USB Audio Devices**

1. **Plug in the USB speaker.**
   
   The primary audio device, `/dev/audio`, points to the USB speaker.
   
   ```
   $ ls -l /dev/audio
   lrwxrwxrwx 1 root root 10 Feb 13 08:46 /dev/audio -> usb/audio0
   ```

2. **(Optional) Remove the speaker. Then, plug it back in.**
   
   If you remove the speaker, the `/dev/audio` device reverts back to on-board audio.
   
   ```
   $ ls -l /dev/audio
   lrwxrwxrwx 1 root root 7 Feb 13 08:47 /dev/audio -> sound/0
   ```

3. **Add a USB microphone.**
   
   ```
   $ ls -l /dev/audio
   lrwxrwxrwx 1 root root 10 Feb 13 08:54 /dev/audio -> usb/audio1
   ```

**How to Identify Your System's Primary Audio Device**

This procedure assumes that you have already connected the USB audio devices.

- **Examine your system's new audio links.**

  Display your system’s new audio links with the `ls` command.

  For example:

  ```
  $ ls -lt /dev/audio*
  lrwxrwxrwx 1 root root 7 Jul 23 15:46 /dev/audio -> usb/audio0
  lrwxrwxrwx 1 root root 10 Jul 23 15:46 /dev/audioctl ->
  usb/audioctl0/
  % 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:
  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:...
  lrwxrwxrwx 1 root root 10 Jul 23 15:46 /dev/audio ->
  usb/audio0
  % ls -lt /dev/"audio"*
  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:...
  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:...
  lrwxrwxrwx 1 root root 10 Jul 23 15:46 /dev/audio ->
  usb/audio0
  ```

  Notice that the primary audio device, `/dev/audio`, is pointing to the newly plugged in USB audio device, `/dev/usb/audio0`.

- You can also examine your system’s USB audio devices with the `prtconf` command and look for the USB device information.
How to Change the Primary USB Audio Device

Select one of the following to change the primary USB audio device:

- If you want the on-board audio device to become the primary audio device, remove the USB audio devices. The /dev/audio link then points to the /dev/sound/0 entry. If the /dev/sound/0 entry is not the primary audio device, then either shut down the system and use the boot -r command, or run the devfsadm -i command as root.

- If you want the USB audio device to become primary audio device, just plug it in and check the device links.

Troubleshooting USB Audio Device 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 to 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 in to the console.

- If you are not logged in to the console when you plug in a USB audio device, root becomes the owner of the device. However, if you log in to the console and attempt to access the USB audio device, then device ownership changes to the console. For more information, see logindevperm(4).

- When you remotely log in with the rlogin command and attempt to access the USB audio device, the ownership does not change. This situation means that, for example, unauthorized users cannot listen to conversations over a microphone that is owned by someone else.

Hot-Plugging USB Devices With the cfgadm Command

The following tasks are provided in this section:

- “How to Display USB Bus Information (cfgadm)” on page 132
- “How to Unconfigure a USB Device” on page 133
- “How to Configure a USB Device” on page 134
- “How to Logically Disconnect a USB Device” on page 134
- “How to Logically Connect a USB Device” on page 134
- “How to Logically Disconnect a USB Device Subtree” on page 135
- “How to Reset a USB Device” on page 135
- “How to Change the Default Configuration of a Multi-Configuration USB Device” on page 136

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 the following:

- An occupant, which represents a hardware resource, such as a USB device, that might be configured into the system
- A receptacle, which is the location that accepts the occupant, such as a USB port
Attachment points are represented by logical and physical attachment point IDs (Ap_Ids). The physical Ap_Id is the physical path name of the attachment point. The logical Ap_Id is a user-friendly alternative for the physical Ap_Id. For more information on Ap_Ids, see \texttt{cfgadm\_usb(1M)}.

The \texttt{cfgadm} command provides the following USB device status information.

<table>
<thead>
<tr>
<th>Receptacle State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty/unconfigured</td>
<td>The device is not physically connected.</td>
</tr>
<tr>
<td>disconnected/unconfigured</td>
<td>The device is logically disconnected and unavailable, even though the device could still be physically connected.</td>
</tr>
<tr>
<td>connected/unconfigured</td>
<td>The device is logically connected, but unavailable. The device is visible in \texttt{prtconf} output.</td>
</tr>
<tr>
<td>connected/configured</td>
<td>The device is connected and available.</td>
</tr>
</tbody>
</table>

The following sections describe how to hot-plug a USB device through the software with the \texttt{cfgadm} command. All of the sample USB device information in these sections has been truncated to focus on relevant information.

\section*{\textbf{How to Display USB Bus Information (\texttt{cfgadm})}}

For examples of using the \texttt{prtconf} command to display USB configuration information, see "How to Display USB Device Information" on page 116.

\subsection*{1 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
```

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.
2 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  ...
usb0/4.5.3  Mfg: Mitsumi Product: Apple USB Mouse NConfigs: 1
         Config: 0  ...
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  ...
usb0/4.5.6  Mfg: 3Com Inc.  Product: U.S.Robotics 56000 Voice USB Modem
         NConfigs: 2  ...
```

▼ How to Unconfigure a USB Device

You can unconfigure a USB device that is still physically connected to the system. However, a
driver will never attach to the device. Note that a USB device remains in the prtconf output
even after that device is unconfigured.

1 Become superuser.

2 Unconfigure the USB device.

For example:

```
# cfgadm -c unconfigure usb0/4.7
```

```
Unconfigure the device: /devices/pci@0,700000/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.

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
```
Hot-Plugging USB Devices With the \texttt{cfgadm} Command

\section*{How to Configure a USB Device}

1. Become superuser.

2. Configure a USB device.
   For example:
   
   \begin{verbatim}
   # cfgadm -c configure usb0/4.7
   \end{verbatim}

3. Verify that the USB device is configured.
   For example:
   
   \begin{verbatim}
   # cfgadm usb0/4.7
   \end{verbatim}

   \begin{tabular}{lllll}
   Ap Id & Type   & Receptacle & Occupant & Condition \\
   \hline
   usb0/4.7 & usbstorage & connected & configured & ok
   \end{tabular}

\section*{How to Logically Disconnect a USB Device}

If you want to remove a USB device from the system and the \texttt{prtconf} output, but you are not physically near the system, just logically disconnect the USB device. The device is still physically connected. However, the device is logically disconnected, unusable, and not visible to the system.

1. Become superuser.

2. Disconnect a USB device.
   For example:
   
   \begin{verbatim}
   # cfgadm -c disconnect -y usb0/4.7
   \end{verbatim}

3. Verify that the device is disconnected.
   For example:
   
   \begin{verbatim}
   # cfgadm usb0/4.7
   \end{verbatim}

   \begin{tabular}{lllll}
   Ap Id & Type   & Receptacle & Occupant & Condition \\
   \hline
   usb0/4.7 & unknown & disconnected & unconfigured & ok
   \end{tabular}

\section*{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.**
For example:
```
# cfgadm -c configure usb0/4.7
```

3 **Verify that the device is connected.**
For example:
```
# cfgadm usb0/4.7
```

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Type</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>usb0/4.7</td>
<td>usb-storage</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
</tbody>
</table>

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.**
For example:
```
# cfgadm -c disconnect -y usb0/4
```

3 **Verify that the USB device subtree is disconnected.**
For example:
```
# cfgadm usb0/4
```

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Type</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>usb0/4</td>
<td>unknown</td>
<td>disconnected</td>
<td>unconfigured</td>
<td>ok</td>
</tr>
</tbody>
</table>

▼ **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 that the device is not in use.**

3 **Reset the device.**
For example:
```
# cfgadm -x usb_reset -y usb0/4.7
```
4 Verify that the device is connected.

For example:

```
# cfgadm usb0/4.7
```

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Type</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>usb0/4.7</td>
<td>usb-storage</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
</tbody>
</table>

**How to Change the Default Configuration of a Multi-Configuration USB Device**

Keep the following in mind when working with multi-configuration USB devices:

- A USB device configuration defines how a device presents itself to the operating system. This method is different from system device configurations discussed in other `cfgadm` sections.
- Some USB devices support multiple configurations, but only one configuration can be active at a time.
- Multi-configuration devices can be identified by examining the `cfgadm -lv` output. `Nconfigs` will be greater than 1.
- The default USB configuration is configuration 1. The current configuration is reflected in `cfgadm -lv` output as `Config`.
- Changes to the default configuration persist across reboots, hot-removes, and the reconfiguration of the device, as long as the device is reconnected to the same port.

1 Make sure that the device is not in use.

2 Change the default USB configuration.

For example:

```
# cfgadm -x usb_config -o config=2 usb0/4
```

```
Setting the device: /devices/pci@1f,0/usb@c,3:4
to USB configuration 2
This operation will suspend activity on the USB device
Continue (yes/no)? yes
```

3 Verify that the device changed.

For example:

```
# cfgadm -lv usb0/4
```

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
<th>Information</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>usb0/4</td>
<td>connected</td>
<td>unconfigured</td>
<td>ok</td>
<td>Mfg: Sun 2000</td>
<td>USB-B0B0 aka Robotech</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>With 6 EPPS High Clk Mode</td>
<td>NConfigs: 7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Product: USB-8080 aka Robotech</td>
<td>Config: 2 : EVAL Board Setup unavailable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>usb-device</td>
<td>/devices/pci@1f,0/usb@c,3:4</td>
</tr>
</tbody>
</table>

Note that `Config: now shows 2.`
Using InfiniBand Devices (Overview/Tasks)

InfiniBand (IB) is a I/O technology based on switch fabrics. It provides high bandwidth, low latency interconnect for attaching I/O devices to hosts and for host-to-host communication.

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

- “Overview of InfiniBand Devices” on page 137
- “Dynamically Reconfiguring IB Devices (cfgadm)” on page 140

For information on the procedures associated with using IB devices, see the following:

- “Dynamically Reconfiguring IB Devices (Task Map)” on page 138
- “Using the uDAPL Application Interface With InfiniBand Devices” on page 149

For general information about dynamic reconfiguration and hot-plugging, see Chapter 4, “Dynamically Configuring Devices (Tasks).”

Overview of InfiniBand Devices

IB devices are managed by the Solaris IB nexus driver. This driver supports 5 types of devices:

- IB Port devices
- IB virtual physical point of attachment (VPPA) devices
- IB HCA service (HCA_SVC) devices
- Pseudo devices
- I/O controller (IOC) devices

The IB nexus driver queries the Solaris IB Device Manager (IBDM) for services, referred in this guide as communication services, to enumerate the IB Port, HCA_SVC, and IB VPPA devices.

The Port devices bind a communication service to a given port# of a Host Channel Adapter (HCA). The VPPA devices bind a communication service to a port#, p_key# combination instead. The HCA_SVC devices bind a communication service to a given HCA. Note that the
Port devices and the HCA_SVC devices always use a p_key (partition key) whose value is zero. The Port, HCA_SVC, and VPPA devices are children of the HCA and are enumerated through the ib.conf file. For more information, see ib(7D).

The IOC devices are children of the IB nexus driver and are part of an I/O unit. The pseudo devices are also children of the IB nexus driver and refer to all other devices that provide their own configuration files to enumerate. For more information, see ib(4).

The possible IB device tree path name(s) are listed in the following table.

<table>
<thead>
<tr>
<th>IOC device</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>IB pseudo device</td>
<td>/ib/&lt;driver&gt;@&lt;unit-address&gt;</td>
</tr>
<tr>
<td>IB VPPA device</td>
<td>/pci@1f,2000/pci@1/pci15b3,5a44@0/ibport@&lt;port#&gt;,&lt;p_key&gt;,&lt;service&gt;</td>
</tr>
<tr>
<td>IB HCA_SVC device</td>
<td>/pci@1f,2000/pci@1/pci15bc,5a44@0/ibport@0,&lt;service&gt;</td>
</tr>
<tr>
<td>IB Port device</td>
<td>/pci@1f,2000/pci@1/pci15b3,5a44@0/ibport@&lt;port#&gt;,0,&lt;service&gt;</td>
</tr>
<tr>
<td>HCA</td>
<td>/pci@1f,2000/pci@1/pci15b3,5a44@0</td>
</tr>
</tbody>
</table>

Note that the IB HCA_SVC devices have zero as the port# and the p_key.

The IB components in the preceding table are described as follows:

<services> Is a communication service. For example, ipib is the communication service used by the ibd kernel client driver.

<p_key> Is the partition key value being used.

<port> Is the port number.

<unit-address> Refers to IB kernel client driver’s property by this name specified in its driver.conf file. For more information, see driver.conf(4).

---

**Dynamically Reconfiguring IB Devices (Task Map)**

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display IB device information.</td>
<td>Display information about the IB devices on your system.</td>
<td>&quot;How to Display IB Device Information&quot; on page 141</td>
</tr>
<tr>
<td>Configure or unconfigure an IOC device.</td>
<td>Select one of the following:</td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Description</td>
<td>For Instructions</td>
</tr>
<tr>
<td>------</td>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Unconfigure an IOC device.</td>
<td>&quot;How to Unconfigure an IOC Device&quot; on page 142</td>
</tr>
<tr>
<td></td>
<td>Configure an IOC device.</td>
<td>&quot;How to Configure an IOC Device&quot; on page 143</td>
</tr>
<tr>
<td>Configure or unconfigure a port or VPPA device.</td>
<td>Select one of the following:</td>
<td>&quot;How to Unconfigure an IB Port, HCA_SVC, or a VPPA Device&quot; on page 143</td>
</tr>
<tr>
<td></td>
<td>Unconfigure a port or a VPPA device.</td>
<td>&quot;How to Configure a IB Port, HCA_SVC, or a VPPA Device&quot; on page 144</td>
</tr>
<tr>
<td></td>
<td>Configure a port or a VPPA device.</td>
<td></td>
</tr>
<tr>
<td>Configure or unconfigure an IB pseudo device.</td>
<td>Select one of the following:</td>
<td>&quot;How to Unconfigure an IB Pseudo Device&quot; on page 144</td>
</tr>
<tr>
<td></td>
<td>Unconfigure an IB pseudo device.</td>
<td>&quot;How to Configure an IB Pseudo Device&quot; on page 145</td>
</tr>
<tr>
<td></td>
<td>Configure an IB pseudo device.</td>
<td></td>
</tr>
<tr>
<td>Display kernel IB clients of an HCA.</td>
<td>You might need to display information about kernel IP clients of an HCA, particularly if you're going to unconfigure an HCA.</td>
<td>&quot;How to Display Kernel IB Clients of an HCA&quot; on page 145</td>
</tr>
<tr>
<td>Configure or unconfigure an IB HCA.</td>
<td>Select one of the following:</td>
<td>&quot;How to Unconfigure IB Devices Connected to an HCA&quot; on page 145</td>
</tr>
<tr>
<td></td>
<td>Unconfigure IB devices that are connected to an HCA.</td>
<td>&quot;Configuring an IB HCA&quot; on page 146</td>
</tr>
<tr>
<td></td>
<td>Configure IB devices that are connected to an HCA.</td>
<td></td>
</tr>
<tr>
<td>Update the IB p_key tables.</td>
<td>If the p_key table information of a HCA port changes, IBTF and IBDM need to be notified so that their internal p_key databases are updated.</td>
<td>&quot;How to Update the IB p_key Tables&quot; on page 146</td>
</tr>
<tr>
<td>Display IB communication services</td>
<td>Display the IB communication services that are currently in use by the IBTF.</td>
<td>&quot;How to Display IB Communication Services&quot; on page 147</td>
</tr>
</tbody>
</table>
Dynamically Reconfiguring IB Devices (\texttt{cfgadm})

One can configure or unconfigure an IB device from a running system by using the \texttt{cfgadm} CLI only. This command also provides a way to display the IB fabric, manage communication services, and update p_key table databases. For more information, see \texttt{cfgadm\_ib(1M)}.

The \texttt{cfgadm} CLI manages dynamic reconfiguration, referred to in this guide as DR, of the entire IB fabric as seen by a host. The \texttt{cfgadm} operations are supported on all the IB devices, such as Port, VPPA, HCA\_SVC, IOC, and pseudo devices.

The \texttt{cfgadm} command displays information about attachment points (Ap\_Ids), which are locations in the system where DR operations can occur. For details on the Ap\_Ids that \texttt{cfgadm} supports, see \texttt{cfgadm\_ib(1M)}. Note that all IB Ap\_Ids are shown as connected.

The \texttt{cfgadm} command provides the following IB device status information.

<table>
<thead>
<tr>
<th>Receptacle State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connected/configured/ok</td>
<td>The device is connected and available. The \texttt{devinfo} node is present.</td>
</tr>
<tr>
<td>connected/unconfigured/unknown</td>
<td>The device is unavailable and no \texttt{devinfo} node or device driver exists for this device. Or, the device was never configured for use by \texttt{ib} nexus driver. The device might be known to the IB Device Manager.</td>
</tr>
</tbody>
</table>

The following sections describe how to dynamically reconfigure (DR) IB devices with the \texttt{cfgadm} command. All of the sample IB device information in these sections has been truncated to focus on relevant information.
How to Display IB Device Information

You can use the `prtconf` command to display general information about IB devices. For example:

```
$ prtconf
    pci, instance #0
    pci15b3,5a44, instance #0
    ibport, instance #253
    ibport, instance #254
    ibport, instance #255
    ...
    ib, instance #0
    ioc, instance #243
    ioc, instance #244
    ioc, instance #245
    ioc, instance #246
    ioc, instance #247
    ioc, instance #248
    ibgen, instance #249
```

In the preceding example, `pci15b3,5a44` refers to an IB HCA.

Use the following steps to display specific IB device information.

1. Become superuser.

2. Display IB fabric information.

For example:

```
# cfgadm -a

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Type</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ib:1730000008070</td>
<td>IB-Fabric</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib:1730000007F5198</td>
<td>IB-IOC</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib:1730000007F5199</td>
<td>IB-IOC</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib:1730000008070</td>
<td>IB-IOC</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib:1730000008071</td>
<td>IB-PORT</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib:1730000008072</td>
<td>IB-PORT</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib:1730000008071</td>
<td>IB-PORT</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib:1730000008072</td>
<td>IB-PORT</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib:1730000008072,0,ipib</td>
<td>IB-VPPA</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib:1730000008072,8001,ipib</td>
<td>IB-VPPA</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
<tr>
<td>ib::ibgen,0</td>
<td>IB-PSEUDO</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
</tbody>
</table>
```

In the preceding example output, the components are described as follows:

- `Ap_Id ib::1730000008072,0,ipib`: Identifies an IB Port device that is connected to port 2 and is bound to the ipib service.
- `Ap_Id ib::1730000008072,8001,ipib`: Identifies an IB VPPA device that is connected to port 2, using a `p_key` value of `0x8001`, and is bound to the ibd service.
Ap_Id ib:: 1730000008070,0,hnfs
Identifies an IB HCA_SVC device bound to the
hnfs service.

Ap_Id ib::1730000007F5198
Identifies an IOC device.

Ap_Id ib::ibgen,0
Identifies a pseudo device.

3 Display specific IB device information.
For example, for an IB VPPA device:

```
# cfgadm -al -s "cols=ap_id:info" ib::1730000008072,8001,ipib
Ap Id Information
ib::1730000008072,8001,ipib ipib
```

For example, for an IB HCA device:

```
# cfgadm -al -s "cols=ap_id:info" hca::1730000008070
Ap Id Information
hca::1730000008070 VID: 0x15b3, PID: 0x5a44, #ports: 0x2,
port1 GUID: 0x1730000008071, port2 GUID: 0x1730000008072
```
The preceding output displays the number of ports and their GUIDs.

▼ How to Unconfigure an IOC Device

You can unconfigure an IB device that is still physically connected to the system, but a driver
will never attach to it.

1 Become superuser.

2 Unconfigure the IB device.
For example:

```
# cfgadm -c unconfigure ib::1730000007F5198
Unconfigure the device: /devices/ib:fabric::1730000007F5198
This operation will suspend activity on the IB device
Continue (yes/no)? y
```

3 Verify that the device is unconfigured.
For example:

```
# cfgadm -a ib::1730000007F5198
ib::1730000007F5198 IB-IOC connected unconfigured unknown
```
How to Configure an IOC Device

1. Become superuser.

2. Configure a IB device.
   For example:
   
   ```
   # cfgadm -yc configure ib::1730000007F5198
   ```

3. Verify that the IB device is configured.
   For example:
   
   ```
   # cfgadm -al ib::1730000007F5198
   ```
   
<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Type</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ib::1730000007F5198</td>
<td>IB-IOC</td>
<td>connected</td>
<td>configured</td>
<td>ok</td>
</tr>
</tbody>
</table>

How to Unconfigure an IB Port, HCA_SVC, or a VPPA Device

Use the following steps if you want to remove an IB Port, HCA_SVC, or a VPPA device from the system.

The example below illustrates how to unconfigure a VPPA device, but the same procedure applies to Port and HCA_SVC devices as well.

1. Become superuser.

2. Unconfigure the IB VPPA device.
   For example:
   
   ```
   # cfgadm -c unconfigure ib::1730000007F51,8001,ipib
   ```
   
   Unconfigure the device: /devices/ib:fabric::1730000007F51,8001,ipib
   This operation will suspend activity on the IB device
   Continue (yes/no)? Y

3. Verify that the device is disconnected.
   For example:
   
   ```
   # cfgadm -a ib::1730000007F51,8001,ipib
   ```
   
<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>Type</th>
<th>Receptacle</th>
<th>Occupant</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ib::1730000007F51,8001,ipib</td>
<td>IB-VPPA</td>
<td>connected</td>
<td>unconfigured</td>
<td>unknown</td>
</tr>
</tbody>
</table>
How to Configure a IB Port, HCA_SVC, or a VPPA Device

Use the following steps if you want to configure an IB Port, HCA_SVC, or a VPPA device on the system.

The example below illustrates how to configure a VPPA device, but similar steps can be used to configure Port and HCA_SVC devices as well.

1 Become superuser.

2 Configure the IB VPPA device.
   For example:
   ```
   # cfgadm -c configure ib::1730000007F51,8001,ipib
   ```

3 Verify that the device is connected.
   For example:
   ```
   # cfgadm -a ib::1730000007F51,8001,ipib
   Ap_Id Type Receptacle Occupant Condition
   ib::1730000007F51,8001,ipib IB-VPPA connected configured ok
   ```

Note – A cfgadm based configure or unconfigure operation of IB Port and HCA_SVC devices is similar to the preceding examples for an IB VPPA device.

How to Unconfigure an IB Pseudo Device

Use the following steps if you want to remove an IB pseudo device from the system.

1 Become superuser.

2 Unconfigure the IB pseudo device.
   For example:
   ```
   # cfgadm -c unconfigure ib::ibgen,0
   Unconfigure the device: /devices/ib::fabric::ibgen,0
   This operation will suspend activity on the IB device
   Continue (yes/no)? Y
   ```

3 Verify that the device is disconnected.
   ```
   # cfgadm -a ib::ibgen,0
   Ap_Id Type Receptacle Occupant Condition
   ib::ibgen,0 IB-PSEUDO connected unconfigured unknown
   ```
How to Configure an IB Pseudo Device

Use the following steps to configure an IB pseudo device.

1. Become superuser.

2. Configure the IB pseudo device.
   For example:
   ```
   # cfgadm -yc configure ib::ibgen,0
   ```

3. Verify that the device is connected.
   For example:
   ```
   # cfgadm -a ib::ibgen,0
   Ap_Id      Type  Receptacle Occupant    Condition
   ib::ibgen,0 IB-PSEUDO connected configured ok
   ```

How to Display Kernel IB Clients of an HCA

The following IB `cfgadm` plugin command can be invoked to list kernel IB clients using this HCA. Note that the last column would show a “yes” if a kernel IB client uses another HCA. IB Managers and kernel clients that do not use the HCA are shown with an Ap_Id of “-”.

- Display kernel IB clients of an HCA.

For example:
```
$ cfgadm -x list_clients hca:173000007F50
Ap_Id      IB Client      Alternate HCA
ib::1730000007F5ID0 ibgen     no
ib::1730000007F5ID1 ibgen     no
ib::1730000007F51,8001,ipib ibd    no
ib::ibgen,0 ibgen     no
- ibdm      no
- ibmf      no
- nfs/ib    no
```
1 Become superuser.

2 List the kernel IB clients of the HCA.
For example:
```
# cfgadm -x list_clients hca:173000007F50
```

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>IB Client</th>
<th>Alternate HCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ib::173000007F51D0</td>
<td>ibgen</td>
<td>no</td>
</tr>
<tr>
<td>ib::173000007F51D1</td>
<td>ibgen</td>
<td>no</td>
</tr>
<tr>
<td>ib::173000007F51,001,ipib</td>
<td>ibd</td>
<td>no</td>
</tr>
<tr>
<td>- ibgen,0</td>
<td>ibgen</td>
<td>no</td>
</tr>
<tr>
<td>- ibdm</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>- ibmf</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>- nfs/ib</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

3 Unconfigure kernel IB clients, such as Port, VPPA, HCA_SVC, or IOC devices, that do not have alternate HCAs present.
For example:
```
# cfgadm -x unconfig_clients hca:1730000008070
```

Unconfigure Clients of HCA /devices/ib:1730000008070
This operation will unconfigure IB clients of this HCA
Continue (yes/no)? y

4 Verify that the kernel IB clients of the HCA are unconfigured.
```
# cfgadm -x list_clients hca:173000007F50
```

<table>
<thead>
<tr>
<th>Ap_Id</th>
<th>IB Client</th>
<th>Alternate HCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>- ibdm</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>- ibmf</td>
<td>no</td>
<td></td>
</tr>
<tr>
<td>- nfs/ib</td>
<td>no</td>
<td></td>
</tr>
</tbody>
</table>

### Configuring an IB HCA

Invoke the bus-specific `cfgadm` plugin to configure the HCA. The exact details are beyond the scope of this chapter.

#### How to Update the IB p_key Tables

If the p_key table information of an HCA ports changes, for example, additional p_keys are enabled or disabled, InfiniBand Transport Framework (IBTF) and IBDM need to be notified so that their internal p_key databases are updated. The `cfgadm` command helps update the p_key databases of IBTF and IBDM. For more information, see `ibtl(7D)` and `ibdm(7D)`.

1 Become superuser.
2 Update the p_key tables.
   For example:
   ```
   # cfgadm -x update_pkey_tbls -y ib
   ```

▼ **How to Display IB Communication Services**

Use the following steps to display the communication services that are currently in use by the IBTF.

1 Become superuser.

2 Display IB communication services.
   For example:
   ```
   # cfgadm -x list_services ib
   Port communication services:
   
   srp
   VPPA communication services:
   
   ibd
   HCA_SVC communication services:
   
   hnfs
   ```

▼ **How to Add a VPPA Communication Service**

Use the following steps to add a new VPPA communication service.

Similar steps can be used to add a new HCA_SVC or a port communication service.

1 Become superuser.

2 Add a new VPPA communication service.
   For example:
   ```
   # cfgadm -o comm=vppa,service=new -x add_service ib
   ```

3 Verify that the new service has been added.
   For example:
   ```
   # cfgadm -x list_services ib
   Port communication services:
   
   srp
   VPPA communication services:
   
   ibd
   new
   HCA_SVC communication services:
   
   nfs_service
   ```
How to Remove an Existing IB Port, HCA_SVC, or a VPPA Communication Service

Use the following steps to delete an existing IB Port, HCA_SVC, or a VPPA communication service.

1. Become superuser.

2. Remove a VPPA communication service.
   For example:
   ```
   # cfgadm -o comm=vppa,service=new -x delete_service ib
   ```

3. Verify that the communication service has been removed.
   For example:
   ```
   # cfgadm -x list_services ib
   Port communication services:
   srp
   VPPA communication services:
   ibd
   HCA_SVC communication services:
   hnfs
   ```

How to Update an IOC Configuration

Use the following steps to update properties of all the IOC device nodes or for a particular IOC Ap_Id. The properties that can get updated are as follows:

- port-list
- port-entries
- service-id
- service-name

For more information on these properties, see `ib(7D)`.

Note that these properties may not get updated if there is no configuration change. The following example describes how to update a particular IOC's configuration. If you need to update the configuration of all the IOCs, then specify the static `ib Ap_Id` instead of the particular IOC Ap_Id.

1. Become superuser.
2 Update the configuration of an IOC.

For example:

```
# cfgadm -x update_ioc_conf ib::1730000007F5198
```

This operation can update properties of IOC devices.

Continue (yes/no)? y

3 Verify that the properties have been updated by running `prtconf -v`.

Using the uDAPL Application Interface With InfiniBand Devices

User Direct Access Programming Library (uDAPL) is a standard API that promotes data center application data messaging performance, scalability, and reliability over Remote Direct Memory Access (RDMA) capable interconnects such as InfiniBand. The uDAPL interface is defined by the DAT collaborative. For more information about the DAT collaborative, go to the following site:

http://www.datcollaborative.org

The Solaris release provides the following uDAPL features:

- A standard DAT registry library, `libdat`. For more information, see `libdat(3LIB)`.
- A standard service provider registration file, `dat.conf`. For more information, see `dat.conf(4)`.
- Support for multiple service providers so that each provider specifies their own uDAPL library path, version number, and so on, in their own `service_provider.conf` file. For more information, see `service_provider.conf(4)`.
- An administrative tool, the `datadm` command, to configure `dat.conf`. For more information, see `datadm(1M)`.
- A new resource control property, `project.max-device-locked-memory`, to regulate the amount of locked down physical memory.
- A naming scheme that uses either IPv4 or IPv6 addresses that leverage the IP infrastructure, such as ARP in IPv4 and neighbor discovery in IPv6, for address resolution. The Solaris uDAPL Interface Adapter directly maps to an IPoIB device instance.
- Support for the standard Address Translation Scheme that is used by the DAT collaborative community.
- A uDAPL service provider library to support the Mellanox Tavor Host Channel Adapter with automatic registration to the `dat.conf` registration file.
- Supports both SPARC platform and x86 platforms.
How to Enable uDAPL

1 Become superuser.

2 Confirm that the following packages are installed. Or, install them, if needed.
   - SUNWib – Sun InfiniBand Framework
   - SUNWtavor – Sun Tavor HCA Driver
   - SUNWipoib – Sun IP over InfiniBand
   - SUNWudapl – Sun User Direct Access Programming Library (root)
   - SUNWudaplu – Sun User Direct Access Programming Library (usr)
   - SUNWudapltr – Sun uDAPL for Tavor (root)
   - SUNWudapltu – Sun uDAPL for Tavor (usr)

3 Select one of the following to plum the IPoIB interfaces.
   - Manually plum the interfaces with the ifconfig and datadm commands.
     For example:
     ```
     # ifconfig ibd1 plumb
     # ifconfig ibd1 192.168.0.1/24 up
     # datadm -a /usr/share/dat/SUNWudaplt.conf
     ```
   - Automatically plum the interfaces by doing the following:
     - Create the following file with the appropriate IP address.
       ```
       /etc/hostname.ibd1
       ```
     - Reboot the system.

Updating the DAT Static Registry

You can use the datadm command to maintain the DAT static registry, the dat.conf file. For more information about this file, see dat.conf(4).

The datadm command can also be used to register or unregister a service provider to the dat.conf file. For more information, see datadm(1M).

When IPoIB interface adapters are added or removed, run the datadm command to update the dat.conf file to reflect the current state of the system. A new set of interface adapters for all the service providers that are currently installed will be regenerated.
How to Update the DAT Static Registry

1. Become superuser.

2. Update the DAT static registry after you add or remove IPoIP interface adapters from the system.
   
   ```
   # datadm -u
   ```

3. Display the updated DAT static registry.
   
   ```
   # datadm
   ```

How to Register a Service Provider in the DAT Static Registry

1. Become superuser.

2. Update the DAT static registry after you add Sun’s service provider for the Mellanox Tavor Host Channel Adapter.
   
   ```
   # datadm -a /usr/share/dat/SUNWudaplt.conf
   ```

3. Display the updated DAT static registry.
   
   ```
   # datadm -v
   ```

How to Unregister a Service Provider from the DAT Static Registry

1. Become superuser.

2. Update the DAT static registry after you remove Sun’s service provider for the Mellanox Tavor Host Channel Adapter from the system.
   
   ```
   # datadm -r /usr/share/dat/SUNWudaplt.conf
   ```

3. Display the updated DAT static registry.
   
   ```
   # datadm -v
   ```
Managing Disks (Overview)

This chapter provides overview information about Oracle Solaris disk slices and introduces the `format` utility.

This is a list of overview information in this chapter:

- “What’s New in Disk Management?” on page 153
- “Where to Find Disk Management Tasks” on page 154
- “Overview of Disk Management” on page 154
- “Partitioning a Disk” on page 162

For instructions on how to add a disk to your system, see Chapter 10, “SPARC: Setting Up Disks (Tasks),” or “x86: Setting Up Disks for ZFS File Systems (Task Map)” on page 222.

What’s New in Disk Management?

The following disk management features are new in this Oracle Solaris release:

- “Advanced Format Disk Support” on page 153

Advanced Format Disk Support

Previous Oracle Solaris releases support disks with a physical block size and a logical block size of 512 bytes. This is the traditional disk block size that is an industry standard.

Currently, disk manufacturers are providing larger capacity disks, also known as advanced format (AF) disks, which is a general term that describes a hard disk drive that exceeds a 512-byte block size.
AF disks are generally in the 4-KB block size range, but vary as follows:

- 4-KB native disk (4kn) – Has a physical and logical block size of 4 KB
- 512-byte emulation (512e) – Has a physical block size of 4 KB but reports a logical block size of 512 bytes

For comparison purposes, Oracle Solaris introduces the 512-byte native (512n) disk term, which is a traditional disk with 512-byte block size.

Oracle Solaris 10 1/13 supports advanced format disks, in addition to traditional 512n disks, in the following ways:

- Non-root UFS or ZFS file systems are supported on 4kn and 512e disks
- SPARC: You can boot and install a UFS or ZFS root file system on 512e disk.
- x86: You can boot and install a UFS root file system on a 512e disk.

Before purchasing advanced format drives, confirm with your device manufacturer that their 512e devices have a power-safe feature to prevent data loss after a power failure when data is still in transit.

**Where to Find Disk Management Tasks**

Use these references to find step-by-step instructions for managing disks.

<table>
<thead>
<tr>
<th>Disk Management Task</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format a disk and examine a disk label.</td>
<td>Chapter 9, “Administering Disks (Tasks)”</td>
</tr>
<tr>
<td>Add a new disk to a SPARC system.</td>
<td>Chapter 10, “SPARC: Setting Up Disks (Tasks)”</td>
</tr>
<tr>
<td>Add a new disk to an x86 system.</td>
<td>“x86: Setting Up Disks for ZFS File Systems (Task Map)” on page 222</td>
</tr>
<tr>
<td>Hot-plug a SCSI or PCI disk.</td>
<td>Chapter 4, “Dynamically Configuring Devices (Tasks)”</td>
</tr>
</tbody>
</table>

**Overview of Disk Management**

Managing disks in the Oracle Solaris OS usually involves setting up the system and running the Oracle Solaris installation program to create the appropriate disk slices and file systems and to install the Oracle Solaris OS. Occasionally, you might need to use the `format` utility to add a new disk drive or replace a defective disk drive.
The following information is described in this section:

- “Disk Terminology” on page 155
- “About Disk Labels” on page 155
- “EFI (GPT) Disk Label” on page 156
- “About Disk Slices” on page 159
- “format Utility” on page 159

**Disk Terminology**

Before you can effectively use the information that is described in this section, you should be familiar with basic disk architecture. In particular, you should be familiar with the terms in the following table.

<table>
<thead>
<tr>
<th>Disk Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Track</td>
<td>A concentric ring on a disk that passes under a single stationary disk head as the disk rotates.</td>
</tr>
<tr>
<td>Cylinder</td>
<td>The set of tracks with the same nominal distance from the axis about which the disk rotates.</td>
</tr>
<tr>
<td>Sector</td>
<td>Section of each disk platter.</td>
</tr>
<tr>
<td>Block</td>
<td>A data storage area on a disk.</td>
</tr>
<tr>
<td>Disk controller</td>
<td>A chip and its associated circuitry that controls the disk drive.</td>
</tr>
<tr>
<td>Disk label</td>
<td>Part of the disk, usually starting from first sector, that contains disk geometry and partition information.</td>
</tr>
<tr>
<td>Device driver</td>
<td>A kernel module that controls a physical (hardware) or virtual device.</td>
</tr>
</tbody>
</table>

For additional information, see your disk manufacturer’s product information.

**About Disk Labels**

A special area of every disk is set aside for storing information about the disk’s controller, geometry, and slices. This 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 or partitions.
The Oracle Solaris release supports the following two disk labels:

- **SMI** – The traditional VTOC label for disks that are less than 2 TB in size.
- **EFI** – Provides support for disks that are larger than 2 TB. The Extensible Firmware Interface GUID Partition Table (EFI GPT) disk label is also available for disks less than 2 TB.

If you fail to label a disk after you create slices, the slices will be unavailable because the OS has no way of “knowing” about the slices.

### EFI (GPT) Disk Label

The EFI label provides support for physical disks and virtual disk volumes that are greater than 2 TB in size. This release also includes disk utilities for managing disks greater than 2 TB in size.

The following file system products support file systems that are greater than 1 TB in size:

- The Oracle Solaris ZFS file system supports file systems that are greater than 1 TB in size.
- The legacy Solaris Volume Manager software can also be used to manage disks that are greater than 1 TB.

You can use the `format -e` command to apply an EFI label to a disk, if the system is running a supported Oracle Solaris release. However, you should review the important information in “Restrictions of the EFI Disk Label” on page 157 before attempting to apply an EFI label.

You can also use the `format -e` command to reapply a VTOC label, if the EFI label is no longer needed. For example:

```
# format -e
Specify disk {enter its number}: 2
selecting c0t5d0
[disk formatted]
.
.
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0
Warning: This disk has an EFI label. Changing to SMI label will erase all current partitions.
Continue? yes
Auto configuration via format.dat[no]? 
Auto configuration via generic SCSI-2[no]?
format> quit
```

**Caution** – Keep in mind that changing disk labels will destroy any data on the disk.
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 2 terabytes in size.
- Provides usable slices 0-6, where partition 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 usually start at sector 34. This feature means that no partition can start at sector zero (0).
- EFI (GPT) labels do not use the notion of geometry. The partitions are defined based on logical blocks.
- Some information that was stored in the alternate cylinder area is now stored in the last two cylinders of a disk or Solaris partition.
- 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 2 terabytes is appropriate for your environment:

- Layered software products that are intended for systems with VTOC-labeled disks might be incapable of accessing a disk with an EFI disk label.
- On x86 based systems, you can use the `fdisk` command on a disk with an EFI label that is greater than 2 TB in size.
- Use the `format` utility to partition disks with EFI labels.
- The EFI specification prohibits overlapping partitions. The entire disk is represented by `cxydz`.
- The EFI disk label 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 to 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.

x86: Support for EFI-Labeled Disks

Oracle Solaris support for the EFI disk label is available on x86 systems. Use the following command to add an EFI label on an x86 system:
Managing Disks With EFI Labels

Use the following table to locate information on managing disks with EFI labels.

<table>
<thead>
<tr>
<th>Task</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the system is not installed, install it.</td>
<td>Oracle Solaris 10 1/13 Installation Guide: Basic Installations</td>
</tr>
<tr>
<td>The system is already installed, but the root pool disk was damaged or needs to be replaced.</td>
<td>“SPARC: How to Set Up a Disk for a ZFS Root File System” on page 209 or “x86: How to Set Up a Disk for a ZFS Root File System” on page 224</td>
</tr>
<tr>
<td>The system is already installed, but you need to set up a disk for a non-root pool.</td>
<td>“SPARC: How to Set Up a Disk for a ZFS File System” on page 214 or “x86: How to Set Up a Disk for a ZFS File System” on page 229</td>
</tr>
</tbody>
</table>

Troubleshooting Problems With EFI Disk Labels

Use the following error messages and solutions to troubleshoot problems with EFI-labeled disks.

Cause
Booting a system running a SPARC or x86 kernel with a disk greater than 1 terabyte.

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 a disk to a system running an older Solaris release.

Solution
Add the disk to a system running the Solaris release that supports the EFI disk label.
About Disk Slices

Files that are 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 that are set aside for use by that file system. Each disk slice appears to the Oracle Solaris OS (and to the system administrator) as though it were a separate disk drive.

For information about file systems, see Chapter 14, “Managing File Systems (Overview).”

Note – Slices are sometimes referred to as partitions. Certain interfaces, such as the \textit{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.

Using Raw Data Slices

The disk label is stored in block 0 of each disk. So, third-party database applications that create raw data slices must not start at block 0. Otherwise, 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

format Utility

Read the following overview of the \textit{format} utility and its uses before proceeding to the “how-to” or reference sections.

The \textit{format} utility is a system administration tool that is used to prepare hard disk drives for use on your Oracle Solaris system.

The following table describes the features and associated benefits of the \textit{format} utility.
TABLE 7-1 Features and Benefits of the format Utility

<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Searches your system for all attached disk drives</td>
<td>Reports on the following:</td>
</tr>
<tr>
<td></td>
<td>■ Target location</td>
</tr>
<tr>
<td></td>
<td>■ Disk geometry</td>
</tr>
<tr>
<td></td>
<td>■ Whether the disk is formatted</td>
</tr>
<tr>
<td></td>
<td>■ If the disk has mounted partitions</td>
</tr>
<tr>
<td>Retrieves disk labels</td>
<td>Convenient for repair operations</td>
</tr>
<tr>
<td>Repairs defective sectors</td>
<td>Allows administrators to repair disk drives with recoverable errors</td>
</tr>
<tr>
<td></td>
<td>instead of sending the drive back to the manufacturer</td>
</tr>
<tr>
<td>Formats and analyzes a disk</td>
<td>Creates sectors on the disk and verifies each sector</td>
</tr>
<tr>
<td>Partitions a disk</td>
<td>Divides a disk into slices or partitions. ZFS file systems do not</td>
</tr>
<tr>
<td></td>
<td>correspond to disk slices or partitions, except for the ZFS root pool.</td>
</tr>
<tr>
<td>Labels a disk</td>
<td>Writes disk name and configuration information to the disk for future</td>
</tr>
<tr>
<td></td>
<td>retrieval (usually for repair operations)</td>
</tr>
</tbody>
</table>

The format utility options are described in Chapter 13, "The format Utility (Reference)."

When to Use the format Utility

Disk drives are partitioned and labeled by the Oracle Solaris installation utility when you install Oracle Solaris. You can use the format utility to do the following:

■ Display slice or partition information
■ Partition a disk
■ 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 partition a disk. These steps are covered in Chapter 10, “SPARC: Setting Up Disks (Tasks),” and “x86: Setting Up Disks for ZFS File Systems (Task Map)” on page 222.

See the following section for guidelines on using the format utility.
### Guidelines for Using the format Utility

**TABLE 7-2  format Utility Guidelines**

<table>
<thead>
<tr>
<th>Task</th>
<th>Guidelines</th>
<th>For More Information</th>
</tr>
</thead>
</table>
| Format a disk. | - Any existing data is destroyed when you reformat a disk.  
- The need for formatting a disk drive has decreased as more and more manufacturers ship their disk drives formatted and partitioned. You might not need to use the format utility when you add or replace a disk drive to an existing system.  
- If a disk has been relocated and is displaying many disk errors, you can attempt to relabel it. | “How to Format a Disk” on page 185 or “How to Label a Disk” on page 189 |
| Set up a disk that contains a UFS or ZFS root file system. | In a non-redundant configuration, a ZFS root file system data from the damaged disk must be restored from a backup medium. Otherwise, the system will have to be reinstalled by using the installation utility. | “SPARC: How to Set Up a Disk for a ZFS Root File System” on page 209 or “x86: How to Set Up a Disk for a ZFS Root File System” on page 224, or, if the system must be reinstalled, Oracle Solaris 10 1/13 Installation Guide: Basic Installations. |
| Create a VTOC labeled disk slice for a ZFS root pool disk or UFS root file system. | - The best way to use a ZFS storage pool is by creating a pool with whole disks.  
- If a disk is intended to be used for a root pool, you must create a disk slice. This is long-standing boot limitation. | “SPARC: How to Create a Disk Slice for a ZFS Root File System” on page 210 or “x86: How to Set Up a Disk for a ZFS Root File System” on page 224 |
| Set up a disk that contains a UFS or ZFS non-root file system. | A disk that is used for a non-root UFS or ZFS file system usually contains space for user or data files. You can attach or add another disk to a root pool or a non-root pool for more disk space. | “SPARC: How to Set Up a Disk for a ZFS File System” on page 214 or “x86: How to Set Up a Disk for a ZFS File System” on page 229 |

### 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 184.

If you determine that a disk is not formatted, use the format utility to format the disk.
When you format a disk, you accomplish 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 185.

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.

**Partitioning a Disk**

The following information is described in this section:

- “Partition Table Terminology” on page 163
- “Displaying Partition Table Information” on page 163
- “Using the Free Hog Slice” on page 165

The `format` utility is most often used by system administrators to partition a disk. The steps are as follows:

- Determine which slices are needed.
- Determine the size of each slice or partition.
- Use the `format` utility to partition the disk.
- Label the disk with new partition information.
- Create the file system for each partition.

The easiest way to partition a disk is to use the `modify` command from the partition menu of the `format` utility. The `modify` command enables you to create partitions by specifying the size of each partition, 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.
Partition Table Terminology

An important part of the disk label is the partition table. The partition table 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>
<thead>
<tr>
<th>TABLE 7–3</th>
<th>Partition Table Terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Partition Term</strong></td>
<td><strong>Value</strong></td>
</tr>
</tbody>
</table>
| Number | 0–7 | VTOC – Partitions or slices, numbered 0–7.  
| | | EFI – Partitions 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 state is the default for 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 about displaying the partition table, see the following references:

- "Displaying Partition Table Information" on page 163
- "How to Display Disk Slice Information" on page 187
- "How to Examine a Disk Label" on page 193

Displaying Partition Table Information

The following `format` utility output shows an example of a partition table from a 74-GB disk with a VTOC label displayed:

```
Total disk cylinders available: 38756 + 2 (reserved cylinders)

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>3 - 2083</td>
<td>4.00GB</td>
<td>(2081/0/0)</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>2084 - 3124</td>
<td>2.00GB</td>
<td>(1041/0/0)</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wm</td>
<td>0 - 38755</td>
<td>74.51GB</td>
<td>(38756/0/0)</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
</tbody>
</table>
```
The partition table that is displayed by the `format` utility contains the following information.

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part</td>
<td>Partition or slice number. See Table 7–3 for a description of this column.</td>
</tr>
<tr>
<td>Tag</td>
<td>Partition tag. See Table 7–3 for a description of this column.</td>
</tr>
<tr>
<td>Flag</td>
<td>Partition flag. See Table 7–3 for a description of this column.</td>
</tr>
<tr>
<td>Cylinders</td>
<td>The starting and ending cylinder number for the slice. Not displayed on EFI-labeled disks.</td>
</tr>
<tr>
<td>Size</td>
<td>The slice or partition size in MB.</td>
</tr>
<tr>
<td>Blocks</td>
<td>The total number of cylinders and the total number of sectors per slice. Not displayed on EFI-labeled disks.</td>
</tr>
<tr>
<td>First Sector</td>
<td>EFI – The starting block number. Not displayed on VTOC-labeled disks.</td>
</tr>
<tr>
<td>Last Sector</td>
<td>EFI – The ending block number. Not displayed on VTOC-labeled disks.</td>
</tr>
</tbody>
</table>

The following is an example of an EFI disk label displayed by using the `prtvtoc` command.

```bash
# prtvtoc /dev/rdsk/c4t1d0s0
*/dev/rdsk/c4t1d0s0 partition map
*
* Dimensions:
* 512 bytes/sector
* 257694156 sectors
* 2576940000 accessible sectors
*
* Flags:
* 1: unmountable
* 10: read-only
* *
* Partition Tag Flags Sector Count Sector Mount Directory
 0  2  00  00  34  629145600  629145633
 1  4  00  629145634  629145600  1258291233
 6  4  00  1258291234  1318633404  2576924637
 8 11  00  2576924638  16384  2576941021
```

The output of the `prtvtoc` command provides information in the following three sections:

- Dimensions
- Flags
Partition Table

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition</td>
<td>Partition or slice number. For a description of this column, see Table 7–3.</td>
</tr>
<tr>
<td>Tag</td>
<td>Partition tag. For a description of this column, see Table 7–3.</td>
</tr>
<tr>
<td>Flags</td>
<td>Partition flag. For a description of this column, see Table 7–3.</td>
</tr>
<tr>
<td>First Sector</td>
<td>The first sector of the slice or partition.</td>
</tr>
<tr>
<td>Sector Count</td>
<td>The total number of sectors in the slice or partition.</td>
</tr>
<tr>
<td>Last Sector</td>
<td>The last sector of the slice or partition.</td>
</tr>
<tr>
<td>Mount Directory</td>
<td>The last mount point directory for the file system.</td>
</tr>
</tbody>
</table>

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 a Disk Slice for a ZFS Root File System” on page 210 or “x86: How to Create a Disk Slice for a ZFS Root File System” on page 225.
Managing Disk Use (Tasks)

This chapter describes how to optimize disk space by locating unused files and large directories.

This is a list of the information in this chapter:
- “Managing Disk Use (Task Map)” on page 167
- “Displaying Information About Files and Disk Space” on page 168
- “Checking the Size of Files” on page 171
- “Checking the Size of Directories” on page 174
- “Finding and Removing Old or Inactive Files” on page 176

Managing Disk Use (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display information about files and disk space.</td>
<td>Display information about how disk space is used by using the <code>df</code> command.</td>
<td>“How to Display Information About Files and Disk Space” on page 169</td>
</tr>
<tr>
<td>Display the size of files.</td>
<td>Display information about the size of files by using the <code>ls</code> command with the <code>-lh</code> options.</td>
<td>“How to Display the Size of Files” on page 171</td>
</tr>
<tr>
<td>Find large files.</td>
<td>The <code>ls -s</code> command allows you to sort files by size, in descending order.</td>
<td>“How to Find Large Files” on page 172</td>
</tr>
<tr>
<td>Find files that exceed a specified size limit.</td>
<td>Locate and display the names of files that exceed a specified size by using the <code>find</code> command with the <code>-size</code> option and the value of the specified size limit.</td>
<td>“How to Find Files That Exceed a Specified Size Limit” on page 173</td>
</tr>
</tbody>
</table>
### Displaying Information About Files and Disk Space

This table summarizes the commands available for displaying information about file size and disk space.

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Man Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>df</td>
<td>Reports the number of free disk blocks and files</td>
<td>df(1M)</td>
</tr>
<tr>
<td>du</td>
<td>Summarizes disk space allocated to each subdirectory</td>
<td>du(1)</td>
</tr>
<tr>
<td>find -size</td>
<td>Searches recursively through a directory based on the size specified with the -size option</td>
<td>find(1)</td>
</tr>
<tr>
<td>ls -lh</td>
<td>Lists the size of a file in the power of 1024 scaling</td>
<td>ls(1)</td>
</tr>
</tbody>
</table>
### Command Description Man Page

**zpool list**
Displays pool space size, how much space is allocated to datasets and internal metadata and how much space is unallocated.

**zfs list**
Displays amount of space consumed by datasets and any descendents, the amount of space available to the dataset and its descendents, and the amount of data that is referenced by this dataset, which may or may not be shared with other datasets in the pool.

The `zpool list` and `zfs list` commands are better than the legacy `df` and `du` commands for determining your available ZFS storage pool and file system space. With the legacy commands, you cannot easily discern between pool and file system space, nor do the legacy commands account for space that is consumed by descendant file systems or snapshots.

#### How to Display Information About Files and Disk Space

1. **Select the file system type:**
   - UFS
   - ZFS

2. **Display information about how UFS disk space is used.**

   ```bash
   $ df
   / (/dev/dsk/c0t0d0s0 ): 101294 blocks 105480 files
   /devices (/devices ): 0 blocks 0 files
   /system/contract (ctfs ): 0 blocks 2147483578 files
   /proc (/proc ): 0 blocks 1871 files
   /etc/mnttab (/mnttab ): 0 blocks 0 files
   /etc/svc/volatile (swap ): 992704 blocks 16964 files
   /system/object (objfs ): 0 blocks 2147483530 files
   /usr (/dev/dsk/c0t0d0s6 ): 503774 blocks 290189 files
   /dev/fd (/dev/fd ): 0 blocks 0 files
   /var/run (/var/run ): 992704 blocks 16964 files
   /tmp (/tmp ): 992704 blocks 16964 files
   /opt (/dev/dsk/c0t0d0s5 ): 23914 blocks 6947 files
   /export/home (/dev/dsk/c0t0d0s7 ): 16818 blocks 7160 files
   ``

3. **Display information about how ZFS disk space is used.**

   For example, the following root pool (`rpool`) has 10.0 GB allocated and 58.0 GB free.

   ```bash
   # zpool list rpool
   NAME SIZE ALLOC FREE CAP HEALTH ALTROOT
   rpool 68G 10.0G 58.0G 14% ONLINE -
   ```
If you compare the pool space accounting with the file system space accounting by reviewing the USED columns of your individual file systems minus some space for pool metadata, you can see that the pool space is accounted for. For example:

```
# zfs list -r rpool
NAME     USED   AVAIL  REFER MOUNTPOINT
rpool    10.2G  56.8G  106K /rpool
rpool/ROOT 5.04G  56.8G  31K legacy
rpool/ROOT/solaris 5.04G  56.8G  5.04G /
rpool/dump   1.00G  56.8G  1.00G -
rpool/export  63K   56.8G  32K /export
rpool/export/home 31K   56.8G  31K /export/home
rpool/swap    4.13G  56.9G  4.00G -
```

**Example 8–1** Displaying File Size Information in 1024 Bytes on a System With a UFS Root File System

In the following example, file system information for a system with a UFS root file system is displayed in 1024 bytes.

```
$ df -h
Filesystem size used avail capacity Mounted on
/dev/dsk/c0t0d0s0 249M 200M 25M 90% /
/devices 0K 0K 0K 0% /devices
tfs 0K 0K 0K 0% /system/contract
proc 0K 0K 0K 0% /proc
mnttab 0K 0K 0K 0% /etc/mnttab
swap 485M 376K 485M 1% /etc/svc/volatile
objs 0K 0K 0K 0% /system/object
/dev/dsk/c0t0d0s6 3.2G 2.9G 214M 94% /usr
fd 0K 0K 0K 0% /dev/fd
swap 485M 40K 485M 1% /var/run
swap 485M 40K 485M 1% /tmp
/dev/dsk/c0t0d0s5 13M 1.7M 10M 15% /opt
/dev/dsk/c0t0d0s7 9.2M 1.0M 7.3M 13% /export/home
```

Although /proc and /tmp are local file systems, they are not UFS file systems. /proc is a PROCFS file system, /var/run and /tmp are TMPFS file systems, and /etc/mnttab is an MNTFS file system.

**Example 8–2** Displaying Total Number of Blocks and Files Allocated for a UFS File System

The following example shows a list of all mounted file systems, device names, total 512-byte blocks used, and the number of files. The second line of each two-line entry displays the total number of blocks and files that are allocated for the file system.

```
$ df -t
/ (/dev/dsk/c0t0d0s0 ): 101294 blocks 105480 files
total: 509932 blocks 129024 files
/devices (/devices ): 0 blocks 0 files
total: 0 blocks 113 files
```
Checking the Size of Files

You can check the size of files and sort them by using the `ls` command. You can find files that exceed a size limit by using the `find` command. For more information, see `ls(1)` and `find(1).

Note – If you run out of space in the `/var` directory, do not symbolically link the `/var` directory to a directory on a file system with more disk space. Doing so, even as a temporary measure, might cause problems for certain daemon processes and utilities.

![How to Display the Size of Files](image)

1 Change to the directory where the files you want to check are located.

2 Display the size of the files.

```bash
$ ls [-lh] [-s]
```

- `l` Displays a list of files and directories in long format, showing the sizes in bytes. (See the example that follows.)

- `h` Scales file sizes and directory sizes into KB, MB, GB, or TB when the file or directory size is larger than 1024 bytes. This option also modifies the output displayed by the `-o`, `-n`, `-@`, and `-g` options to display file or directory sizes in the new format. For more information, see the `ls(1)` man page.

- `s` Displays a list of the files and directories, showing the sizes in blocks.
Example 8–3   Displaying the Size of Files

The following example shows that the `lastlog` and `messages` files are larger than the other files in the `/var/adm` directory.

```bash
$ cd /var/adm
$ ls -lh
```

```
total 148
drwxrwr-x  5 adm adm  512 Nov 26  09:39 acct/
drwxr-xr-x  2 adm adm  512 Nov 26  09:25 exactl/
-r--r--r--  1 root other 342K Nov 26  13:56 lastlog
drwxr-xr-x  2 adm adm  512 Nov 26  09:25 log/
-r--r--r--  1 root root  20K Nov 26  13:55 messages
drwxr-xr-x  2 adm adm  512 Nov 26  09:25 passwd/
drwxrwxr-x  2 adm sys  512 Nov 26  09:39 sa/
drwxr-xr-x  2 root sys  512 Nov 26  09:49 sm.bin/
-rw-r--r--  1 root bin  0   Nov 26  09:25 spellhist
drwxr-xr-x  2 root sys  512 Nov 26  09:25 streams/
-rw-r--r--  1 root bin  3.3K Nov 26  13:56 utmpx
-rw-r--r--  1 root root  0   Nov 26  10:17 vold.log
-rw-r--r--  1 adm adm  19K Nov 26  13:56 wtmpx
```

The following example shows that the `lpsched.1` file uses two blocks.

```bash
$ cd /var/lp/logs
$ ls -s
```

```
total 2
```

```
  0  lpsched
  2  lpsched.1
```

How to Find Large Files

1 Change to the directory that you want to search.

2 Display the size of files in blocks from largest to smallest.
   - If the characters or columns for the files are different, use the following command to sort a list of files by block size, from largest to smallest.

     ```bash
     $ ls -l | sort +4rn | more
     ```

     Note that this command sorts files in a list by the character that is in the fourth field, starting from the left.
   - If the characters or columns for the files are the same, use the following command to sort a list of files by block size, from largest to smallest.

     ```bash
     $ ls -s | sort -nr | more
     ```

     Note that this command sorts files in a list, starting with the left most character.
Example 8–4  Finding Large Files (Sorting by the Fifth Field’s Character)

$ cd /var/adm
$ ls -l | sort +4rn | more
-r-r--r-- 1 root root  4568368 Oct 17 08:36 lastlog
-rw-r--r-- 1 adm adm  697040 Oct 17 12:30 pacct.9
-rw-r--r-- 1 adm adm  280520 Oct 17 13:05 pacct.2
-rw-r--r-- 1 adm adm  277360 Oct 17 12:55 pacct.4
-rw-r--r-- 1 adm adm  264080 Oct 17 12:45 pacct.6
-rw-r--r-- 1 adm adm  255840 Oct 17 12:40 pacct.7
-rw-r--r-- 1 adm adm  254120 Oct 17 13:10 pacct.1
-rw-r--r-- 1 adm adm  250360 Oct 17 12:25 pacct.10
-rw-r--r-- 1 adm adm  248800 Oct 17 13:00 pacct.3
-rw-r--r-- 1 adm adm  247200 Oct 17 12:35 pacct.8
-rw-r--r-- 1 adm adm  246720 Oct 17 13:15 pacct.0
-rw-r--r-- 1 adm adm  245920 Oct 17 12:50 pacct.5
-rw-r--r-- 1 root root  190229 Oct  5 03:02 messages.1
-rw-r--r-- 1 adm adm  156800 Oct 17 13:17 pacct
-rw-r--r-- 1 adm adm  129084 Oct 17 08:36 wtmpx

Example 8–5  Finding Large Files (Sorting by the Left Most Character)

In the following example, the lastlog and messages files are the largest files in the /var/adm directory.

$ cd /var/adm
$ ls -s | sort -nr | more
48 lastlog
30 messages
24 wtmpx
18 pacct
 8 utmpx
 2 vold.log
 2 sulog
 2 sm.bin/
 2 sa/
 2 passwd/
 2 pacct1
 2 log/
 2 acct/
 0 spellhist
 0 aculog
total 144

▼ How to Find Files That Exceed a Specified Size Limit

- To locate and display the names of files that exceed a specified size, use the find command.

$ find directory -size +nnn
directory    Identifies the directory that you want to search.
-size +nnn    Is a number of 512-byte blocks. Files that exceed this size are listed.
Example 8–6  Finding Files That Exceed a Specified Size Limit

The following example shows how to find files larger than 400 blocks in the current working directory. The `-print` option displays the output of the `find` command.

```
$ find . -size +400 -print
./Howto/howto.doc
./Howto/howto.doc.backup
./Howto/howtotest.doc
./Routine/routineBackupconcepts.doc
./Routine/routineIntro.doc
./Routine/routineTroublefsck.doc
./.record
./Mail/pagination
./Config/configPrintadmin.doc
./Config/configPrintsetup.doc
./Config/configMailappx.doc
./Config/configMailconcepts.doc
./snapshot.rs
```

Checking the Size of Directories

You can display the size of directories by using the `du` command and options. Additionally, you can find the amount of disk space used by user accounts on local UFS file systems by using the `quot` command. For more information about these commands, see `du(1)` and `quot(1M)`.

▼ How to Display the Size of Directories, Subdirectories, and Files

- Display the size of one or more directories, subdirectories, and files by using the `du` command. Sizes are displayed in 512-byte blocks.

```
$ du [-as] [directory ...]
```

- `du` Displays the size of each directory that you specify, including each subdirectory beneath it.
- `-a` Displays the size of each file and subdirectory, and the total number of blocks that are contained in the specified directory.
- `-s` Displays the total number of blocks that are contained in the specified directory.
- `-h` Displays the size of each directory in 1024-byte blocks.
- `-H` Displays the size of each directory in 1000-byte blocks.
Identifies one or more directories that you want to check. Separate multiple directories in the command-line syntax with spaces.

**Example 8–7** Displaying the Size of Directories, Subdirectories, and Files

The following example shows the sizes of two directories:

```bash
$ du -s /var/adm /var/spool/lp
130 /var/adm
40 /var/spool/lp
```

The following example shows the sizes of two directories and includes the sizes of all the subdirectories and files that are contained within each directory. The total number of blocks that are contained in each directory is also displayed:

```bash
$ du /var/adm /var/spool/lp
2 /var/adm/exacct
2 /var/adm/log
2 /var/adm/streams
2 /var/adm/acct/fiscal
2 /var/adm/acct/nite
2 /var/adm/acct/sum
8 /var/adm/acct
2 /var/adm/sa
2 /var/adm/sm.bin
258 /var/adm
4 /var/spool/lp/admins
2 /var/spool/lp/requests/printing....
4 /var/spool/lp/requests
4 /var/spool/lp/system
2 /var/spool/lp/fifos
24 /var/spool/lp
```

The following example shows directory sizes in 1024-byte blocks:

```bash
$ du -h /usr/share/audio
796K /usr/share/audio/samples/au
797K /usr/share/audio/samples
798K /usr/share/audio
```

**How to Display the User Ownership of Local UFS File Systems**

1. Become superuser.

2. Display users, directories, or file systems, and the number of 1024-byte blocks used.
   
   ```bash
   # quot [-a] [filesystem ...]
   ```
-a  Lists all users of each mounted UFS file system and the number of 1024-byte blocks used.

_filesystem_  Identifies a UFS file system. Users and the number of blocks used are displayed for that file system.

**Note** – The `quot` command works only on local UFS file systems.

**Example 8–8**  Displaying the User Ownership of Local UFS File Systems

In the following example, users of the root (/) file system are displayed. In the subsequent example, users of all mounted UFS file systems are displayed.

```bash
# quot / 
/dev/rdsk/c0t0d0s0:
43340  root
3142   rimmer
 47   uucp
  35   lp
  30   adm
   4   bin
   4   daemon

# quot -a 
/dev/rdsk/c0t0d0s0 (/):
43340  root
3150   rimmer
 47   uucp
  35   lp
  30   adm
   4   bin
   4   daemon
/dev/rdsk/c0t0d0s6 (/usr):
460651  root
20632   bin
  791   uucp
   46   lp
   4   daemon
   1   adm
/dev/rdsk/c0t0d0s7 (/export/home):
   9   root
```

**Finding and Removing Old or Inactive Files**

Part of the job of cleaning up heavily loaded file systems involves locating and removing files that have not been used recently. You can locate unused files by using the `ls` or `find` commands. For more information, see `ls(1)` and `find(1)`. 
Other ways to conserve disk space include emptying temporary directories such as the directories located in `/var/tmp` or `/var/spool`, and deleting core and crash dump files. For more information about crash dump files, refer to Chapter 17, “Managing System Crash Information (Tasks),” in System Administration Guide: Advanced Administration.

### How to List the Newest Files

- List files, displaying the most recently created or changed files first, by using the `ls -t` command.

  ```
  $ ls -t [directory]
  
  -t Sorts files by latest time stamp first.
  
  directory Identifies the directory that you want to search.
  ```

**Example 8–9** Listing the Newest Files

The following example shows how to use the `ls -tl` command to locate the most recently created or changed files within the `/var/adm` directory. The `sulog` file was created or edited most recently.

```
$ ls -tl /var/adm

```

### How to Find and Remove Old or Inactive Files

1. Become superuser.

2. Find files that have not been accessed for a specified number of days and list them in a file.

   ```
   # find directory -type f [-atime +nnn] [-mtime +nnn] -print > filename
   ```
Finding and Removing Old or Inactive Files

`directory` 
Identifies the directory you want to search. Directories below this directory are also searched.

`-atime +nnn` 
Finds files that have not been accessed within the number of days (nnn) that you specify.

`-mtime +nnn` 
Finds files that have not been modified within the number of days (nnn) that you specify.

`filename` 
Identifies the file that contains the list of inactive files.

3 Remove the inactive files found listed in the previous step.
   
   `# rm 'cat filename'`
   
   where `filename` identifies the file that was created in the previous step. This file contains the list of inactive files.

Example 8–10 Finding and Removing Old or Inactive Files

The following example shows files in the `/var/adm` directory and the subdirectories that have not been accessed in the last 60 days. The `/var/tmp/deadfiles` file contains the list of inactive files. The `rm` command removes these inactive files.

```sh
# find /var/adm -type f -atime +60 -print > /var/tmp/deadfiles &
# more /var/tmp/deadfiles
/var/adm/aculog
/var/adm/spellhist
/var/adm/wtmpx
/var/adm/sa/sa13
/var/adm/sa/sa27
/var/adm/sa/sa11
/var/adm/sa/sa23
/var/adm/sulog
/var/adm/vold.log
/var/adm/messages.1
/var/adm/messages.2
/var/adm/messages.3
# rm 'cat /var/tmp/deadfiles'
#
```

▼ How to Clear Out Temporary Directories

1 Become superuser

2 Change to the directory that you want to clean out.
   
   `# cd directory`
Caution – Ensure that you are in the correct directory before completing Step 3. Step 3 deletes all files in the current directory.

3 Delete the files and subdirectories in the current directory.
   # rm -r *

4 Change to other directories that contain unnecessary, temporary or obsolete subdirectories and files.

5 Delete these subdirectories and files by repeating Step 3.

Example 8–11 Clearing Out Temporary Directories

The following example shows how to clear out the mywork directory, and how to verify that all files and subdirectories were removed.

   # cd mywork
   # ls
   filea.000
   fileb.000
   filec.001
   # rm -r *
   # ls

▼ How to Find and Delete core Files

1 Become superuser.
   Change to the directory where you want to search for core files.

2 Find and remove any core files in this directory and its subdirectories.
   # find . -name core -exec rm {} \\

Example 8–12 Finding and Deleting core Files

The following example shows how to find and remove core files from the jones user account by using the find command.

   # cd /home/jones
   # find . -name core -exec rm {} \

Finding and Removing Old or Inactive Files
How to Delete Crash Dump Files

Crash dump files can be very large. If you have enabled your system to store these files, do not retain them for longer than necessary.

1  **Change to the directory where crash dump files are stored.**
   # cd /var/crash/system
   where system identifies a system that created the crash dump files.

   **Caution** – Ensure you are in the correct directory before completing Step 3. Step 3 deletes all files in the current directory.

2  **Remove the crash dump files.**
   # rm *

3  **Verify that the crash dump files were removed.**
   # ls

**Example 8–13 Deleting Crash Dump Files**

The following example shows how to remove crash dump files from the system venus, and how to verify that the crash dump files were removed.

# cd /var/crash/venus
# rm *
# ls
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 Oracle Solaris.

This is a list of the information in this chapter:

- "Administering Disks (Task Map)" on page 181
- "Identifying Disks on a System" on page 182
- "Formatting a Disk" on page 184
- "Displaying Disk Slices" on page 186
- "Creating and Examining a Disk Label" on page 189
- "Recovering a Corrupted Disk Label" on page 195
- "Adding a Third-Party Disk" on page 197

For overview information about disk management, see Chapter 7, "Managing Disks (Overview)."

## Administering Disks (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identify the disks on a system.</td>
<td>If you are not sure of the types of disks on a system, use the <code>format</code> utility to identify the disk types.</td>
<td>“How to Identify the Disks on a System” on page 182</td>
</tr>
<tr>
<td>Format the disk.</td>
<td>Determine whether a disk is already formatted by using the <code>format</code> utility.</td>
<td>“How to Determine if a Disk Is Formatted” on page 184</td>
</tr>
</tbody>
</table>
### 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 detailed information on using the `format` utility, see Chapter 13, “The format Utility (Reference).”

#### How to Identify the Disks on a System

1. Become superuser.

2. Identify the disks that are recognized on the system by using the `format` utility.

   ```
   # format
   ``

   The format utility displays a list of disks that it recognizes under AVAILABLE DISK SELECTIONS.

#### Example 9–1 Identifying the Disks on a System

The following example shows `format` command output.

```
# format
AVAILABLE DISK SELECTIONS:
  0. c2t0d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
     /pci@1c,600000/scsi@2/sd@0,0
  1. c2t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
     /pci@1c,600000/scsi@2/sd@1,0
  2. c2t2d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
     /pci@1c,600000/scsi@2/sd@2,0
  3. c2t3d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
```
The 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 “Device Naming Conventions” on page 62. The following example uses a wildcard to display the four disks that are connected to a controller 0:

```
# format /dev/rdsk/c0t6*
AVAILABLE DISK SELECTIONS:
  0. /dev/rdsk/c0t600A0B800022024E000054AC4970A629d0p0 <...>
     /scsi_vhci/disk@g600a0b800022024e000054ac4970a629
  1. /dev/rdsk/c0t600A0B800022024E000054AE4970A711d0p0 <...>
     /scsi_vhci/disk@g600a0b800022024e000054ae4970a711
  2. /dev/rdsk/c0t600A0B800022028A00005444970A834d0p0 <...>
     /scsi_vhci/disk@g600a0b800022028a000054444970a834
  3. /dev/rdsk/c0t600A0B800022028A00005454970A8EAd0p0 <...>
     /scsi_vhci/disk@g600a0b800022028a000054544970a8ea
```

Specify disk (enter its number):

The following example shows how to identify the disk information:

```
# format 0. c2t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
   /pci@1c,600000/scsi@2/sdq@0,0
```

The output identifies that disk 0 (target 0) is connected to the second SCSI host adapter (scsi@2), which is connected to the second PCI interface (/pci@1c, 600000/...). The output also associates both the physical and logical device name to the disk’s marketing name, SUN36G.

Some disks do not have a marketing name. If the format output does not identify disks by their marketing names, then you can use the format utility’s type and label features as described in the following steps to include the disk’s marketing name.

The following steps must be done while the disk or system is inactive, which means booting from an installation DVD or the network, unless the disk is currently unused and it will not contain the Oracle Solaris release. In addition, the final step is to relabel the disk, which can remove any existing partition information or data.

```
# format
Searching for disks...done
AVAILABLE DISK SELECTIONS:
  0. c2t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
     /pci@1c,600000/scsi@2/sdq@0,0
  1. c2t1d0 <SEAGATE-ST336607LSUN36G-0307-33.92GB>
     /pci@1c,600000/scsi@2/sdq@1,0
  2. c2t2d0 <SEAGATE-ST336607LSUN36G-0507-33.92GB>
     /pci@1c,600000/scsi@2/sdq@2,0
```
3. c2t3d0 <drive type unknown> /pci@1c,600000/scsi@2/sd@3,0
   Specify disk (enter its number): 3
   selecting c2t3d0
   [disk formatted]
   format> type
   AVAILABLE DRIVE TYPES:
   0. Auto configure
   1. other
   Specify disk type (enter its number): 0
c2t3d0: configured with capacity of 33.92GB
   <SEAGATE-ST336607LSUN36G-0507-33.92GB>
   [disk formatted]
   format> label
   Ready to label disk, continue? yes
   format> quit

More Information
If the format Utility Does Not Recognize a Disk ...

- Go to Chapter 10, "SPARC: Setting Up Disks (Tasks)," or "x86: Setting Up Disks for ZFS File
  Systems (Task Map)" on page 222.
- Go to “How to Label a Disk” on page 189.
- Connect the disk to the system by using your disk hardware documentation.

Formatting a Disk

Disks are typically 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 the disk. However, most disks are already formatted.
- Use the Oracle Solaris installation utility 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.

2. Invoke the format utility.
   # format
A numbered list of disks is displayed.

3 **Type the number of the disk that you want to check.**

Specify disk (enter its number): 0

4 **Verify that the disk you chose is formatted by noting the following message:**

[disk formatted]

**Example 9–2**  Determining if a Disk Is Formatted

The following example shows that disk `c2t3d0` is formatted.

```
# format
AVAILABLE DISK SELECTIONS:
  0. c2t0d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
     /pci@1c,600000/scsi@2/sd@0,0
  1. c2t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
     /pci@1c,600000/scsi@2/sd@1,0
  2. c2t2d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
     /pci@1c,600000/scsi@2/sd@2,0
  3. c2t3d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
     /pci@1c,600000/scsi@2/sd@3,0
Specify disk (enter its number): 3
selecting c2t3d0
[disk formatted]
```

**How to Format a Disk**

Disks are formatted by the manufacturer. Reformating a disk should occur rarely. The process is time-consuming and removes all data from the disk.

1 **Become superuser.**

2 **Invoke the format utility.**

```
# format
```

A numbered list of disks is displayed.

3 **Type the number of the disk that you want to format.**

Specify disk (enter its number): 3

---

**Caution** – Do not select the disk that contains the root file system. If you format a root pool disk, you delete the OS 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
    The protection information is not enabled
    The disk will be formatted with protection type 0

    Ready to format. Formatting cannot be interrupted
    and takes 169 minutes (estimated). Continue? yes

5 Verify that the disk format was successful by noting the following messages:

    Beginning format. The current time is Fri Apr 1 ...
    Formatting...
    done
    Verifying media...
    pass 0 - pattern = 0xc6dec6de
    14086/23/734
    pass 1 - pattern = 0x6db6db6d
    14086/23/734
    Total of 0 defective blocks repaired.

6 Exit the `format` utility.

    format> quit

Displaying Disk Slices

The best way to create ZFS storage pools is to use whole disks instead of disk slices because whole disks are easier to manage. The only time you need to use a disk slice is when the disk is intended for the ZFS root pool. This is a long-standing boot limitation. For non-root pools, use whole disks. When you create a pool with whole disks, an EFI label is applied. See the EFI disk label example that follows.

If you need to prepare a disk for use as a root pool disk, create a slice 0 that contains the entire disk capacity, as shown in the VTOC disk label example that follows.

For information about setting up disks for use with ZFS storage pools, see Chapter 10, "SPARC: Setting Up Disks (Tasks)," or “x86: Setting Up Disks for ZFS File Systems (Task Map)” on page 222.

Note – The `format` utility uses the term `partition` instead of `slice`. 
How to Display Disk Slice Information

You might need to display disk slice information if the disk is intended to be used for the ZFS root pool. It must also include a SMI label.

1. Become superuser.

2. Invoke the `format` utility.
   ```
   # format
   ```
   A numbered list of disks is displayed.

3. Type the number of the disk for which you want to display slice information.
   ```
   Specify disk (enter its number): 1
   ```

4. Select the partition menu.
   ```
   format> partition
   ```

5. Display the slice information for the selected disk.
   ```
   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.

Example 9–3

Displaying Disk Slice Information

The following example displays slice information for a disk with a VTOC label.

```
# format
Searching for disks...done
Specify disk (enter its number): 3
Selecting c2t3d0
format> partition
partition> print
Current partition table (c2t3d0):
Total disk cylinders available: 14087 + 2 (reserved cylinders)
<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 14086</td>
<td>136.71GB</td>
<td>(14087/0/0) 286698624</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 14086</td>
<td>136.71GB</td>
<td>(14087/0/0) 286698624</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0)</td>
</tr>
</tbody>
</table>
```
Displaying Disk Slices

For a detailed description of the slice information in these examples, see Chapter 7, "Managing Disks (Overview)."

The following example shows the slice information for a disk with an EFI label:

```
# format
Searching for disks...done
Specify disk (enter its number): 3
selecting c2t3d0
[disk formatted]
format> partition
partition> print
Current partition table (default):
Total disk sectors available: 286722878 + 16384 (reserved sectors)

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>First Sector</th>
<th>Size</th>
<th>Last Sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>usr</td>
<td>wm</td>
<td>34</td>
<td>136.72GB</td>
<td>286722911</td>
</tr>
<tr>
<td>1</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>reserved</td>
<td>wm</td>
<td>286722912</td>
<td>8.00MB</td>
<td>286739295</td>
</tr>
</tbody>
</table>
```

```
partition> q
format> q
```

The following example shows the slice information for an EFI (GPT) labeled disk for a root pool:

```
# format
Searching for disks...done
Specify disk (enter its number): 0
selecting c2t0d0
[disk formatted]
format> partition
partition> print
Current partition table (original):
Total disk sectors available: 27246525 + 16384 (reserved sectors)

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>First Sector</th>
<th>Size</th>
<th>Last Sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>BIOS_boot</td>
<td>wm</td>
<td>256</td>
<td>256.00MB</td>
<td>524543</td>
</tr>
<tr>
<td>1</td>
<td>usr</td>
<td>wm</td>
<td>524544</td>
<td>12.74GB</td>
<td>27246558</td>
</tr>
<tr>
<td>2</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>reserved</td>
<td>wm</td>
<td>27246559</td>
<td>8.00MB</td>
<td>27262942</td>
</tr>
</tbody>
</table>
```

```
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 using new disks. 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 similar to the following:

```
c2t3d0: configured with capacity of 136.73GB
```

▼ How to Label a Disk

If you want to put an EFI label on disk smaller than 2 terabytes, see Example 9–5.

1 Become superuser.

2 Invoke the `format` utility.

   `# format`

   A numbered list of disks is displayed.

3 **Type the number of the disk that you want to label.**

   `Specify disk (enter its number):`

   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.

4 **Select one of the following to label the disk:**

   - 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.
   
   - If the disk is labeled but you want to change the disk type, or if the `format` utility was not able to automatically configure the disk, proceed to Step 6 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.

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.
   format> q

Example 9–4 Labeling a Disk

The following example shows how to automatically configure and label a 36-GB disk.

# format
AVAILABLE DISK SELECTIONS:
   0. c0t3d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
      /pci@1c,600000/scsi@2/sd@0,0
   1. c0t1d0 <SEAGATE-ST336607LSUN36G-0307-33.92GB>
      /pci@1c,600000/scsi@2/sd@1,0
   2. c0t2d0 <SEAGATE-ST336607LSUN36G-0507-33.92GB>
      /pci@1c,600000/scsi@2/sd@2,0
   3. c0t3d0 <SEAGATE-ST336607LSUN36G-0507-33.92GB>
      /pci@1c,600000/scsi@2/sd@3,0
Specify disk (enter its number): 3
   c0t3d0: configured with capacity of 33.92GB
   Disk not labeled. Label it now? yes
   format> verify
   format> q

Example 9–5 Labeling a Disk With an EFI Label

The following example shows how to use the format -e command to label a disk with an EFI label or an SMI (VTOC) 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 157.

# format -e
Searching for disks...done
### Example 9–6: Change a EFI-Labeled Disk to an SMI-Labeled Disk

The following example shows how to use the *format -e* command to change an EFI labeled disk to an SMI-labeled disk that can be used for a ZFS root pool.

On an x86 based system, you will first have to change the EFI *fdisk* partition to a Solaris partition. For example:

```bash
# format -e
select disk ...
format> fdisk
FORMAT MENU:
   disk - select a disk
type - select (define) a disk type
   Total disk size is 17833 cylinders
   Cylinder size is 16065 (512 byte) blocks

+---+-----------------+---------------------+---------+---------+------------+---------+
| 1 | EFI             | 0                  | 17833   | 17834   | 100        |         |
+---+-----------------+---------------------+---------+---------+------------+---------+
```

**SELECT ONE OF THE FOLLOWING:**

1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Edit/View extended partitions
6. Exit (update disk configuration and exit)
7. Cancel (exit without updating disk configuration)

Enter Selection: 3
Specify the partition number to delete (or enter 0 to exit): 1
This will make all files and programs in this partition inaccessible (type "y" or "n"). y
Enter Selection: 1
Select the partition type to create:
1=SOLARIS2  2=UNIX  3=PCIXOS  4=Other  5=DOS12
6=DOS16  7=DOSEXT  8=DOSBIG  9=DOS16LBA  A=x86 Boot
B=Diagnostic  C=FAT32  D=FAT32LBA  E=DOSEXTLBA  F=EFI (Protective)
G=EFI_SYS  0=Exit? 1
Specifying the percentage of disk to use for this partition
(or type "c" to specify the size in cylinders). 100
Should this become the active partition? If yes, it will be activated
each time the computer is reset or turned on.
Please type "y" or "n". y
Enter Selection: 6
Partition 1 is now the active partition.

On a SPARC based system, follow these steps:

```
# format -e
Searching for disks...done
AVAILABLE DISK SELECTIONS:
 0. c2t0d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
    /pci@1c,600000/scsi@2/sd@0,0
 1. c2t1d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
    /pci@1c,600000/scsi@2/sd@1,0
 2. c2t2d0 <FUJITSU-MAP3147N SUN146G-0501-136.73GB>
    /pci@1c,600000/scsi@2/sd@2,0
 3. c2t3d0 <FUJITSU-MAP3147N SUN146G-0501-136.73GB>
    /pci@1c,600000/scsi@2/sd@3,0
Specify disk (enter its number): 3
selecting c2t0d0
[Disk formatted]
format> label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0
Ready to label disk, continue? yes
```

You should also check the default partition table to ensure that it is optimal for a root pool slice,
which means that the bulk of the disk space is in slice 0. See the steps below to increase the size
of slice 0.

```
format> partition
partition> print
Current partition table (default):
Total disk cylinders available: 14086 + 2 (reserved cylinders)

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>1 - 13</td>
<td>129.16MB</td>
<td>(13/0/0) 264576</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>14 - 26</td>
<td>129.16MB</td>
<td>(13/0/0) 264576</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 14086</td>
<td>136.71GB</td>
<td>(14087/0/0) 286698624</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wms</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wms</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wms</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wms</td>
<td>27 - 14084</td>
<td>136.43GB</td>
<td>(14085/0/0) 286108416</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wms</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>8</td>
<td>boot</td>
<td>wu</td>
<td>0 - 0</td>
<td>0.94MB</td>
<td>(1/0/0) 28352</td>
</tr>
<tr>
<td>9</td>
<td>alternates</td>
<td>wms</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
</tbody>
</table>
```
partition> modify
Select partitioning base:
  0. Current partition table (default)
  1. All Free Hog
Choose base (enter number) [0]? 1

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 14084</td>
<td>136.69GB</td>
<td>(14084/0/0) 286657920</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>8</td>
<td>boot</td>
<td>wu</td>
<td>0 - 0</td>
<td>9.94MB</td>
<td>(1/0/0) 20352</td>
</tr>
<tr>
<td>9</td>
<td>alternates</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
</tbody>
</table>

Do you wish to continue creating a new partition table based on above table [yes]? yes
Free Hog partition[6]? 0
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]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>1 - 14084</td>
<td>136.68GB</td>
<td>(14084/0/0) 286637568</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 14084</td>
<td>136.69GB</td>
<td>(14084/0/0) 286657920</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>8</td>
<td>boot</td>
<td>wu</td>
<td>0 - 0</td>
<td>9.94MB</td>
<td>(1/0/0) 20352</td>
</tr>
<tr>
<td>9</td>
<td>alternates</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
</tbody>
</table>

Okay to make this the current partition table [yes]? yes
Enter table name (remember quotes): "c2t0d0"
Ready to label disk, continue? yes
partition> quit
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 7, “Managing Disks (Overview).”

1. Become superuser.
2 Display the disk label information.
   
   # prtvtoc /dev/rdsk/device-name
   
   where *device-name* is the raw disk device you want to examine.

Example 9–7 Examining a Disk Label

The following example shows disk label information for a disk with a VTOC label:

```
# prtvtoc /dev/rdsk/c2t3d0s0
* /dev/rdsk/c2t3d0s0 partition map
* Dimensions:
  * 512 bytes/sector
  * 848 sectors/track
  * 24 tracks/cylinder
  * 20352 sectors/cylinder
  * 14089 cylinders
  * 14087 accessible cylinders
* Flags:
  * 1: unmountable
  * 10: read-only
  *

  First   Sector  Last
  Partition Tag  Flags  Sector  Count  Sector  Mount Directory
  0  2     00  0 286698624  286698623
  2  5     01  0 286698624  286698623
```

The following example shows disk label information for a non-root pool disk with an EFI label:

```
# prtvtoc /dev/dsk/c8t3d0
* /dev/dsk/c8t3d0 partition map
* Dimensions:
  * 512 bytes/sector
  * 143374738 sectors
  * 143374671 accessible sectors
* Flags:
  * 1: unmountable
  * 10: read-only
  *
  Unallocated space:
  *
  Unallocated space:
  *
```
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 re-created 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 Oracle Solaris DVD or the network in single-user mode to access the disk.

   For information on booting the system, see *Oracle Solaris Administration: Basic Administration*.

2. **Relabel the disk.**

   ```
   # format
   ```

   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 this message:

   ```
   cwtxdy: configured with capacity of abcMB
   ```

   The `format` utility then displays a numbered list of disks on the system.

3. **Type the number of the disk that you need to recover.**

   Specify disk (enter its number): 1

4. **Select one of the following to determine how to label the disk.**

   - If the disk was configured successfully, do the following:

     a. **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.
        ```
Recovering a Corrupted Disk Label

Backup label contents:
Volume name = < >
ascii name = <FUJITSU-MAP3147N SUN146G-0501>
ncyl = 14089
ncyl = 14087
ncyl = 2
ncyl = 24
nsect = 848

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 14086</td>
<td>136.71GB</td>
<td>(14087)</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 14086</td>
<td>136.71GB</td>
<td>(14087)</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

b. If the format utility was able to find a backup label, and the backup label contents appear satisfactory, 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

c. Exit the format utility.

    format> q

- If the disk was not configured successfully, do the following:

a. Specify the disk type by using the type command:

    format> type
    The Available Drives Type menu is displayed.

b. 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

c. 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

d. 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.
e. If the format utility was able to find a backup label, and the backup label contents appear satisfactory, 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.
```

f. Exit the format utility.

```
format> q
```

5 Verify the file systems on the recovered disk.

For information on using the zpool scrub command for ZFS file systems, see zpool(1M).

---

**Adding a Third-Party Disk**

Oracle Solaris supports many third-party disks. However, for the disk to be recognized, you might need to supply a device driver.

Other options for adding disks are as follows:

- If you are adding a SCSI disk, you might try the format utility's automatic configuration feature.
- You might try hot-plugging a PCI, SCSI, or USB disk. For more information, see Chapter 3, “Managing Devices (Tasks).”

**Note** – Oracle cannot guarantee that its format utility will work properly with all third-party disk drivers. If the disk driver is not compatible with the format utility, the disk drive vendor should supply you with a custom disk formatting program.

Typically, you discover that software support is missing when you invoke the format utility and find that the disk type is not recognized.

Refer to the appropriate configuration procedure for adding system disks or secondary disks in Chapter 10, “SPARC: Setting Up Disks (Tasks),” or “x86: Setting Up Disks for ZFS File Systems (Task Map)” on page 222.
This chapter describes how to set up disks on a SPARC system.

For information on the procedures associated with setting up disks on a SPARC system, see the following sections:

- “SPARC: Setting Up Disks for UFS File Systems (Task Map)” on page 199
- “SPARC: Setting up Disks for ZFS File Systems (Task Map)” on page 207

For overview information about disk management, see Chapter 7, “Managing Disks (Overview).” For step-by-step instructions on setting up disks on an x86 based system, see Chapter 11, “x86: Setting Up Disks (Tasks).”

**SPARC: Setting Up Disks for UFS File Systems (Task Map)**

The following task map identifies the procedures for setting up a disk for a UFS root file system or a disk for a UFS file system on a SPARC based system.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
</table>
| 1. Connect the disk. | **Disk for a UFS root file system**  
Connect the new disk and boot from a local or remote Oracle Solaris DVD.  
**Disk for a UFS file system**  
Connect the disk. | “SPARC: How to Set Up a Disk for a UFS Root File System” on page 200  
“SPARC: How to Set Up a Disk for a UFS File System” on page 201 |
| 2. Create slices and label the disk for UFS file systems. | Create disk slices and label the disk. | “SPARC: How to Create Disk Slices for a UFS File System” on page 202 |
Task Description For Instructions

3. Create UFS filesystems. Create UFS filesystems on the disk slices by using the newfs command. You must re-create the root (/) or /usr file system, or both. “SPARC: How to Create a UFS File System” on page 206

4. Restore UFS file systems. Restore the UFS root (/) or /usr file system, or both. If necessary, restore file systems on other disks. “Restoring UFS Files and File System Backups (Task Map)” on page 369

5. Install boot blocks for a UFS root file system. Install the boot blocks on the root (/) file system so that the system can boot. “SPARC: How to Install Boot Blocks for a UFS Root File System” on page 207

SPARC: Setting Up Disks for UFS File Systems

A disk can contain the UFS root (/) and /usr file systems, user or data file systems, or a combination of file systems. If the disk contains the UFS root (/) or /usr file systems, you have two ways to recover:

- You can reinstall the entire Oracle Solaris OS.
- Or, you can replace the disk and restore your Oracle Solaris OS file systems from a backup medium.

A disk that contains user or data file systems can be replaced and the file systems restored, if necessary, without having to reinstall or reboot the system.

▼ SPARC: How to Set Up a Disk for a UFS Root File System

You can use this procedure to replace a disk that contained the UFS root or /usr file system. Refer to your hardware installation guide for information on replacing a disk.

1. Disconnect the damaged system disk from the system, if necessary.

2. Connect the replacement disk to the system and check the disk’s physical connections.

3. Follow the instructions in the following table, depending on whether you are booting from a local Oracle Solaris DVD or a remote Oracle Solaris DVD from the network.
<table>
<thead>
<tr>
<th>Boot Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>From an Oracle Solaris DVD in a local drive</td>
<td>1. Make sure the Oracle Solaris DVD is in the drive.</td>
</tr>
<tr>
<td></td>
<td>2. Boot from the media to single-user mode:</td>
</tr>
<tr>
<td></td>
<td>ok boot cdrom -s</td>
</tr>
<tr>
<td>From the network</td>
<td>Boot from the network to single-user mode:</td>
</tr>
<tr>
<td></td>
<td>ok boot net -s</td>
</tr>
</tbody>
</table>

After a few minutes, the root prompt (#) is displayed.

More Information  
**After You Set Up a Disk for a UFS Root File System ...**

Create slices for the UFS root and /usr file systems and a disk label on the disk. Go to “SPARC: How to Create Disk Slices for a UFS File System” on page 202.

▼ **SPARC: How to Set Up a Disk for a UFS File System**

If you are adding a disk with an EFI disk label, see “EFI (GPT) Disk Label” on page 156 for more information.

Generally, most modern bus types support hot-plugging. This means you can insert a disk in an empty slot and the system recognizes it. For more information about hot-plugging devices, see Chapter 4, “Dynamically Configuring Devices (Tasks).”

1  Become superuser or assume an equivalent role.

2  (Optional) 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.

3  Connect the disk to the system and check the disk’s physical connections.  
Refer to the disk’s hardware installation guide for details.

4  Use the format utility to confirm that the new disk is recognized.

More Information  
**After You Connect a Disk for a UFS File System ...**

After the disk is connected, create slices and a disk label on the disk. Go to “SPARC: How to Create Disk Slices for a UFS File System” on page 202.
**SPARC: How to Create Disk Slices for a UFS File System**

1. Become superuser or assume an equivalent role.

2. Invoke the `format` utility.
   
   ```
   # format
   ```
   A numbered list of available disks is displayed.

3. Select the number of the disk that you want to repartition.
   
   ```
   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 165.

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 Example 10–1.

10. Make the displayed partition table the current partition table by answering y when prompted.

   ```
   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"
where 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 utility.
format> q

Example 10–1 SPARC: Creating Disk Slices for a UFS Root File System

The following example shows the format utility being used to divide a 18-GB 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
AVAILABLE DISK SELECTIONS:
  0. /dev/rdsk/c1t0d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@0,0
  1. /dev/rdsk/c1t1d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@1,0
  2. /dev/rdsk/c1t8d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@8,0
  3. /dev/rdsk/c1t9d0s0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
    /sbus@2,0/QLGC,isp@2,10000/sd@9,0
Specify disk (enter its number): 0
selecting c1t0d0
[disk formatted]
format> partition
partition> print
partition> modify
Select partitioning base:
  0. Current partition table (original)
  1. All Free Hog
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 - 7505  16.86GB  (7506/0/0)  35368272
   3 unassigned  wu  0  0  (0/0/0)  0
   4 unassigned  wu  0  0  (0/0/0)  0
   5 unassigned  wu  0  0  (0/0/0)  0
   6  usr  wu  0  0  (0/0/0)  0
```
Choose base (enter number) [0]? 1
Table based on above table? [yes]
yes
Free Hog partition? [6]? 6
Enter size of partition '0' [0b, 0c, 0.00mb, 0.00gb]: 4gb
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]: 4gb
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 1780</td>
<td>4.00GB</td>
<td>(1781/0/0) 8392072</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>1781 - 3561</td>
<td>4.00GB</td>
<td>(1781/0/0) 8392072</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 7505</td>
<td>16.86GB</td>
<td>(7506/0/0) 35368272</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wu</td>
<td>3562 - 7505</td>
<td>8.86GB</td>
<td>(3944/0/0) 18584128</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
</tbody>
</table>

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

Example 10–2 SPARC: Creating Disk Slices for a UFS File System

The following example shows the format utility being used to divide a 18-GB disk into one slice for the /export/home file system.

```
# format /dev/rdsk/c1t0d0
AVAILABLE DISK SELECTIONS:
  0. /dev/rdsk/c1t0d0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus2,0/QLGC,ispq2,10000/sdq0,0
  1. /dev/rdsk/c1t1d0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus2,0/QLGC,ispq2,10000/sdq1,0
  2. /dev/rdsk/c1t8d0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus2,0/QLGC,ispq2,10000/sdq8,0
  3. /dev/rdsk/c1t9d0 <SUN18G cyl 7506 alt 2 hd 19 sec 248>
     /sbus2,0/QLGC,ispq2,10000/sdq9,0
Specify disk (enter its number): 1
selecting c1t1d0
[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         |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 7505</td>
<td>16.86GB</td>
<td>(7506/0/0) 35368272</td>
</tr>
</tbody>
</table>
```
The following example shows how to use the format utility to divide a 1.15 terabyte disk with an EFI label into three 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  root  wm     0         0          0
   1  usr  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
   7  unassigned  wm  0         0          0
   8  reserved  wm  2579624638 8.00MB  2579641021

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]:

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>First Sector</th>
<th>Size</th>
<th>Last Sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>usr</td>
<td>wm</td>
<td>34</td>
<td>400.00GB</td>
<td>838860833</td>
</tr>
<tr>
<td>3</td>
<td>usr</td>
<td>wm</td>
<td>838860834</td>
<td>400.00GB</td>
<td>1677721633</td>
</tr>
<tr>
<td>4</td>
<td>usr</td>
<td>wm</td>
<td>1677721634</td>
<td>428.77GB</td>
<td>2576924637</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>reserved</td>
<td>wm</td>
<td>2576924638</td>
<td>8.00MB</td>
<td>2576941021</td>
</tr>
</tbody>
</table>

Ready to label disk, continue? yes

partition> q

More Information
After You Create Disk Slices and Label a Disk for UFS File Systems...

After you create disk slices and label the disk, you can create file systems on the disk. Go to “SPARC: How to Create a UFS File System” on page 206.

▼ SPARC: How to Create a UFS File System

1 Become superuser or assume an equivalent role.

2 Create a file system for each slice.
   
   # newfs /dev/rdsk/cwtxdysz
   
   where /dev/rdsk/cwtxdysz is the raw device for the file system to be created.

   For more information about the newfs command, see Chapter 15, "Creating and Mounting File Systems (Tasks),” or newfs(1M).

3 Verify the new file system by mounting it.
   
   # mount /dev/dsk/cwtxdysz /mnt
   # ls
   lost+found

More Information
After Creating a UFS File System ...

- Disk for a UFS root file system – You need to restore the root (/) and /usr file systems on the disk.
  - Go to Restoring UFS Files and File System Backups (Task Map).
  - After the root (/) and /usr file systems are restored, install the boot block. Go to “SPARC: How to Install Boot Blocks for a UFS Root File System” on page 207.
- **Disk for a UFS file system** – You might need to restore file systems on the new disk. Go to “Restoring UFS Files and File System Backups (Task Map)” on page 369. If you are not restoring file systems on the new disk, you are finished adding a secondary disk.

- For information on making the UFS file systems available to users, see “Mounting and Unmounting Oracle Solaris File Systems” on page 286.

▼ **SPARC: How to Install Boot Blocks for a UFS Root File System**

1. Become superuser or assume an equivalent role.

2. Install a boot block for a UFS root file system.
   
   ```bash
   # installboot /usr/platform/`uname -i`/lib/fs/ufs/bootblk /dev/rdsk/cwtxdy0
   ```
   
   For more information, see `installboot(1M)`.

3. Verify that the boot blocks are installed by rebooting the system to run level 3.
   
   ```bash
   # init 6
   ```

**Example 10–3**  
**SPARC: Installing a Boot Block for a UFS Root File System**

The following example shows how to install the boot block on a UFS root file system.

```bash
# installboot /usr/platform/sun4u/lib/fs/ufs/bootblk /dev/rdsk/c0t0d0s0
```

**SPARC: Setting up Disks for ZFS File Systems (Task Map)**

The following task map identifies the procedures for setting up a ZFS root pool disk for a ZFS root file system or a non-root ZFS pool disk on a SPARC based system.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
</table>
| 1. Set up the disk for a ZFS root file system. | *Disk for a ZFS Root File System*  
Connect the new disk or replace the existing root pool disk and boot from a local or remote Oracle Solaris DVD. | “SPARC: How to Set Up a Disk for a ZFS Root File System” on page 209 |
| 2. Create a disk slice for a ZFS root file system | Create a disk slice for a disk that is intended for a ZFS root pool. This is a long-standing boot limitation. | “SPARC: How to Create a Disk Slice for a ZFS Root File System” on page 210 |
3. Install the boot blocks for a ZFS root file system, if necessary.
   If you replace a disk that is intended for the root pool by using the `zpool replace` command, then you must install the boot blocks manually so that the system can boot from the replacement disk.
   "SPARC: How to Install Boot Blocks for a ZFS Root File System" on page 213

4. Set up a disk for ZFS file system.
   Disk for a ZFS File System
   Set up a disk for a ZFS file system.
   "SPARC: How to Set Up a Disk for a ZFS File System" on page 214

### SPARC: Setting Up Disks for ZFS File Systems

Although the procedures that describe how to set up a disk can be used with a ZFS file system, a ZFS file system is not directly mapped to a disk or a disk slice. You must create a ZFS storage pool before creating a ZFS file system. For more information, see *Oracle Solaris ZFS Administration Guide*

The root pool contains the root file system that is used to boot the Oracle Solaris OS. If a root pool disk becomes damaged and the root pool is not mirrored, the system might not boot. If a root pool disk becomes damaged, you have two ways to recover:

- You can reinstall the entire Oracle Solaris OS.
- Or, you can replace the root pool disk and restore your file systems from snapshots or from a backup medium. You can reduce system down time due to hardware failures by creating a redundant root pool. The only supported redundant root pool configuration is a mirrored root pool.

A disk that is used in a non-root pool usually contains space for user or data files. You can attach or add another disk to a root pool or a non-root pool for more disk space. Or, you can replace a damaged disk in a pool in the following ways.

- A disk can be replaced in a non-redundant pool if all the devices are currently ONLINE.
- A disk can be replaced in a redundant pool if enough redundancy exists among the other devices.
- In a mirrored root pool, you can replace a disk or attach a disk and then detach the failed disk or a smaller disk to increase a pool’s size.

In general, setting up a disk on the system depends on the hardware so review your hardware documentation when adding or replacing a disk on your system. If you need to add a disk to an existing controller, then it might just be a matter of inserting the disk in an empty slot, if the system supports hot-plugging. If you need to configure a new controller, see "Dynamic Reconfiguration and Hot-Plugging" on page 67.
SPARC: How to Set Up a Disk for a ZFS Root File System

Refer to your hardware installation guide for information on replacing a disk.

1. Disconnect the damaged disk from the system, if necessary.

2. Connect the replacement disk to the system and check the disk’s physical connections, if necessary.

3. Follow the instructions in the following table, depending on whether you are booting from a local Oracle Solaris DVD or a remote Oracle Solaris DVD from the network.

<table>
<thead>
<tr>
<th>Boot Type</th>
<th>Action</th>
</tr>
</thead>
</table>
| From an Oracle Solaris DVD in a local drive | 1. Make sure the Oracle Solaris 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.

More Information

After You Set Up a Disk for a ZFS Root File System...

After the disk is connected or replaced, you can create a slice and update the disk label. Go to “SPARC: How to Create a Disk Slice for a ZFS Root File System” on page 210.

SPARC: Creating a Disk Slice for a ZFS Root File System

You must create a disk slice for a disk that is intended for a ZFS root pool. This is a long-standing boot limitation. Review the following root pool disk requirements:

- Must contain a disk slice and an SMI (VTOC) label.
- An EFI label is not supported for a root pool disk.
- Must be a single disk or be part of mirrored configuration. A non-redundant configuration nor a RAIDZ configuration is supported for the root pool.
- All subdirectories of the root file system that are part of the OS image, with the exception of /var, must be in the same dataset as the root file system.
- All Solaris OS components must reside in the root pool, with the exception of the swap and dump devices.
Another restriction is that the /var directory or dataset must be a single dataset. For example, you cannot create a descendent /var file system, such as /var/tmp, if you want to also use Oracle Solaris Live Upgrade to migrate or patch a ZFS BE or create a ZFS flash archive of this pool.

In general, you should create a disk slice with the bulk of disk space in slice 0. Attempting to use different slices on a disk and share that disk among different operating systems or with a different ZFS storage pool or storage pool components is not recommended.

**SPARC: How to Create a Disk Slice for a ZFS Root File System**

In general, the root pool disk is installed automatically when the system is installed. If you need to replace a root pool disk or attach a new disk as a mirrored root pool disk, see the steps below.

1. Become superuser.

2. Offline and unconfigure the failed disk, if necessary.
   
   Some hardware requires that you offline and unconfigure a disk before attempting the zpool replace operation to replace a failed disk. For example:
   
   ```
   # zpool offline rpool c2t1d0s0
   # cfgadm -c unconfigure c2::dsk/c2t1d0
   ```

3. Physically connect the new or replacement disk to the system, if necessary.
   
   a. Physically remove the failed disk.
   
   b. Physically insert the replacement disk.
   
   c. Configure the replacement disk, if necessary. For example:
      
      ```
      # cfgadm -c configure c2::dsk/c2t1d0
      ```
      
      On some hardware, you do not have to reconfigure the replacement disk after it is inserted.

4. Confirm that the disk is accessible by reviewing the format output.

   For example, the format command sees 4 disks connected to this system.

   ```
   # format -e
   AVAILABLE DISK SELECTIONS:
   0. c2t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107> /pci@1c,600000/scsi@2/sd@0,0
   1. c2t1d0 <SEAGATE-ST336607LSUN36G-0307-33.92GB> /pci@1c,600000/scsi@2/sd@1,0
   2. c2t2d0 <SEAGATE-ST336607LSUN36G-0507-33.92GB> /pci@1c,600000/scsi@2/sd@2,0
   3. c2t3d0 <SEAGATE-ST336607LSUN36G-0507-33.92GB> /pci@1c,600000/scsi@2/sd@3,0
   ```
5 Select the disk to be used for the ZFS root pool.

6 Confirm that the disk has an SMI label by printing the partition (slice) information.

For example, the partition (slice) output for c2t1d0 shows that this disk has an EFI label because it identifies first and last sectors.

Specify disk (enter its number): 1
selecting c2t1d0
[Disk formatted]

```
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
expand - expand label to use whole disk
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
!<cmd> - execute <cmd>, then return
quit
```

Current partition table (original):
Total disk sectors available: 71116508 + 16384 (reserved sectors)

```
Part Tag Flag  First Sector  Size       Last Sector
 0 usr  wm   256        33.91GB    71116541
 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         0
 8 reserved  wm   71116542  8.00MB     71132925
```

If the disk contains an EFI label, relabel the disk with an SMI label.

For example, the c2t1d0 disk is relabeled with an SMI label, but a default partition table is provided that does not provide an optimal slice configuration.

```
partition> label
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0
Auto configuration via format.dat[no]?
Auto configuration via generic SCSI-2[no]?
partition> p
Current partition table (default):
Total disk cylinders available: 24620 + 2 (reserved cylinders)
```
### Create an optimal slice configuration for a ZFS root pool disk.

Set the free hog partition so that all the unallocated disk space is collected in slice 0. Then, press return through the slice size fields to create one large slice 0.

---

**8 Create an optimal slice configuration for a ZFS root pool disk.**

Set the free hog partition so that all the unallocated disk space is collected in slice 0. Then, press return through the slice size fields to create one large slice 0.

**partition> modify**

Select partitioning base:

- 0. Current partition table (default)
- 1. All Free Hog

Choose base (enter number) [0]? 1

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 90</td>
<td>128.37MB</td>
<td>(91/0/0) 262899</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>91 - 181</td>
<td>128.37MB</td>
<td>(91/0/0) 262899</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 24619</td>
<td>33.92GB</td>
<td>(24620/0/0) 71127180</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wu</td>
<td>182 - 24619</td>
<td>33.67GB</td>
<td>(24438/0/0) 70601382</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
</tbody>
</table>

Do you wish to continue creating a new partition table based on above table[yes]?

Free Hog partition[6]? 0

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]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0 - 90</td>
<td>128.37MB</td>
<td>(91/0/0) 262899</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>91 - 181</td>
<td>128.37MB</td>
<td>(91/0/0) 262899</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 24619</td>
<td>33.92GB</td>
<td>(24620/0/0) 71127180</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wu</td>
<td>182 - 24619</td>
<td>33.67GB</td>
<td>(24438/0/0) 70601382</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wu</td>
<td>0</td>
<td>0</td>
<td>(0/0/0) 0</td>
</tr>
</tbody>
</table>

Okay to make this the current partition table[yes]?
Enter table name (remember quotes): "c2t1d0"

Ready to label disk, continue? yes

**partition> quit**

**format> quit**
9 Let ZFS know that the failed disk is replaced.
   # zpool replace rpool c2t1d0s0
   # zpool online rpool c2t1d0s0

   On some hardware, you do not have to online the replacement disk after it is inserted.
   
   If you are attaching a new disk to create a mirrored root pool or attaching a larger disk to replace a smaller disk, use syntax similar to the following:

   # zpool attach rpool c0t0d0s0 c1t0d0s0

10 If a root pool disk is replaced with a new disk, apply the boot blocks after the new or replacement disk is resilvered.

   For example:
   # zpool status rpool
   installboot -F zfs /usr/platform/`uname -i`/lib/fs/zfs/bootblk /dev/rdsk/c2t1d0s0

11 Verify that you can boot from the new disk.

12 If the system boots from the new disk, detach the old disk.

   This step is only necessary if you attach a new disk to replace a failed disk or a smaller disk.
   # zpool detach rpool c0t0d0s0

13 Set up the system to boot automatically from the new disk, either by using the eeprom command or the setenv command from the SPARC boot PROM.

More Information

After You Have Created a Disk Slice for a ZFS Root File System ...

After you have created a disk slice for the ZFS root file system and you need to restore root pool snapshots to recover your root pool, see "Recovering the ZFS Root Pool or Root Pool Snapshots" in Oracle Solaris ZFS Administration Guide.

SPARC: How to Install Boot Blocks for a ZFS Root File System

1 Become superuser.

2 Install a boot block for a ZFS root file system.

   # installboot -F zfs /usr/platform/`uname -i`/lib/fs/zfs/bootblk
   /dev/rdsk/cwtxdyd0s0

   For more information, see installboot(1M).
3 Verify that the boot blocks are installed by rebooting the system to run level 3.
   # init 6

Example 10–4 SPARC: Installing Boot Blocks for a ZFS Root File System

If you physically replace the disk that is intended for the root pool and the Oracle Solaris OS is then reinstalled, or you attach a new disk for the root pool, the boot blocks are installed automatically. If you replace a disk that is intended for the root pool by using the \texttt{zpool replace} command, then you must install the boot blocks manually so that the system can boot from the replacement disk.

The following example shows how to install boot blocks for a ZFS root file system.

\begin{verbatim}
# installboot -F zfs /usr/platform/'uname -i'/lib/fs/zfs/bootblk /dev/rdsk/c0t1d0s0
\end{verbatim}

\textbf{SPARC: How to Set Up a Disk for a ZFS File System}

If you are setting up a disk to be used with a non-root ZFS file system, the disk is relabeled automatically when the pool is created or when the disk is added to the pool. If a pool is created with whole disks or when a whole disk is added to a ZFS storage pool, an EFI label is applied. For more information about EFI disk labels, see "EFI (GPT) Disk Label" on page 156.

Generally, most modern bus types support hot-plugging. This means you can insert a disk in an empty slot and the system recognizes it. For more information about hot-plugging devices, see Chapter 4, "Dynamically Configuring Devices (Tasks)."

1 Become superuser.

2 Connect the disk to the system and check the disk’s physical connections.
   Refer to the disk’s hardware installation guide for details.

3 Offline and unconfigure the failed disk, if necessary.
   Some hardware requires that you offline and unconfigure a disk before attempting the \texttt{zpool replace} operation to replace a failed disk. For example:
   \begin{verbatim}
   # zpool offline tank clt1d0
   # cfgadm -c unconfigure cl::dsk/clt1d0
   <Physically remove failed disk clt1d0>
   <Physically insert replacement disk clt1d0>
   # cfgadm -c configure cl::dsk/clt1d0
   \end{verbatim}
   On some hardware, you do not to reconfigure the replacement disk after it is inserted.
4 **Confirm that the new disk is recognized.**
Review the output of the `format` utility to see if the disk is listed under AVAILABLE DISK SELECTIONS. Then, quit the `format` utility.

```
# format
```

5 **Let ZFS know that the failed disk is replaced, if necessary.**

```
# zpool replace tank c1t1d0
# zpool online tank c1t1d0
```

Confirm that the new disk is resilvering.

```
# zpool status tank
```

6 **Attach a new disk to an existing ZFS storage pool, if necessary.**

For example:

```
# zpool attach tank mirror c1t0d0 c2t0d0
```

Confirm that the new disk is resilvering.

```
# zpool status tank
```

For more information, see Chapter 3, "Managing Oracle Solaris ZFS Storage Pools," in *Oracle Solaris ZFS Administration Guide*. 
This chapter describes how to set up disks on an x86 based system.

For information on the procedures associated with setting up disks on an x86 based system, see the following sections:

- “x86: Setting Up Disks for UFS File Systems (Task Map)” on page 217
- “x86: Setting Up Disks for ZFS File Systems (Task Map)” on page 222
- “Creating and Changing Solaris fdisk Partitions” on page 230

For overview information about disk management, see Chapter 7, “Managing Disks (Overview).” For step-by-step instructions on setting up disks on a SPARC based system, see Chapter 10, “SPARC: Setting Up Disks (Tasks).”

### x86: Setting Up Disks for UFS File Systems (Task Map)

The following task map identifies the procedures for setting up a disk for a UFS root file system or a disk for a UFS file system on an x86 based system.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Connect the disk.</td>
<td><em>Disk for a UFS root file system</em></td>
<td>“x86: How to Set Up a Disk for a UFS Root File System” on page 218</td>
</tr>
<tr>
<td></td>
<td>Connect the new disk and boot from a local or remote Oracle Solaris DVD.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><em>Disk for a UFS file system</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Connect the disk.</td>
<td>“x86: How to Connect a Disk for a UFS File System” on page 219</td>
</tr>
<tr>
<td>2. Create or change the fdisk partition</td>
<td>The disk for an x86 based system must contain an fdisk partition.</td>
<td></td>
</tr>
</tbody>
</table>
Task Description For Instructions

3. 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 231 and “x86: How to Create Disk Slices for UFS File Systems” on page 219

4. Create UFS file systems. Create UFS file systems on the disk slices with the newfs command. You must re-create the root (/) or /usr file system (or both). “x86: How to Create a UFS File System” on page 221

5. Restore UFS file systems. Restore the UFS root (/) or /usr file system (or both) on the system disk. If necessary, restore file systems on other disks “Restoring UFS Files and File System Backups (Task Map)” on page 369

6. Install boot blocks for a UFS root file system. Install the boot blocks on the root (/) file system so that the system can boot. “x86: How to Install Boot Blocks for a UFS Root File System” on page 222

x86: Setting Up Disks for UFS File Systems

A disk can contain the UFS root (/) and /usr file systems, user or data file systems, or a combination of file systems. If the disk contains the UFS root (/) or /usr file systems, you have two ways to recover:

- You can reinstall the entire Oracle Solaris OS.
- Or, you can replace the disk and restore your Oracle Solaris OS file systems from a backup medium.

A disk that contains user or data file systems can be replaced and the file systems restored, if necessary, without having to reinstall or reboot the system.

▼ x86: How to Set Up a Disk for a UFS Root File System

You can use this procedure to replace a disk that contained the UFS root or /usr file system. Refer to your hardware installation guide for information on replacing a disk.

1. Disconnect the damaged system disk from the system, if necessary.

2. Connect the replacement disk to the system, and check the disk’s physical connections.

3. Follow the instructions in the following table, depending on whether you are booting from a local Oracle Solaris DVD or a remote Oracle Solaris DVD from the network.
### Boot Type  
#### Action

**From an Oracle Solaris DVD in a local drive**
1. Make sure the Oracle Solaris DVD is in the drive.
2. Select the option to boot from the media.

**From the network**
3. Select the option to boot from the network.

---

**More Information**

After You Set Up a Disk for a UFS Root File System...

Create an `fdisk` partition, if necessary. Go to "x86: How to Create a Solaris `fdisk` Partition" on page 231.

### x86: How to Connect a Disk for a UFS File System

If you are adding a disk with an EFI disk label, see "EFI (GPT) Disk Label" on page 156 for more information.

Generally, most modern bus types support hot-plugging. This means you can insert a disk in an empty slot and the system recognizes it. For more information about hot-plugging devices, see Chapter 4, "Dynamically Configuring Devices (Tasks)."

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

2. **(Optional) If the disk is unsupported by the Oracle Solaris software, add the device driver for the disk by following the instructions included with the hardware.**

3. **Connect the disk to the system and check the disk's physical connections.**
   Refer to the disk's hardware installation guide for details.

4. **Use the format utility to confirm that the new disk is recognized.**

---

**More Information**

After You Connect a Disk for a UFS File System...

After the disk is connected, create an `fdisk` partition. Go to "x86: How to Create a Solaris `fdisk` Partition" on page 231.

### x86: How to Create Disk Slices for UFS File Systems

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

2. **Confirm that the disk has a valid Solaris `fdisk` partition.**
   For information about creating or changing a Solaris `fdisk` partition, see “Creating and Changing Solaris `fdisk` Partitions” on page 230.
3 Invoke the `format` utility.
   
   ```bash
   # format
   ```
   A numbered list of disks is displayed.

4 Type the number of the disk that you want to repartition.
   Specify disk (enter its number): `disk-number`
   where `disk-number` is the number of the disk that you want to repartition.

5 Select the partition menu.
   ```
   format> partition
   ```

6 Display the current partition (slice) table.
   ```
   partition> print
   ```

7 Start the modification process.
   ```
   partition> modify
   ```

8 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 165.

9 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

10 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 the following:
    - root (slice 0) and swap (slice 1) and/or
    - /usr (slice 6)
    After you identify the slices, the new partition table is displayed.

11 Make the displayed partition table the current partition table by answering yes when prompted.
    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 7.

12 Name the partition table.
   Enter table name (remember quotes): "partition-name"
   where `partition-name` is the name for the new partition table.
Label the disk with the new partition table after you have finished allocating slices on the new disk.

Ready to label disk, continue? yes

Quit the partition menu.

partition> quit

Verify the new disk label.

format> verify

Exit the format utility.

format> quit

More Information

After You Create Disk Slices and Label a Disk ...

After you create disk slices and label the disk, you can create file systems on the disk. Go to “x86: How to Create a UFS File System” on page 221.

▼ x86: How to Create a UFS File System

1 Become superuser or assume an equivalent role.

2 Create a file system for each slice.

   # newfs /dev/rdsk/cwtxdydz

   where /dev/rdsk/cwtxdydz is the raw device for the file system to be created.

   For more information about the newfs command, see Chapter 15, “Creating and Mounting File Systems (Tasks),” or newfs(1M).

3 Verify the new file system by mounting it.

   # mount /dev/dsk/cwtxdydz /mnt
   # ls /mnt
   lost+found

More Information

After You Create a UFS File System ...

- Disk for a UFS root file system – You need to restore the root (/) and /usr file systems on the disk.
  - Go to “Restoring UFS Files and File System Backups (Task Map)” on page 369.
  - After the root (/) and /usr file systems are restored, install the boot block. Go to “x86: How to Install Boot Blocks for a UFS Root File System” on page 222.

- Disk for a UFS file system – You might need to restore file systems on the new disk. Go to “Restoring UFS Files and File System Backups (Task Map)” on page 369. If you are not restoring file systems on the new disk, you are finished adding a secondary disk.
x86: Setting Up Disks for ZFS File Systems (Task Map)

- For information on making the file systems available to users, see “Mounting and Unmounting Oracle Solaris File Systems” on page 286.

x86: How to Install Boot Blocks for a UFS Root File System

1. Become superuser or assume an equivalent role.

2. Install the boot blocks for a UFS root file system.
   ```bash
   # /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/cwtxdysz
   ``
   For more information, see `installgrub(1M)`.

3. Verify that the boot blocks are installed by rebooting the system to run level 3.
   ```bash
   # init 6
   ```

Example 11–1 x86: Installing Boot Blocks for a UFS Root File System

The following example shows how to install the boot blocks for a UFS root file system on an x86 system.

```bash
# /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1d0s0
stage1 written to partition 0 sector 0 (abs 2016)
stage2 written to to partition 0, 227 sectors starting 50 (abs 2066)
```

x86: Setting Up Disks for ZFS File Systems (Task Map)

The following task map identifies the procedures for setting up a ZFS root pool disk for a ZFS root file system on an x86 based system.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
</table>
| 1. Set up the disk for a ZFS root file system. | *Disk for a ZFS Root File System*  
Connect the new disk or replace the existing root pool disk and boot from a local or remote Oracle Solaris DVD. | "x86: How to Set Up a Disk for a ZFS Root File System” on page 224 |
| 2. Create or change an fdisk partition, if necessary. | *The disk must contain a valid Solaris fdisk partition.* | "Creating and Changing Solaris fdisk Partitions” on page 230 |
| 3. Create a disk slice for the ZFS root file system. | *Create a disk slice for a disk that is intended for a ZFS root pool. This is a long-standing boot limitation.* | "x86: How to Create a Solaris fdisk Partition” on page 231 and "x86: How to Create a Disk Slice for a ZFS Root File System” on page 225 |
### Task Description

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Install the boot blocks for a ZFS root file system.</td>
<td>If you replace a disk that is intended for the root pool by using the <code>zpool replace</code> command, then you must install the boot blocks manually so that the system can boot from the replacement disk.</td>
<td>“x86: How to Install Boot Blocks for a ZFS Root File System” on page 228</td>
</tr>
</tbody>
</table>

| 5. Set up a disk for a ZFS file system. | Disk for a ZFS File System
Connect the disk. | “x86: How to Set Up a Disk for a ZFS File System” on page 229 |

---

### x86: Setting Up Disks for ZFS File Systems

Although the procedures that describe how to set up a disk and create an fdisk partition can be used with a ZFS file systems, a ZFS file system is not directly mapped to a disk or a disk slice. You must create a ZFS storage pool before creating a ZFS file system. For more information, see *Oracle Solaris ZFS Administration Guide*.

The root pool contains the root file system that is used to boot the Oracle Solaris OS. If a root pool disk becomes damaged and the root pool is not mirrored, the system might not boot. If a root pool disk becomes damaged, you have two ways to recover:

- You can reinstall the entire Oracle Solaris OS.
- Or, you can replace the root pool disk and restore your file systems from snapshots or from a backup medium. You can reduce system down time due to hardware failures by creating a redundant root pool. The only supported redundant root pool configuration is a mirrored root pool.

A disk that is used in a non-root pool usually contains space for user or data files. You can attach or add another disk to a root pool or a non-root pool for more disk space. Or, you can replace a damaged disk in a pool in the following ways:

- A disk can be replaced in a non-redundant pool if all the devices are currently ONLINE.
- A disk can be replaced in a redundant pool if enough redundancy exists among the other devices.
- In a mirrored root pool, you can replace a disk or attach a disk and then detach the failed disk or a smaller disk to increase a pool’s size.

In general, setting up a disk to the system depends on the hardware so review your hardware documentation when adding or replacing a disk on your system. If you need to add a disk to an existing controller, then it might just be a matter of inserting the disk in an empty slot, if the system supports hot-plugging. If you need to configure a new controller, see “Dynamic Reconfiguration and Hot-Plugging” on page 67.
**x86: How to Set Up a Disk for a ZFS Root File System**

Refer to your hardware installation guide for information on replacing a disk.

1. **Disconnect the damaged disk from the system, if necessary.**
2. **Connect the replacement disk to the system, and check the disk’s physical connections.**
3. **Follow the instructions in the following table, depending on whether you are booting from a local Oracle Solaris DVD or a remote Oracle Solaris DVD from the network.**

<table>
<thead>
<tr>
<th>Boot Type</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>From an Oracle Solaris DVD in a local drive</td>
<td>1. Make sure the Oracle Solaris DVD is in the drive.</td>
</tr>
<tr>
<td></td>
<td>2. Select the option to boot from the media.</td>
</tr>
<tr>
<td>From the network</td>
<td>3. Select the option to boot from the network.</td>
</tr>
</tbody>
</table>

More Information

After You Set Up a Disk for a ZFS Root File System...

After the disk is connected or replaced, create an `fdisk` partition. Go to “x86: How to Create a Solaris `fdisk` Partition” on page 231.

**x86: Creating a Disk Slice for a ZFS Root File System**

You must create a disk slice for a disk that is intended for a ZFS root pool. This is a long-standing boot limitation. Review the following root pool disk requirements:

- Must contain a disk slice and an SMI (VTOC) label.
- An EFI label is not supported for a root pool disk.
- A root pool disk on an x86 system must contain an `fdisk` partition.
- Must be a single disk or be part of mirrored configuration. A non-redundant configuration nor a RAIDZ configuration is supported for the root pool.
- All subdirectories of the root file system that are part of the OS image, with the exception of `/var`, must be in the same dataset as the root file system.
- All Solaris OS components must reside in the root pool, with the exception of the swap and dump devices.
- Another restriction is that the `/var` directory or dataset must be a single dataset. For example, you cannot create a descendent `/var` file system, such as `/var/tmp`, if you want to also use Oracle Solaris Live Upgrade to migrate or patch a ZFS BE or create a ZFS flash archive of this pool.

On an x86 based system, you must first create an `fdisk` partition. Then, create a disk slice with the bulk of disk space in slice 0.
Attempting to use different slices on a disk and share that disk among different operating systems or with a different ZFS storage pool or storage pool components is not recommended.

▼ **x86: How to Create a Disk Slice for a ZFS Root File System**

In general, the root pool disk is installed automatically when the system is installed. If you need to replace a root pool disk or attach a new disk as a mirrored root pool disk, see the steps below.

For a full description of fdisk partitions, see “x86: Guidelines for Creating an fdisk Partition” on page 230.

1. **Become superuser.**
2. **Offline and unconfigure the failed disk, if necessary.**
   Some hardware requires that you offline and unconfigure a disk before attempting the zpool replace operation to replace a failed disk. For example:
   ```
   # zpool offline rpool c2t1d0
   # cfgadm -c unconfigure c2::dsk/c2t1d0
   ```
3. **Physically connect the new or replacement disk to the system, if necessary.**
   a. **Physically remove the failed disk.**
   b. **Physically insert the replacement disk.**
   c. **Configure the replacement disk, if necessary.** For example:
      ```
      # cfgadm -c configure c2::dsk/c2t1d0
      ```
      On some hardware, you do not have to reconfigure the replacement disk after it is inserted.
4. **Confirm that the disk is accessible by reviewing the format output.**
   For example, the format command sees 4 disks connected to this system.
   ```
   # format -e
   AVAILABLE DISK SELECTIONS:
   1. c8t0d0 <Sun-STK RAID INT-V1.0 cyl 17830 alt 2 hd 255 sec 63> /pci@0,0/pci10de,375f/pci100e,286@0/disk@0,0
   2. c8t1d0 <Sun-STK RAID INT-V1.0-136.61GB> /pci@0,0/pci10de,375f/pci100e,286@0/disk@1,0
   3. c8t2d0 <Sun-STK RAID INT-V1.0-136.61GB> /pci@0,0/pci10de,375f/pci100e,286@0/disk@2,0
   4. c8t3d0 <Sun-STK RAID INT-V1.0-136.61GB> /pci@0,0/pci10de,375f/pci100e,286@0/disk@3,0
   ```
5. **Select the disk to be used for the ZFS root pool.**
   Specify disk (enter its number): 1
   selecting c8t0d0
   [disk formatted]

Chapter 11 • x86: Setting Up Disks (Tasks) 225
6 **Review the status of the fdisk partition.**

- If the disk has no fdisk partition, you will see a message similar to the following:

  ```
  format> fdisk
  No Solaris fdisk partition found.
  ```

  If so, go to step 4 to create an fdisk partition.

- If the disk has an EFI fdisk or some other partition type, go to step 5 to create a Solaris fdisk partition.

- If the disk has a Solaris fdisk partition, go to step 6 to create a disk slice for the root pool.

7 **If necessary, create a Solaris fdisk partition by selecting the fdisk option.**

   ```
   format> fdisk
   No fdisk table exists. The default partition for the disk is:
   a 100% "SOLARIS System" partition
   ```

   Type 'y' to accept the default partition, otherwise type "n" to edit the partition table. y

Then, go to step 6 to create a disk slice for the root pool.

8 **If the disk has an EFI fdisk partition, then you will need to create a Solaris fdisk partition.**

   If you print the disk's partition table with the format utility, and you see the partition table refers to the first sector and the size, then this is an EFI partition. You will need to create a Solaris fdisk partition as follows:

   - Select fdisk from the format options.

     ```
     # format -e c8t0d0
     selecting c8t0d0
     [disk formatted]
     format> fdisk
     ```

   - Delete the existing EFI partition by selecting option 3, Delete a partition.

     Enter Selection: 3
     Specify the partition number to delete (or enter 0 to exit): 1
     Are you sure you want to delete partition 1? This will make all files and programs in this partition inaccessible (type "y" or "n"). y

     Partition 1 has been deleted.

   - Create a new Solaris partition by selecting option 1, Create a partition.

     Enter Selection: 1
     Select the partition type to create: 1
     Specify the percentage of disk to use for this partition (or type "c" to specify the size in cylinders). 100
Should this become the active partition? If yes, it will be activated each time the computer is reset or turned on. Please type "y" or "n". y
Partition 1 is now the active partition.

- Update the disk configuration and exit.

Enter Selection: 6
format>

- Display the SMI partition table. If the default partition table is applied, then slice 0 might be 0 in size or it might be too small. See the next step.

format> partition
partiton> print

9  Confirm that the disk has an SMI label by printing the partition (slice) information and review the slice 0 size information.

Set the free hog partition so that all the unallocated disk space is collected in slice 0. Then, press return through the slice size fields to create one large slice 0.

partition> modify
Select partitioning base:
  0. Current partition table (default)
  1. All Free Hog

Choose base (enter number) [0]? 1

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 17829</td>
<td>136.58GB (17830/0/0)</td>
<td>286438950</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>boot</td>
<td>wu</td>
<td>0 - 0</td>
<td>7.84MB (1/0/0)</td>
<td>16065</td>
</tr>
<tr>
<td>9</td>
<td>alterates</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
</tbody>
</table>

Do you wish to continue creating a new partition table based on above table[yes]?
Free Hog partition[6]? 0
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]:
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:

<table>
<thead>
<tr>
<th>Part</th>
<th>Tag</th>
<th>Flag</th>
<th>Cylinders</th>
<th>Size</th>
<th>Blocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>root</td>
<td>wm</td>
<td>1 - 17829</td>
<td>136.58GB (17829/0/0)</td>
<td>286422885</td>
</tr>
<tr>
<td>1</td>
<td>swap</td>
<td>wu</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>backup</td>
<td>wu</td>
<td>0 - 17829</td>
<td>136.58GB (17830/0/0)</td>
<td>286438950</td>
</tr>
<tr>
<td>3</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>usr</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>unassigned</td>
<td>wm</td>
<td>0</td>
<td>0 (0/0/0)</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>boot</td>
<td>wu</td>
<td>0 - 0</td>
<td>7.84MB (1/0/0)</td>
<td>16065</td>
</tr>
</tbody>
</table>
Do you wish to continue creating a new partition table based on above table? [yes]?
Yes
Enter table name (remember quotes): "c8t0d0"

Ready to label disk, continue? yes

10 Let ZFS know that the failed disk is replaced.
   # zpool replace rpool c2t1d0s0
   # zpool online rpool c2t1d0s0

On some hardware, you do not have to online the replacement disk after it is inserted.

If you are attaching a new disk to create a mirrored root pool or attaching a larger disk to replace
a smaller disk, use syntax similar to the following:

   # zpool attach rpool c0t0d0s0 c1t0d0s0

11 If a root pool disk is replaced with a new disk, apply the boot blocks.
   For example:
   # installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c8t0d0s0

12 Verify that you can boot from the new disk.

13 If the system boots from the new disk, detach the old disk.
   This step is only necessary if you attach a new disk to replace a failed disk or a smaller disk.
   # zpool detach rpool c0t0d0s0

14 Set up the system to boot automatically from the new disk by reconfiguring the system's BIOS.

More Information

After You Have Created a Disk Slice for the ZFS Root File System ...

After you have created a disk slice for the ZFS root file system and you need to restore root pool
snapshots to recover your root pool, see “Recovering the ZFS Root Pool or Root Pool Snapshots” in Oracle Solaris ZFS Administration Guide.

▼ x86: How to Install Boot Blocks for a ZFS Root File System

1 Become superuser.

2 Install the boot blocks on the system disk.
   # installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c0t0d0s0
   For more information, see installgrub(1M).

3 Verify that the boot blocks are installed by rebooting the system to run level 3.
   # init 6
Example 11–2  x86: Installing Boot Blocks for a ZFS Root File System

If you physically replace the disk that is intended for the root pool and the Oracle Solaris OS is then reinstalled, or you attach a new disk for the root pool, the boot blocks are installed automatically. If you replace a disk that is intended for the root pool by using the `zpool replace` command, then you must install the boot blocks manually so that the system can boot from the replacement disk.

The following example shows how to install the boot blocks for a ZFS root file system.

```
# installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c1d0s0
stage2 written to partition 0, 277 sectors starting at 50 (abs 16115)
stagel written to partition 0 sector 0 (abs 16065)
```

▼  x86: How to SetUp a Disk for a ZFS File System

If you are setting up a disk to be used with a non-root ZFS file system, the disk is relabeled automatically when the pool is created or when the disk is added to the pool. If a pool is created with whole disks or when a whole disk is added to a ZFS storage pool, an EFI label is applied. For more information about EFI disk labels, see “EFI (GPT) Disk Label” on page 156.

Generally, most modern bus types support hot-plugging. This means you can insert a disk in an empty slot and the system recognizes it. For more information about hot-plugging devices, see Chapter 4, “Dynamically Configuring Devices (Tasks).”

1 Become superuser.

2 Connect the disk to the system and check the disk's physical connections.
Refer to the disk's hardware installation guide for details.

3 Offline and unconfigure the failed disk, if necessary.
Some hardware requires that you offline and unconfigure a disk before attempting the `zpool replace` operation to replace a failed disk. For example:

```
# zpool offline tank c1t1d0
# cfgadm -c unconfigure cl::dsk/c1t1d0
  <Physically remove failed disk c1t1d0>
  <Physically insert replacement disk c1t1d0>
# cfgadm -c configure cl::dsk/c1t1d0
```

On some hardware, you do not to reconfigure the replacement disk after it is inserted.

4 Confirm that the new disk is recognized.
Review the output of the `format` utility to see if the disk is listed under `AVAILABLE DISK SELECTIONS`. Then, quit the `format` utility.

```
# format
```
Let ZFS know that the failed disk is replaced, if necessary.

```
# zpool replace tank c1t1d0
# zpool online tank c1t1d0
```

Confirm that the new disk is resilvering.

```
# zpool status tank
```

Attach a new disk to an existing ZFS storage pool, if necessary.

For example:

```
# zpool attach tank mirror c1t0d0 c2t0d0
```

Confirm that the new disk is resilvering.

```
# zpool status tank
```

For more information, see Chapter 3, "Managing Oracle Solaris ZFS Storage Pools," in Oracle Solaris ZFS Administration Guide.

Creating and Changing Solaris fdisk Partitions

Review the following sections for guidelines and examples of creating or changing Solaris fdisk partitions.

**x86: Guidelines for Creating an fdisk Partition**

Follow these guidelines when you set up one or more fdisk partitions.

- 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 sufficient space is available) to create a new partition.
Solaris slices are also called partitions. Certain interfaces might refer to a slice as a partition.

`fdisk` partitions are supported only on x86 based systems. To avoid confusion, Oracle Solaris documentation tries to distinguish between `fdisk` partitions and the entities within the Solaris `fdisk` partition. These entities might be called slices or partitions.

**x86: How to Create a Solaris `fdisk` Partition**

**Before You Begin**

If you need information about `fdisk` partitions, see "x86: Guidelines for Creating an `fdisk` Partition" on page 230.

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

2. **Invoke the `format` utility.**
   ```
   # format
   
   A numbered list of disks is displayed.
   ```
   For more information, see `format(1M)`.

3. **Type the number of the disk on which to create a Solaris `fdisk` partition.**
   Specify disk (enter its number): `disk-number`
   where `disk-number` is the number of the disk on which you want to create a Solaris `fdisk` partition.

4. **Select the `fdisk` menu.**
   ```
   format> fdisk
   
   The `fdisk` menu that is displayed depends upon whether the disk has existing `fdisk` partitions.
   ```

5. **Create and activate a Solaris `fdisk` partition that spans the entire disk by specifying y at the prompt. Then, go to step 13.**
   No `fdisk` table exists. The default partition for the disk is:
   a 100% "SOLARIS System" partition
   ```
   Type "y" to accept the default partition, otherwise type "n" to edit the partition table.
   y
   ```

6. **Specify n at the prompt if you do not want the Solaris `fdisk` partition to span the entire disk.**
   Type "y" to accept the default partition, otherwise type "n" to edit the partition table.
   ```
   n
   ```
Total disk size is 17848 cylinders
Cylinder size is 16065 (512 byte) blocks

<table>
<thead>
<tr>
<th>Cylinder</th>
<th>Status</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
<th>%</th>
</tr>
</thead>
</table>

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Edit/View extended partitions
6. Exit (update disk configuration and exit)
7. Cancel (exit without updating disk configuration)
Enter Selection:

Select option 1, Create a partition, to create an fdisk partition.
Enter Selection: 1

8 Create a Solaris fdisk partition by selecting 1(=Solaris2).

Select the partition type to create:
1=SOLARIS2 2=UNIX 3=PCIXOS 4=Other
5=DOS12 6=DOS16 7=DOSEX 8=DOSBIG
9=DOS16LBA A=x86 Boot B=Diagnostic C=FAT32
D=FAT32LBA E=DOSEX TLBA F=EFI 0=Exit? 1

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.

Specify the percentage of disk to use for this partition (or type "c" to specify the size in cylinders). nn

Activate the Solaris fdisk partition by typing y at the prompt.

Should this become the active partition? If yes, it will be activated each time the computer is reset or turned on.

Please type "y" or "n". y

The Enter Selection prompt is displayed after the fdisk partition is activated.

Select option 1, Create a partition, to create another fdisk partition.

See steps 8–10 for instructions on creating an fdisk partition.

Update the disk configuration, and exit the fdisk menu from the selection menu.
Selection: 6

Relabel the disk by using the label command.

format> label
Ready to label disk, continue? yes
format>
Example 11–3  x86: Creating a Solaris fdisk Partition That Spans the Entire Drive

The following example uses the format 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
No fdisk table exists. The default partitioning for your disk is:

   a 100% "SOLARIS System" partition.

Type 'y' to accept the default partition, otherwise type "n" to edit the partition table. y

format> label
Ready to label disk, continue? yes
format> quit
```

More Information  After You Create a Solaris fdisk Partition ...

After you create a Solaris fdisk partition on the disk, you can use for a root pool disk or non-root pool disk. For more information, see “SPARC: Setting Up Disks for UFS File Systems” on page 200.

Changing the fdisk Partition Identifier

The Solaris fdisk partition identifier on x86 systems has been changed from 130 (0x82) to 191 (0xbf). All Oracle Solaris commands, utilities, and drivers have been updated to work with either fdisk identifier. There is no change in fdisk functionality.

How to Change the Solaris fdisk Identifier

A new fdisk menu option enables you to switch back and forth between the new and old identifier. The fdisk identifier can be changed even when the file system that is contained in the partition is mounted.
Two type values in the fdisk menu reflect the old and new identifiers as follows:

- Solaris identifies 0x82
- Solaris2 identifies 0xbf

1 Become superuser.

2 Display the current fdisk identifier.

For example:

```
Total disk size is 39890 cylinders
Cylinder size is 4032 (512 byte) blocks

Partition Status Type Start End Length %
========= ====== ============ ===== === ====== ===
1 Active x86 Boot 1 6 6 0
2 Solaris2 7 39889 39883 100
```

3 Select option 4 from the fdisk menu to change the fdisk partition identifier back to 0x82.

4 Select option 5 to update your disk configuration and exit.

5 If necessary, select option 4 from the fdisk menu to change the fdisk partition identifier back to 0xbf.

For example:

```
Total disk size is 39890 cylinders
Cylinder size is 4032 (512 byte) blocks

Partition Status Type Start End Length %
========= ====== ============ ===== === ====== ===
1 Active x86 Boot 1 6 6 0
2 Solaris 7 39889 39883 100
```

6 Select option 5 to update your disk configuration and exit.
This chapter describes how to configure iSCSI targets in the Oracle Solaris release. For information about the procedures associated with configuring iSCSI targets, see "Configuring Solaris iSCSI Target Devices (Tasks)" on page 237.

For information about configuring iSCSI initiators, see Chapter 4, "Configuring Oracle Solaris iSCSI Initiators," in Oracle Solaris SAN Configuration and Multipathing Guide.

For troubleshooting Solaris iSCSI configuration problems, see “Troubleshooting iSCSI Configuration Problems” on page 256.

Oracle Solaris iSCSI Technology (Overview)

iSCSI is an acronym for Internet SCSI (Small Computer System Interface), an Internet Protocol (IP)-based storage networking standard for linking data storage subsystems. This networking standard was developed by the Internet Engineering Task Force (IETF). For more information about the iSCSI technology, see RFC 3720:

http://www.ietf.org/rfc/rfc3720.txt

By carrying SCSI commands over IP networks, the iSCSI protocol enables you to access block devices from across the network as if they were connected to the local system.

If you want to use storage devices in your existing TCP/IP network, the following solutions are available:

- iSCSI block devices or tape – Translates SCSI commands and data from the block level into IP packets. Using iSCSI in your network is advantageous when you need to have block-level access between one system and the target device, such as a tape device or a database. Access to a block-level device is not locked so that you could have multiple users or systems accessing a block-level device such as an iSCSI target device.
NFS – Transfers file data over IP. The advantage of using NFS in your network is that you can share file data across many systems. Access to file data is locked appropriately when many users are accessing data that is available in an NFS environment.

Here are the benefits of using Solaris iSCSI targets and initiators:

- The iSCSI protocol runs across existing Ethernet networks.
  - You can use any supported network interface card (NIC), Ethernet hub, or Ethernet switch.
  - One IP port can handle multiple iSCSI target devices.
  - You can use existing infrastructure and management tools for IP networks.
- You might have existing Fibre-Channel devices that can be connected to clients without the cost of Fibre-Channel HBAs. In addition, systems with dedicated arrays can now export replicated storage with ZFS or UFS file systems.
- There is no upper limit on the maximum number of configured iSCSI target devices.
- The protocol can be used to connect to Fibre Channel or iSCSI Storage Area Network (SAN) environments with the appropriate hardware.

Here are the current limitations or restrictions of using the Solaris iSCSI initiator software:

- Support for iSCSI devices that use SLP is not currently available.
- Boot support for iSCSI devices is not currently available.
- iSCSI targets cannot be configured as dump devices.
- iSCSI supports multiple connections per session, but the current Solaris implementation only supports a single connection per session.
  
  For more information, see RFC 3720.
- Transferring large amounts of data over your existing network can have an impact on performance.

**Identifying Solaris iSCSI Software and Hardware Requirements**

- Solaris iSCSI software and devices
- Solaris 10 release (at least the 1/06 release) for Solaris iSCSI initiator software
- Solaris 10 release (at least the 8/07 release) for Solaris iSCSI target software
- The following Solaris 10 software packages:
  - SUNWiscsir – Sun iSCSI Device Driver (root)
  - SUNWiscsiu – Sun iSCSI (usr)
  - SUNWiscsitgtr – Sun iSCSI Target (root)
Configuring Solaris iSCSI Target Devices (Tasks)

This is a general list of tasks associated with configuring Solaris iSCSI target devices. Some of the tasks are optional depending on your network configuration needs. Some of the links below will take you to separate documents that describe network configuration and initiator configuration.

- “Identifying Solaris iSCSI Software and Hardware Requirements” on page 236
- Oracle Solaris Administration: Network Interfaces and Network Virtualization
- “Configuring and Managing Solaris iSCSI Target Devices” on page 238
- “How to Configure an iSCSI Initiator and Target Discovery” in Oracle Solaris SAN Configuration and Multipathing Guide
- “How to Access iSCSI Disks” on page 241
- “Configuring Authentication in Your iSCSI-Based Storage Network” on page 243
- “Setting Up Solaris iSCSI Multipathed Devices” on page 247
- “Monitoring Your iSCSI Configuration” on page 251

Oracle Solaris iSCSI Terminology

Review the following terminology before configuring iSCSI targets and initiators.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiator</td>
<td>The driver that initiates SCSI requests to the iSCSI target.</td>
</tr>
<tr>
<td>Target device</td>
<td>The iSCSI storage component.</td>
</tr>
<tr>
<td>Discovery</td>
<td>The process that presents the initiator with a list of available targets.</td>
</tr>
<tr>
<td>Discovery method</td>
<td>The way in which the iSCSI targets can be found. Three methods are currently available:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Configuring Dynamic or Static Target Discovery

Determine whether you want to configure one of the dynamic device discovery methods or use static iSCSI initiator targets to perform device discovery.

- **Dynamic device discovery** – If an iSCSI node exposes many targets, such as an iSCSI to Fibre-Channel bridge, you can supply the iSCSI node IP address/port combination and allow the iSCSI initiator to use the SendTargets features to perform device discovery.

  Two dynamic device discovery methods are available:

  - SendTargets - If an iSCSI node exposes a large number of targets, such as an iSCSI to Fibre-Channel bridge, you can supply the iSCSI node IP address/port combination and allow the iSCSI initiator to use the SendTargets features to perform the device discovery.
  
  - iSNS - iSNS (Internet Storage Name Service) allows the iSCSI initiator to discover the targets to which it has access using as little configuration information as possible. It also provides state change notification to notify the iSCSI initiator when changes in the operational state of storage nodes occur. To use the iSNS discovery method, you can supply the iSNS server address/port combination and allow the iSCSI initiator to query the iSNS servers that you specified to perform the device discovery. The default port for the iSNS server is 3205. For more information about iSNS, see RFC 4171:
    
    http://www.ietf.org/rfc/rfc4171.txt

    The iSNS discovery service provides an administrative model to discover all targets in a network.

  - **Static device discovery** – If an iSCSI node has few targets or if you want to restrict the targets that the initiator attempts to access, you can statically configure the `target-name` by using the following static target address naming convention:

    `target,target-address[:port-number]`

    You can determine the static target address from the array’s management tool.

---

**Note** – Do not configure an iSCSI target to be discovered by both static and dynamic device discovery methods. The consequence of using redundant discovery methods might be slow performance when the initiator is communicating with the iSCSI target device.

---

**Configuring and Managing Solaris iSCSI Target Devices**

You can use the `iscsiadm` command to set up and manage your Solaris iSCSI target devices, which can be disk or tape devices. For the device that you select as your iSCSI target, you must provide an equivalently sized ZFS or UFS file system as the backing store for the iSCSI daemon.
For information about setting up a Solaris iSCSI target device with ZFS, see “Using a ZFS Volume as a Solaris iSCSI Target” in Oracle Solaris ZFS Administration Guide.

After the target device is set up, use the iscsiadm command to identify your iSCSI targets, which will discover and use the iSCSI target device.

For more information, see iscsiadm(1M) and iscsiadm(1M).

The basic process is as follows:

- Identify the backing store directory – For each target and logical unit that is created, the iSCSI daemon needs to store some information. By default, the backing store for this device is also located in the base directory. So, if the host system has a large ZFS pool to use, it might be easiest to allow the daemon to store everything in that location. If the backing store needs to be spread out, it’s possible to specify the backing store location during the creation of each logical unit.

- Create the iSCSI target – By default, the CLI assumes that the requested device type is an LBA of logical unit 0. If a pass through mode is desired for character devices, the -raw option must be used. After the creation of the first LUN, other LUNs might be created for the same iSCSI target by specifying -lun number.

The daemon starts a background task that initializes the LUN to zeros. If, during that initialization, the underlying file system becomes 100 percent full, the daemon removes the target. During this initialization, the LUN is marked as being offline and cannot be used by an initiator. During this time, however, it is possible to have an initiator discover this LUN. The Solaris initiator waits until it receives an Inventory Change notification and then automatically brings the device online.

▼ How to Create an iSCSI Target

This procedure assumes that you are logged in to the local system that contains the iSCSI targets.

1 Become superuser.

2 Create a ZFS pool and volume.
   
   target# zpool create sanpool mirror c1t2d0 c1t3d0
   target# zfs create sanpool/voll

3 Identify the target’s backing store directory.
   For example:
   
   target# iscsitadm modify admin -d /sanpool/voll

4 Create an iSCSI target.
   For example:
   
   target# iscsitadm create target --size 2g sanpool
5 Display information about the iSCSI target.
   For example:
   target# iscsitadm list target -v sanpool

6 Set up your iSCSI initiator to discover and use this target.
   For more information, see “How to Configure iSCSI Target Discovery” on page 240.

▼ How to Configure iSCSI Target Discovery
   This procedure assumes that you are logged in to the local system where you want to configure
   access to an iSCSI target device.

1 Become superuser.

2 Configure the target device to be discovered dynamically or statically using one of the following
   methods:
   ■ Configure the device to be dynamically discovered (SendTargets).
     For example:
     initiator# iscsiadm add discovery-address 10.0.0.1:3260
   ■ Configure the device to be dynamically discovered (iSNS).
     For example:
     initiator# iscsiadm add iSNS-server 10.0.0.1:3205
   ■ Configure the device to be statically discovered.
     For example:
     initiator# iscsiadm add static-config eui.5000ABCD78945E2B,10.0.0.1
   The iSCSI connection is not initiated until the discovery method is enabled. See the next step.

3 Enable the iSCSI target discovery method using one of the following:
   ■ If you have configured a dynamically discovered (SendTargets) device, enable the
     SendTargets discovery method.
     initiator# iscsiadm modify discovery --sendtargets enable
   ■ If you have configured a dynamically discovered (iSNS) device, enable the iSNS discovery
     method.
     initiator# iscsiadm modify discovery --iSNS enable
   ■ If you have configured static targets, enable the static target discovery method.
     initiator# iscsiadm modify discovery --static enable

4 Create the iSCSI device links for the local system.
   initiator# devfsadm -i iscsi
How to Configure iSNS Discovery for the Solaris iSCSI Target

If your network includes a third-party iSNS server or a Sun iSNS server, you can set up iSNS target discovery on your Solaris iSCSI targets.

This procedure assumes that you are logged in to the local system where you want to access a configured iSCSI target device.

1. Become superuser.

2. Add the iSNS server information.
   For example:
   ```bash
   initiator# iscsitadm modify admin --isns-server ip-address or hostname[:port]
   ```
   Identify the `ip-address` of the iSNS server in your network.
   This step adds the iSNS server information to all of the Solaris iSCSI targets.

3. Enable iSNS server discovery.
   For example:
   ```bash
   initiator# iscsitadm modify admin --isns-access enable
   ```
   This step enables iSNS discovery for all of the Solaris iSCSI targets.

How to Access iSCSI Disks

If you want to access the iSCSI disks upon reboot, create the file system on the disk, and add an `/etc/vfstab` entry as you would with a UFS file system on a SCSI device. Then, create a new SMF service for mounting iSCSI disks that depends on the iSCSI initiator service. For more information, see “How to Access iSCSI Disks Upon Reboot” on page 242.

After the devices have been discovered by the Solaris iSCSI initiator, the login negotiation occurs automatically. The Solaris iSCSI driver determines the number of available LUNs and creates the device nodes. Then, the iSCSI devices can be treated as any other SCSI device.

You can view the iSCSI disks on the local system by using the `format` utility.

In the following `format` output, disks 2 and 3 are iSCSI LUNs that are not under MPxIO control. Disks 21 and 22 are iSCSI LUNs under MPxIO control.
3. c3t1d0 <ABCSTORAGE-100E-00-2.2 cyl 20813 alt 2 hd 16 sec 63>
   /iscsi/disk@0000/hostname-010000,0

21. c4t60A98000686F694B2F59775733426B77d0 <ABCSTORAGE-LUN-0.2 cyl 4606 alt 2 hd 16 sec 256>
   /scsi_vhci/ssd@g60a98000686f694b2f59775733434C41

### How to Access iSCSI Disks Upon Reboot

Follow the steps below to access iSCSI disks after the system is rebooted.

1. Become superuser or assume an equivalent role.

2. Add entries for the iSCSI LUN(s) to the /etc/vfstab file. Set the mount at boot option to iscsi.

```bash
initiator# vi /etc/vfstab
#device device mount FS fsck mount mount
#to mount to fscck point type pass at boot options
/dev/dsk/c3t600144F04B555F370000093D00495B00d0s0 - /mnt ufs - no -
```

### How to Remove Discovered iSCSI Targets

After removing a discovery address, iSNS server, or static configuration, or after disabling a discovery method, the associated targets are logged out. If these associated targets are still in use, for example, they have mounted file systems, the logout of these devices will fail, and they will remain on the active target list.

This optional procedure assumes that you are logged into the local system where access to an iSCSI target device has already been configured.

1. Become superuser.

2. (Optional) Disable an iSCSI target discovery method using one of the following:
   - If you need to disable the SendTargets discovery method, use the following command:
     ```bash
     initiator# iscsiadm modify discovery --sendtargets disable
     ```
   - If you need to disable the iSNS discovery method, use the following command:
     ```bash
     initiator# iscsiadm modify discovery --iSNS disable
     ```
   - If you need to disable the static target discovery method, use the following command:
     ```bash
     initiator# iscsiadm modify discovery --static disable
     ```

3. Remove an iSCSI device discovery entry using one of the following:
Remove an iSCSI Send Targets discovery entry.
For example:

```
initiator# iscsiadm remove discovery-address 10.0.0.1:3260
```

Remove an iSCSI iSNS discovery entry.
For example:

```
# iscsiadm remove isns-server 10.0.0.1:3205
```

Remove a static iSCSI discovery entry.
For example:

```
initiator# iscsiadm remove static-config eui.5000ABCD78945E2B,10.0.0.1
```

**Note** - If you attempt to disable or remove a discovery entry that has an associated logical unit in use, the disable or remove operation fails with the following message:

```
logical unit in use
```

If this error occurs, stop all associated I/O on the logical unit, unmount the file systems, and so on. Then, repeat the disable or remove operation.

4 **Remove the iSCSI target device.**

Remove a target by specifying the logical unit number (LUN). If you did not specify a LUN when the target was created, a value of 0 was used. LUN 0 must be the last LUN removed if multiple LUNs are associated with a target.

For example:

```
initiator# iscsitadm delete target --lun 0 sandbox
```

## Configuring Authentication in Your iSCSI-Based Storage Network

Setting up authentication for your iSCSI devices is optional.

In a secure environment, authentication is not required because only trusted initiators can access the targets.

In a less secure environment, the target cannot determine if a connection request is truly from a given host. In that case, the target can authenticate an initiator by using the Challenge-Handshake Authentication Protocol (CHAP).

CHAP authentication uses the notion of a challenge and response, which means that the target challenges the initiator to prove its identity. For the challenge/response method to work, the
iSCSI supports unidirectional and bidirectional authentication:

- **Unidirectional authentication** enables the target to authenticate the identity of the initiator.
- **Bidirectional authentication** adds a second level of security by enabling the initiator to authenticate the identity of the target.

▼ **How to Configure CHAP Authentication for Your iSCSI Initiator**

This procedure assumes that you are logged in to the local system where you want to securely access the configured iSCSI target device.

1. **Become superuser.**

2. **Determine whether you want to configure unidirectional or bidirectional CHAP.**

   - Unidirectional authentication, the default method, enables the target to validate the initiator. Complete steps 3–5 only.
   - Bidirectional authentication adds a second level of security by enabling the initiator to authenticate the target. Complete steps 3–9.

3. **Unidirectional CHAP – Set the secret key on the initiator.**

   For example, the following command initiates a dialogue to define the CHAP secret key.
   
   ```
   initiator# iscsiadm modify initiator-node --CHAP-secret
   ```

   **Note** – The CHAP secret length must be a minimum of 12 characters and a maximum of 16 characters.

4. **(Optional) Unidirectional CHAP – Set the CHAP name on the initiator.**

   By default, the initiator’s CHAP name is set to the initiator node name.

   You can use the following command to change the initiator’s CHAP name.
   
   ```
   initiator# iscsiadm modify initiator-node --CHAP-name new-CHAP-name
   ```

   In the Solaris environment, the CHAP name is always set to the initiator node name by default. The CHAP name can be set to any length text that is less than 512 bytes. The 512-byte length limit is a Solaris limitation. However, if you do not set the CHAP name, it is set to the initiator node name upon initialization.
Unidirectional CHAP – Enable CHAP authentication on the initiator after the secret has been set.

```
initiator# iscsiadm modify initiator-node --authentication CHAP
```

CHAP requires that the initiator node have both a username and a password. The user name is typically used by the target to look up the secret for the given username.

Select one of the following to enable or disable Bidirectional CHAP.

- **Bidirectional CHAP** – Enable bidirectional authentication parameters on the target.
  
  For example:
  
  ```
  initiator# iscsiadm modify target-param -B enable eui.5000ABC78945E2B
  ```

- **Disable bidirectional CHAP**. For example:
  
  ```
  initiator# iscsiadm modify target-param -B disable eui.5000ABC78945E2B
  ```

Bidirectional CHAP – Set the authentication method to CHAP on the target.

For example:

```
initiator# iscsiadm modify target-param --authentication CHAP eui.5000ABC78945E2B
```

Bidirectional CHAP – Set the target device secret key on the target.

For example, the following command initiates a dialogue to define the CHAP secret key:

```
initiator# iscsiadm modify target-param --CHAP-secret eui.5000ABC78945E2B
```

Bidirectional CHAP – Set the CHAP name on the target.

By default, the target's CHAP name is set to the target name.

You can use the following command to change the target's CHAP name:

```
initiator# iscsiadm modify target-param --CHAP-name target-CHAP-name
```

How to Configure CHAP Authentication for Your iSCSI Target

This procedure assumes that you are logged in to the local system that contains the iSCSI targets.

1. **Become superuser.**

2. **Set the CHAP secret name for the target.**

   A convention is to use the host name for the secret name. For example:

   ```
target# iscsitadm modify admin -H stormpike
   ```
Specify the CHAP secret.
The CHAP secret must be between 12 and 16 characters. For example:

```
target# iscsitadm modify admin -C
Enter secret: xxxxxx
Re-enter secret: xxxxxx
```

Create an initiator object that will be associated with one or more targets.
This step is done so that you can associate a friendly name (normally the host name, in this case `monster620`) with the IQN value, instead of typing it in every time. For example:

```
# iscsitadm create initiator -n iqn.1986-03.com.sun: 01:00e081553307.4399f40e monster620
```

Provide the same CHAP name that was used on the initiator.
This name can be different from the friendly name that was used for the initiator object. For example:

```
target# iscsitadm modify initiator -H monster620 monster620
```

Use the same CHAP secret that was used on the initiator.
For example:

```
target# iscsitadm modify initiator -C monster620
Enter secret: xxxxxx
Re-enter secret: xxxxxx
```

Associate the initiator object with one or more targets.
For example:

```
target# iscsitadm modify target -l monster620 sandbox
```

Using a Third-Party RADIUS Server to Simplify CHAP Management in Your iSCSI Configuration

You can use a third-party RADIUS server to simplify CHAP secret management. A RADIUS server is a centralized authentication service. While you must still specify the initiator’s CHAP secret, you are no longer required to specify each target’s CHAP secret on each initiator when using bidirectional authentication with a RADIUS server.

For more information, see:

How to Configure RADIUS for Your iSCSI Configuration

This procedure assumes that you are logged in to the local system where you want to securely access the configured iSCSI target device.
1. Become superuser.

2. Configure the initiator node with the IP address and port (the default port is 1812) of the RADIUS server.
   For example:
   
   ```bash
   initiator# iscsiadm modify initiator-node --radius-server 10.0.0.72:1812
   ```

3. Configure the initiator node with the shared secret of the RADIUS server.
   ```bash
   initiator# iscsiadm modify initiator-node --radius-shared-secret
   ```

   **Note** – The Solaris iSCSI implementation requires that the RADIUS server is configured with a shared secret before the Solaris iSCSI software can interact with the RADIUS server.

4. Enable the RADIUS server.
   ```bash
   initiator# iscsiadm modify initiator-node --radius-access enable
   ```

### Solaris iSCSI and RADIUS Server Error Messages

This section describes the error messages that are related to a Solaris iSCSI and RADIUS server configuration, along with potential solutions for recovery.

**empty RADIUS shared secret**

  **Cause**: The RADIUS server is enabled on the initiator, but the RADIUS shared secret is not set.

  **Solution**: Configure the initiator with the RADIUS shared secret. For more information, see "How to Configure RADIUS for Your iSCSI Configuration" on page 246.

**WARNING: RADIUS packet authentication failed**

  **Cause**: The initiator failed to authenticate the RADIUS data packet. This error can occur if the shared secret configured on the initiator node is different from the shared secret on the RADIUS server.

  Reconfigure the initiator with the correct RADIUS shared secret. For more information, see "How to Configure RADIUS for Your iSCSI Configuration" on page 246.

---

### Setting Up Solaris iSCSI Multipathed Devices

Consider the following guidelines for using Solaris iSCSI multipathed (MPxIO) devices:

- **Solaris iSCSI and MPxIO** – MPxIO supports target port aggregation and availability in Solaris iSCSI configurations that configure multiple sessions per target (MS/T) on the iSCSI initiator.
- Use IPMP for aggregation and failover of two or more NICs.
- A basic configuration for an iSCSI host is a server with two NICs that are dedicated to iSCSI traffic. The NICs are configured by using IPMP. Additional NICs are provided for non-iSCSI traffic to optimize performance.
- Active multipathing can only be achieved by using the Solaris iSCSI MS/T feature, and the failover and redundancy of an IPMP configuration.
  - If one NIC fails in an IPMP configuration, IPMP handles the failover. The MPxIO driver does not notice the failure. In a non-IPMP configuration, the MPxIO driver fails and offline the path.
  - If one target port fails in an IPMP configuration, the MPxIO driver notices the failure and provides the failover. In a non-IPMP configuration, the MPxIO driver notices the failure and provides the failover.
- For information about configuring multiple sessions per target, see “How to Enable Multiple iSCSI Sessions for a Target” on page 248. For information about configuring IPMP, see Part V, “IPMP,” in Oracle Solaris Administration: IP Services.

Solaris iSCSI, Fibre-Channel (FC), and MPxIO – The MPxIO driver provides the following behavior in more complex iSCSI/FC configurations:
- If you have dual iSCSI to FC bridges in an FC SAN, iSCSI presents target paths to MPxIO. MPxIO matches the unique SCSI per LUN identifier, and if they are identical, presents one path to the iSCSI driver.
- If you have a configuration that connects a target by using both iSCSI and FC, the MPxIO driver can provide different transports to the same device. In this configuration, MPxIO utilizes both paths.
- If you are using iSCSI and FC in combination with MPxIO, make sure that the MPxIO settings in the /kernel/drv/fp.conf file and the /kernel/drv/iscsi.conf files match the MPxIO configuration that you want supported. For example, in fp.conf, you can determine whether MPxIO is enabled globally on the HBA or on a per-port basis.

Third-party hardware considerations – Find out if your third-party HBA is qualified to work with Solaris iSCSI and MPxIO.

If you are using a third-party HBA, you might need to ask your third-party HBA vendor for the symmetric-option information for the /kernel/drv/scsi_vhci.conf file.

▶ How to Enable Multiple iSCSI Sessions for a Target

This procedure can be used to create multiple iSCSI sessions that connect to a single target. This scenario is useful with iSCSI target devices that support login redirection or have multiple target portals in the same target portal group. Use iSCSI multiple sessions per target with Solaris SCSI Multipathing (MPxIO). You can also achieve higher bandwidth if you utilize multiple NICs on the host side to connect to multiple portals on the same target.
The MS/T feature creates two or more sessions on the target by varying the initiator’s session ID (ISID). Enabling this feature creates two SCSI layer paths on the network so that multiple targets are exposed through the iSCSI layer to the Solaris I/O layer. The MPxIO driver handles the reservations across these paths.

For more information about how iSCSI interacts with MPxIO paths, see "Setting Up Solaris iSCSI Multipathed Devices" on page 247.

Review the following items before configuring multiple sessions for an iSCSI target:

- A typical MS/T configuration has two or more configured sessions. However, if your storage supports multiple TPGTs and if you are using SendTarget discovery on your host system, then the number of configured sessions can be set to 1. SendTarget discovery automatically detects the existence of multiple paths and multiple target sessions are created.

- Confirm that the mxpio configuration parameter is enabled in the /kernel/drv/iscsi.conf file.

  ```bash
  # cd /kernel/drv
  # grep mxpio iscsi.conf
  iscsi.conf:mxpio-disable="no";
  ```

- Confirm that the multiple network connections are configured by using IPMP.

- Confirm that the multiple network connections are available.

  ```bash
  # ifconfig -a
  ```

1. Become superuser.

2. List the current parameters for the iSCSI initiator and target.

   a. List the current parameters for the iSCSI initiator. For example:

      ```bash
      initiator# iscsiadm list initiator-node
      Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
      Initiator node alias: zzr1200
      .
      .
      Configured Sessions: 1
      ```

   b. List the current parameters of the iSCSI target device. For example:

      ```bash
      initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
      Alias: -
      .
      Configured Sessions: 1
      ```

   The configured sessions value is the number of configured iSCSI sessions that will be created for each target name in a target portal group.
3. Select one of the following to modify the number of configured sessions either at the initiator node to apply to all targets or at a target level to apply to a specific target.

The number of sessions for a target must be between 1 and 4.

- Apply the parameter to the iSCSI initiator node.
  For example:
  ```shell
  initiator# iscsiadm modify initiator-node -c 2
  ```

- Apply the parameter to the iSCSI target.
  For example:
  ```shell
  ```

- Bind configured sessions to one or more local IP addresses.
  Configured sessions can also be bound to a specific local IP address. Using this method, one or more local IP addresses are supplied in a comma-separated list. Each IP address represents an iSCSI session. This method can also be done at the initiator-node or target-param level. For example:
  ```shell
  initiator# iscsiadm modify initiator-node -c 10.0.0.1,10.0.0.2
  ```

  **Note** – If the specified IP address is not routable, the address is ignored and the default Solaris route and IP address is used for this session.

4. Verify that the parameter was modified.

   a. Display the updated information for the initiator node. For example:
      ```shell
      initiator# iscsiadm list initiator-node
      Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c933
      Initiator node alias: zzr1200
      .
      .
      Configured Sessions: 2
      ```

   b. Display the updated information for the target node. For example:
      ```shell
      initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
      Alias: -
      .
      .
      Configured Sessions: 2
      ```

5. List the multiple paths by using the `mpathadm list lu` command to confirm that the OS device name matches the `iscsiadm list` output, and that the path count is 2 or more.
Monitoring Your iSCSI Configuration

You can display information about the iSCSI initiator and target devices by using the `iscsiadm list` command.

1 Become superuser.

2 Display information about the iSCSI initiator.
   For example:
   ```
   # iscsiadm list initiator-node
   Initiator node name: iqn.1996-03.com.sun:01:0003ba4d233b.425c293c
   Initiator node alias: zzr1200
   Login Parameters (Default/Configured):
   Header Digest: NONE/-
   Data Digest: NONE/-
   Authentication Type: NONE
   RADIUS Server: NONE
   RADIUS access: unknown
   Configured Sessions: 1
   ```

3 Display information about which discovery methods are in use.
   For example:
   ```
   # iscsiadm list discovery
   Discovery:
   Static: enabled
   Send Targets: enabled
   iSNS: enabled
   ```

Example 12–1 Displaying iSCSI Target Information

The following example shows how to display the parameter settings for a specific iSCSI target.

```
# iscsiadm list target-param iqn.1992-08.com.abcstorage:sn.33592219
```
# iscsiadm list target-param -v eui.50060e8004275511 Target: eui.50060e8004275511
   
   Alias: -
   Bi-directional Authentication: disabled
   Authentication Type: NONE
   Login Parameters (Default/Configured):
      Data Sequence In Order: yes/-
      Data PDU In Order: yes/-
      Default Time To Retain: 20/-
      Default Time To Wait: 2/-
      Error Recovery Level: 0/-
      First Burst Length: 65536/-
      Immediate Data: yes/-
      Initial Ready To Transfer (R2T): yes/-
      Max Burst Length: 262144/-
      Max Outstanding R2T: 1/-
      Max Receive Data Segment Length: 65536/-
      Max Connections: 1/-
      Header Digest: NONE/-
      Data Digest: NONE/-

   Configured Sessions: 1

The following example output displays the parameters that were negotiated between the target and the initiator.

# iscsiadm list target -v eui.50060e8004275511
Target: eui.50060e8004275511
   
   TPGT: 1
   ISID: 4000002a0000
   Connections: 1
   CID: 0
   IP address (Local): 172.90.101.71:32813
   IP address (Peer): 172.90.101.40:3260
   Discovery Method: Static
   Login Parameters (Negotiated):
      Data Sequence In Order: yes
      Data PDU In Order: yes
      Default Time To Retain: 0
      Default Time To Wait: 3
      Error Recovery Level: 0
      First Burst Length: 65536
      Immediate Data: yes
      Initial Ready To Transfer (R2T): yes
      Max Burst Length: 262144
      Max Outstanding R2T: 1
      Max Receive Data Segment Length: 65536
      Max Connections: 1
      Header Digest: NONE
      Data Digest: NONE
Modifying iSCSI Initiator and Target Parameters

You can modify parameters on both the iSCSI initiator and the iSCSI target device. However, the only parameters that can be modified on the iSCSI initiator are the following:

- iSCSI initiator node name – You can change the initiator node name to a different name. If you change the initiator node name, the targets that were discovered by iSNS might be removed from the initiator’s target list, depending on the discovery domain configuration on the iSNS server at the time when the name was changed. For more information, see “How to Modify iSCSI Initiator and Target Parameters” on page 253.

- Header digest – NONE, the default value or CRC32.

- Data digest – NONE, the default value or CRC32.

- Authentication and CHAP secret – For more information about setting up authentication, see “How to Configure CHAP Authentication for Your iSCSI Target” on page 245.

- Configured sessions – For more information about configuring multiple sessions, see “How to Enable Multiple iSCSI Sessions for a Target” on page 248.

The iSCSI driver provides default values for the iSCSI initiator and iSCSI target device parameters. If you modify the parameters of the iSCSI initiator, the modified parameters are inherited by the iSCSI target device, unless the iSCSI target device already has different values.

Caution – Ensure that the target software supports the parameter to be modified. Otherwise, you might be unable to log in to the iSCSI target device. See your array documentation for a list of supported parameters.

Modifying iSCSI parameters should be done when I/O between the initiator and the target is complete. The iSCSI driver reconnects the session after the changes are made by using the `iscsiadm modify` command.

How to Modify iSCSI Initiator and Target Parameters

The first part of this procedure illustrates how modified parameters of the iSCSI initiator are inherited by the iSCSI target device. The second part of this procedure shows how to actually modify parameters on the iSCSI target device.

This optional procedure assumes that you are logged in to the local system where access to an iSCSI target device has already been configured.

1 Become superuser.
2 List the current parameters of the iSCSI initiator and target device.

a. List the current parameters of the iSCSI initiator. For example:

```
initiator# iscsiadm list initiator-node
Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
Initiator node alias: zzr1200
Login Parameters (Default/Configured):
  Header Digest: NONE/-
  Data Digest: NONE/-
  Authentication Type: NONE
RADIUS Server: NONE
RADIUS access: unknown
Configured Sessions: 1
```

b. List the current parameters of the iSCSI target device. For example:

```
initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
  Alias: -
  Bi-directional Authentication: disabled
  Authentication Type: NONE
Login Parameters (Default/Configured):
  Data Sequence In Order: yes/-
  Data PDU In Order: yes/-
  Default Time To Retain: 20/-
  Default Time To Wait: 2/-
  Error Recovery Level: 0/-
  First Burst Length: 65536/-
  Immediate Data: yes/-
  Initial Ready To Transfer (R2T): yes/-
  Max Burst Length: 262144/-
  Max Outstanding R2T: 1/-
  Max Receive Data Segment Length: 65536/-
  Max Connections: 1/-
  Header Digest: NONE/-
  Data Digest: NONE/-
Configured Sessions: 1
```

Note that both header digest and data digest parameters are currently set to NONE for both the iSCSI initiator and the iSCSI target device.

To review the default parameters of the iSCSI target device, see the `iscsiadm list target-param` output in Example 12–1.

3 Modify the parameter of the iSCSI initiator.

For example, set the header digest to CRC32.

```
initiator# iscsiadm modify initiator-node -h CRC32
```

If you change the initiator node name, the targets that were discovered by iSNS might be logged out and removed from the initiator’s target list, if the new name does not belong to the same discovery domain as that of the targets. However, if the targets are in use, they are not removed. For example, if a file is open or a file system is mounted on these targets, the targets will not be removed.
You might also see new targets after the name change if these targets and the new initiator node name belong to the same discovery domain.

4 Verify that the parameter was modified.

a. Display the updated parameter information for the iSCSI initiator. For example:

   initiator# iscsiadm list initiator-node
   Initiator node name: iqn.1986-03.com.sun:01:0003ba4d233b.425c293c
   Initiator node alias: zzr1200
   Login Parameters (Default/Configured):
     Header Digest: NONE/CRC32
     Data Digest: NONE/-
     Authentication Type: NONE
     RADIUS Server: NONE
     RADIUS access: unknown
   Configured Sessions: 1

   Note that the header digest is now set to CRC32.

b. Display the updated parameter information for the iSCSI target device. For example:

   initiator# iscsiadm list target-param -v iqn.1992-08.com.abcstorage:sn.84186266
   Alias: -
   Bi-directional Authentication: disabled
   Authentication Type: NONE
   Login Parameters (Default/Configured):
     Data Sequence In Order: yes/-
     Data PDU In Order: yes/-
     Default Time To Retain: 20/-
     Default Time To Wait: 2/-
     Error Recovery Level: 0/-
     First Burst Length: 65536/-
     Immediate Data: yes/-
     Initial Ready To Transfer (R2T): yes/-
     Max Burst Length: 262144/-
     Max Connections: 1/-
     Max Outstanding R2T: 1/-
     Max Receive Data Segment Length: 65536/-
     Header Digest: CRC32/-
     Data Digest: NONE/-
   Configured Sessions: 1

   Note that the header digest is now set to CRC32.

5 Verify that the iSCSI initiator has reconnected to the iSCSI target. For example:

   initiator# iscsiadm list target -v iqn.1992-08.com.abcstorage:sn.84186266
   TPGT: 2
   ISID: 4000002a0000
   Connections: 1
   CID: 0
   IP address (Local): nnn.nn.nn.nnn:64369
   IP address (Peer): nnn.nn.nn.nnn:3260
   Discovery Method: SendTargets
Login Parameters (Negotiated):

Header Digest: CRC32
Data Digest: NONE

6  (Optional) Unset an iSCSI initiator parameter or an iSCSI target device parameter.
You can unset a parameter by setting it back to its default setting by using the `iscsiadm modify` command. Or, you can use the `iscsiadm remove` command to reset all target properties to the default settings.

The `iscsiadm modify target-param` command changes only the parameters that are specified on the command line.

The following example shows how to reset the header digest to NONE:

```
initiator# iscsiadm modify target-param -h none iqn.1992-08.com.abcstorage:sn...
```

For information about the `iscsiadm remove target-param` command, see `iscsiadm(1M)`.

## Troubleshooting iSCSI Configuration Problems

The following tools are available to troubleshoot general iSCSI configuration problems:

- `snoop` – This tool has been updated to support iSCSI packets.
- `wireshark` – This product is available from [http://www.wireshark.org/](http://www.wireshark.org/).

Both tools can filter iSCSI packets on port 3260.

The following sections describe various iSCSI troubleshooting and error message resolution scenarios.

### No Connections to the iSCSI Target From the Local System

#### How to Troubleshoot iSCSI Connection Problems

1  Become superuser.

2  List your iSCSI target information.
   For example:

```
initiator# iscsiadm list target
Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zrr1200-01
TPGT: default
ISID: 4000002a0000
Connections: 0
```
3 If no connections are listed in the iscsiadm list target output, check the
/var/adm/messages file for possible reasons why the connection failed.
You can also verify whether the connection is accessible by using the ping command or by
connecting to the storage device’s iSCSI port by using the telnet command to ensure that the
iSCSI service is available. The default port is 3260.
In addition, check the storage device’s log file for errors.

4 If your target is not listed in the iscsiadm list target output, check the /var/adm/messages
file for possible causes.
If you are using SendTargets as the discovery method, try listing the discovery-address using the
-v option to ensure that the expected targets are visible to the host. For example:

initiator# iscsiadm list discovery-address -v 10.0.0.1
Discovery Address: 10.0.0.1:3260
  Target name: eui.210000203787dfc0
    Target address: 10.0.0.1:11824
  Target name: eui.210000203787e07b
    Target address: 10.0.0.1:11824

If you are using iSNS as the discovery method, try enabling the iSNS discovery method and
listing the isns-server using the -v option to ensure that the expected targets are visible to the
host. For example:

initiator# iscsiadm list isns-server -v
iSNS Server IP Address: 10.20.56.56:3205
  Target name: iqn.1992-08.com.xyz:sn.1234566
    Target address: 10.20.56.206:3260, 1
    Target address: 10.20.56.206:3260, 1
    Target address: 10.20.56.206:3260, 1

iSCSI Device or Disk Is Not Available on the Local
System

▼ How to Troubleshoot iSCSI Device or Disk Unavailability

1 Become superuser.

2 Identify the LUNs that were discovered on this target during enumeration.
For example:

# iscsiadm list target -S
Target: iqn.2001-05.com.abcstorage:6-8a0900-37ad70401-bcfff02df8a421df-zzr1200-01
  TPGT: default
The -S option shows which LUNs were discovered on this target during enumeration. If you think a LUN should be listed but it is not, review the /var/adm/messages file to see if an error was reported. Check the storage device’s log files for errors. Also, ensure that any storage device LUN masking is properly configured.

**Use LUN Masking When Using the iSNS Discovery Method**

Avoid using the iSNS discovery domain as the means to control storage authorization to specific initiators. Use LUN masking instead if you want to make sure that only authorized initiators can access a LUN.

If you remove a target from a discovery domain while the target is in use, the iSCSI initiator does not log out from this target. If you do not want this initiator to access this target (and the associated LUNs), you must use LUN masking. Removing the target from the discovery domain is not sufficient.

**General iSCSI Error Messages**

This section describes the iSCSI messages that might be found in the /var/adm/messages file and potential solutions for recovery.

The message format is as follows:

```plaintext
iscsi TYPE (OID) STRING (STATUS-CLASS#/STATUS-DETAIL#)
```

<table>
<thead>
<tr>
<th>TYPE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECTION</td>
<td>Is either connection or session.</td>
</tr>
<tr>
<td>OID</td>
<td>Is the object ID of the connection or session. This ID is unique for an OS instance.</td>
</tr>
<tr>
<td>STRING</td>
<td>Is a description of the condition.</td>
</tr>
<tr>
<td>STATUS-CLASS#/STATUS-DETAIL#</td>
<td>These values are returned in an iSCSI login response as defined by RFC 3720.</td>
</tr>
</tbody>
</table>

**iscsi connection(OID) login failed - Miscellaneous iSCSI initiator errors.**

**Cause:** The device login failed due to some form of initiator error.
iscsi connection(OID) login failed - Initiator could not be successfully authenticated.
  Cause: The device could not successfully authenticate the initiator.
  Solution: If applicable, verify that the settings for CHAP names, CHAP passwords, or the RADIUS server are correct.

iscsi connection(OID) login failed - Initiator is not allowed access to the given target.
  Cause: The device cannot allow the initiator access to the iSCSI target device.
  Solution: Verify your initiator name and confirm that it is properly masked or provisioned by the storage device.

iscsi connection(OID) login failed - Requested ITN does not exist at this address.
  Cause: The device does not provide access to the iSCSI target name (ITN) that you are requesting.
  Solution: Verify that the initiator discovery information is specified properly and that the storage device is configured properly.

iscsi connection(OID) login failed - Requested ITN has been removed and no forwarding address is provided.
  Cause: The device can no longer provide access to the iSCSI target name (ITN) that you are requesting.
  Solution: Verify that the initiator discovery information has been specified properly and that the storage device has been configured properly.

iscsi connection(OID) login failed - Requested iSCSI version range is not supported by the target.
  Cause: The initiator’s iSCSI version is not supported by the storage device.

iscsi connection(OID) login failed - No more connections can be accepted on this Session ID (SSID).
  Cause: The storage device cannot accept another connection for this initiator node to the iSCSI target device.

iscsi connection(OID) login failed - Missing parameters (e.g., iSCSI initiator and/or target name).
  Cause: The storage device is reporting that the initiator or target name has not been properly specified.
  Solution: Properly specify the iSCSI initiator or target name.
iscsi connection \texttt{(OID)} login failed - Target hardware or software error.
\textbf{Cause:} The storage device encountered a hardware or software error.

\textbf{Solution:} Consult the storage documentation, or contact the storage vendor for further assistance.

iscsi connection \texttt{(OID)} login failed - iSCSI service or target is not currently operational.
\textbf{Cause:} The storage device is currently not operational.

\textbf{Solution:} Consult the storage documentation, or contact the storage vendor for further assistance.

iscsi connection \texttt{(OID)} login failed - Target has insufficient session, connection or other resources.
\textbf{Cause:} The storage device has insufficient resources.

\textbf{Solution:} Consult the storage documentation, or contact the storage vendor for further assistance.

iscsi connection \texttt{(OID)} login failed - unable to initialize authentication
iscsi connection \texttt{(OID)} login failed - unable to set authentication
iscsi connection \texttt{(OID)} login failed - unable to set username
iscsi connection \texttt{(OID)} login failed - unable to set password
iscsi connection \texttt{(OID)} login failed - unable to set ipsec
iscsi connection \texttt{(OID)} login failed - unable to set remote authentication
\textbf{Cause:} The initiator was unable to initialize or set authentication properly.

\textbf{Solution:} Verify that your initiator settings for authentication are properly configured.

iscsi connection \texttt{(OID)} login failed - unable to make login pdu
\textbf{Cause:} The initiator was unable to make a login payload data unit (PDU) based on the initiator or storage device settings.

\textbf{Solution:} Try resetting any target login parameters or other nondefault settings.

iscsi connection \texttt{(OID)} login failed - failed to transfer login
iscsi connection \texttt{(OID)} login failed - failed to receive login response
\textbf{Cause:} The initiator failed to transfer or receive a login payload data unit (PDU) across the network connection.

\textbf{Solution:} Verify that the network connection is reachable.
iscsi connection(OID) login failed - received invalid login response (OP CODE)
   Cause: The storage device has responded to a login with an unexpected response.

iscsi connection(OID) login failed - login failed to authenticate with target
   Cause: The initiator was unable to authenticate the storage device.
   Solution: Verify that your initiator settings for authentication are properly configured.

iscsi connection(OID) login failed - initiator name is required
   Cause: An initiator name must be configured to perform all actions.
   Solution: Verify that the initiator name is configured.

iscsi connection(OID) login failed - authentication receive failed
iscsi connection(OID) login failed - authentication transmit failed
   Cause: The initiator was unable to transmit or receive authentication information.
   Solution: Verify network connectivity with the storage device or the RADIUS server, as applicable.

iscsi connection(OID) login failed - login redirection invalid
   Cause: The storage device attempted to redirect the initiator to an invalid destination.
   Solution: Consult the storage documentation, or contact the storage vendor for further assistance.

iscsi connection(OID) login failed - target protocol group tag mismatch,
   expected <TPGT>, received <TPGT>
   Cause: The initiator and target had a TPGT (target portal group tag) mismatch.
   Solution: Verify your TPGT discovery settings on the initiator or the storage device.

iscsi connection(OID) login failed - can’t accept PARAMETER in security stage
   Cause: The device responded with an unsupported login parameter during the security phase of login.
   Solution: The parameter name is noted for reference. Consult the storage documentation, or contact the storage vendor for further assistance.
iscsi connection (OID) login failed - HeaderDigest=CRC32 is required, can’t accept VALUE
iscsi connection (OID) login failed - DataDigest=CRC32 is required, can’t accept VALUE
  Cause: The initiator is only configured to accept a HeaderDigest or DataDigest that is set to CRC32 for this target. The device returned the value of VALUE.
  Solution: Verify that the initiator and device digest settings are compatible.
iscsi connection (OID) login failed - HeaderDigest=None is required, can’t accept VALUE
iscsi connection (OID) login failed - DataDigest=None is required, can’t accept VALUE
  Cause: The initiator is only configured to accept a HeaderDigest or DataDigest that is set to NONE for this target. The device returned the value of VALUE.
  Solution: Verify that the initiator and device digest settings are compatible.
iscsi connection (OID) login failed - can’t accept PARAMETER
  Cause: The initiator does not support this parameter.
iscsi connection (OID) login failed - can’t accept MaxOutstandingR2T VALUE
  Cause: The initiator does not accept MaxOutstandingR2T of the noted VALUE.
iscsi connection (OID) login failed - can’t accept MaxConnections VALUE
  Cause: The initiator does not accept the maximum connections of the noted VALUE.
iscsi connection (OID) login failed - can’t accept ErrorRecoveryLevel VALUE
  Cause: The initiator does not accept an error recovery level of the noted VALUE.
iscsi session (OID) NAME offline
  Cause: All connections for this target NAME have been removed or have failed.
iscsi connection (OID) failure - unable to schedule enumeration
  Cause: The initiator was unable to enumerate the LUNs on this target.
  Solution: You can force LUN enumeration by running the devfsadm -i iscsi command. For more information, see devfsadm(1M).
iscsi connection (OID) unable to connect to target NAME (errno:ERRNO)
  Cause: The initiator failed to establish a network connection.
  Solution: For information about the specific ERRNO on the connection failure, see the /usr/include/sys/errno.h file.
This chapter describes the `format` utility’s menus and commands.

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

- “Recommendations and Requirements for Using the `format` Utility” on page 263
- “`format` Menu and Command Descriptions” on page 263
- “Rules for Input to `format` Commands” on page 269
- “Getting Help on the `format` Utility” on page 271

For a overview of when to use the `format` utility, see “`format` Utility” on page 159.

**Recommendations and Requirements for Using the `format` Utility**

You must become superuser to use the `format` utility. Otherwise, the following error message is displayed when you try to use the `format` utility:

```
$ format
Searching for disks...done
No permission (or no disks found)!
```

**`format` Menu and Command Descriptions**

The `format` menu contents are displayed as follows:

```
FORMAT MENU:
  disk  - select a disk
  type  - select (define) a disk type
  partition  - select (define) a partition table
  current  - describe the current disk
```
format Menu and Command Descriptions

- format - format and analyze the disk
- fdisk - run the fdisk program (x86 only)
- 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
- !<cmd> - execute <cmd>, then return
- quit

format>

The following table describes the main menu items for the format utility.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Command or Menu?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Command</td>
<td>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.</td>
</tr>
<tr>
<td>type</td>
<td>Command</td>
<td>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.</td>
</tr>
<tr>
<td>partition</td>
<td>Menu</td>
<td>Creates and modifies slices. For more information, see &quot;partition Menu&quot; on page 265.</td>
</tr>
</tbody>
</table>
| 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 type at the prompt if no format.dat entry exists  

This command does not apply to IDE disks. IDE disks are preformatted by the manufacturer. |
| fdisk     | Menu             | x86 platform only: Runs the fdisk program to create a Solaris fdisk partition.  

The fdisk command cannot be used on disks with an EFI label that are greater than 1 terabyte in size. |
### TABLE 13-1 The Main Menu Item Descriptions for the format Utility (Continued)

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Command or Menu?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>repair</td>
<td>Command</td>
<td>Repairs a specific block on the current disk.</td>
</tr>
<tr>
<td>label</td>
<td>Command</td>
<td>Writes a new label to the current disk.</td>
</tr>
<tr>
<td>analyze</td>
<td>Menu</td>
<td>Runs read, write, and compare tests. For more information, see “analyze Menu” on page 267.</td>
</tr>
<tr>
<td>defect</td>
<td>Menu</td>
<td>Retrieves and displays defect lists. For more information, see “defect Menu” on page 269. This feature does not apply to IDE disks. IDE disks manage defects automatically.</td>
</tr>
</tbody>
</table>
| backup    | Command          | VToc – Searches for backup labels.  
EFI – Not supported. |
| verify    | Command          | Displays 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          | SCSI disks only – Displays the vendor, product name, and revision level of the current drive. |
| volname   | Command          | Labels the disk with a new eight-character volume name that you specify. |
| quit      | Command          | Exits the format menu. |

### partition Menu

The partition menu contents are displayed as follows:

```
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
```
The following table describes the partition menu items.

**TABLE 13-2  Descriptions for partition Menu Items**

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>change 'n' partition</td>
<td>Enables you to specify the following information for the new partition:</td>
</tr>
<tr>
<td></td>
<td>- Identification tag</td>
</tr>
<tr>
<td></td>
<td>- Permission flags</td>
</tr>
<tr>
<td></td>
<td>- Starting cylinder</td>
</tr>
<tr>
<td></td>
<td>- Size</td>
</tr>
<tr>
<td>select</td>
<td>Enables you to choose a predefined partition table.</td>
</tr>
<tr>
<td>modify</td>
<td>Enables you to change all the slices in the partition table. This command</td>
</tr>
<tr>
<td></td>
<td>is preferred over the individual <code>change 'x' partition</code> commands.</td>
</tr>
<tr>
<td>name</td>
<td>Enables you to specify a name for the current partition table.</td>
</tr>
<tr>
<td>print</td>
<td>Displays the current partition table.</td>
</tr>
<tr>
<td>label</td>
<td>Writes the partition map and the label to the current disk.</td>
</tr>
<tr>
<td>quit</td>
<td>Exits the partition menu.</td>
</tr>
</tbody>
</table>

**x86: fdisk Menu**

The `fdisk` menu is available on x86 based systems only and appears similar to the following.

```
format> fdisk
Total disk size is 8924 cylinders
Cylinder size is 16065 (512 byte) blocks

<table>
<thead>
<tr>
<th>Partition</th>
<th>Status</th>
<th>Type</th>
<th>Start</th>
<th>End</th>
<th>Length</th>
<th>%</th>
</tr>
</thead>
</table>
1          |        | EFI  | 0     | 8924| 8925   | 100|
```

SELECT ONE OF THE FOLLOWING:
1. Create a partition
2. Specify the active partition
3. Delete a partition
4. Change between Solaris and Solaris2 Partition IDs
5. Edit/View extended partitions
6. Exit (update disk configuration and exit)
7. Cancel (exit without updating disk configuration)

Enter Selection:
The following table describes the fdisk menu items.

**TABLE 13–3  x86: Descriptions for fdisk Menu Items**

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a partition</td>
<td>Creates an fdisk partition. You must create a separate partition for each OS such as Oracle Solaris or DOS. There is a maximum of four partitions per disk. You are prompted for the size of the fdisk partition as a percentage of the disk.</td>
</tr>
<tr>
<td>Specify the active partition</td>
<td>Enables you to 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.</td>
</tr>
<tr>
<td>Delete a partition</td>
<td>Deletes a previously created partition. This command destroys all the data in the partition.</td>
</tr>
<tr>
<td>Change between Solaris and Solaris2 Partition IDs</td>
<td>Changes partition IDs from 130 (0x82) to 191 (0xbf) and back again.</td>
</tr>
<tr>
<td>Edit/View extended partitions</td>
<td>Manages partition information that is generally used for booting.</td>
</tr>
<tr>
<td>Exit (update disk configuration and exit)</td>
<td>Writes a new version of the partition table and exits the fdisk menu.</td>
</tr>
<tr>
<td>Cancel (exit without updating disk configuration)</td>
<td>Exits the fdisk menu without modifying the partition table.</td>
</tr>
</tbody>
</table>

**analyze Menu**

The analyze menu contents are displayed as follows:

```
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    - exit
```
### TABLE 13-4 Descriptions for analyze Menu Items

<table>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>Reads each sector on the current disk. Repairs defective blocks as a default.</td>
</tr>
<tr>
<td>refresh</td>
<td>Reads then writes data on the current disk without harming the data. Repairs defective blocks as a default.</td>
</tr>
<tr>
<td>test</td>
<td>Writes a set of patterns to the disk without harming the data. Repairs defective blocks as a default.</td>
</tr>
<tr>
<td>write</td>
<td>Writes a set of patterns to the disk then reads back the data on the disk. Destroys existing data on the disk. Repairs defective blocks as a default.</td>
</tr>
<tr>
<td>compare</td>
<td>Writes a set of patterns to the disk, reads back the data, and then compares it to the data in the write buffer. Destroys existing data on the disk. Repairs defective blocks as a default.</td>
</tr>
<tr>
<td>purge</td>
<td>Removes all data from the disk so that the data cannot 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.</td>
</tr>
<tr>
<td>verify</td>
<td>In the first pass, writes unique data to each block on the entire disk. In the next pass, reads and verifies the data. Destroys existing data on the disk. Repairs defective blocks as a default.</td>
</tr>
<tr>
<td>print</td>
<td>Displays the data in the read/write buffer.</td>
</tr>
<tr>
<td>setup</td>
<td>Enables you to specify the following analysis parameters:</td>
</tr>
<tr>
<td></td>
<td>Analyze entire disk? yes</td>
</tr>
<tr>
<td></td>
<td>Starting block number: depends on drive</td>
</tr>
<tr>
<td></td>
<td>Ending block number: depends on drive</td>
</tr>
<tr>
<td></td>
<td>Loop continuously? no</td>
</tr>
<tr>
<td></td>
<td>Number of passes: 2</td>
</tr>
<tr>
<td></td>
<td>Repair defective blocks? yes</td>
</tr>
<tr>
<td></td>
<td>Stop after first error? no</td>
</tr>
<tr>
<td></td>
<td>Use random bit patterns? no</td>
</tr>
<tr>
<td></td>
<td>Number of blocks per transfer: 126 (0/nn)</td>
</tr>
<tr>
<td></td>
<td>Verify media after formatting? yes</td>
</tr>
<tr>
<td></td>
<td>Enable extended messages? no</td>
</tr>
<tr>
<td></td>
<td>Restore defect list? yes</td>
</tr>
<tr>
<td></td>
<td>Restore disk label? yes</td>
</tr>
<tr>
<td>config</td>
<td>Displays the current analysis parameters.</td>
</tr>
<tr>
<td>quit</td>
<td>Exits the analyze menu.</td>
</tr>
</tbody>
</table>
defect Menu

The defect menu contents are displayed as follows:

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>
<thead>
<tr>
<th>Subcommand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary</td>
<td>Reads the manufacturer’s defect list from the disk drive and updates the in-memory defect list.</td>
</tr>
<tr>
<td>grown</td>
<td>Reads the grown defect list and then updates the in-memory defect list. Grown defects are defects that have been detected during analysis.</td>
</tr>
<tr>
<td>both</td>
<td>Reads both the manufacturer’s defect list and the grown defect list. Then, updates the in-memory defect list.</td>
</tr>
<tr>
<td>print</td>
<td>Displays the in-memory defect list.</td>
</tr>
<tr>
<td>dump</td>
<td>Saves the in-memory defect list to a file.</td>
</tr>
<tr>
<td>quit</td>
<td>Exits the defect menu.</td>
</tr>
</tbody>
</table>

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 specify data, see "Getting Help on the format Utility" on page 271.

Specifying Numbers to format Commands

Several places in the format utility require number as input. You must either specify the appropriate data or select a number from a list of choices. In either case, the help facility causes format to display the upper and lower limits of the number expected. Simply enter the appropriate number. 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] 0x1f

**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 type is sufficient to uniquely identify the command desired.

For example, use `p` to access the `partition` menu from the `format` menu. Then, type `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**

At certain points in the `format` utility, 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` subcommand that is 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 typing a question mark (?). The format utility displays a brief description of what type of input is needed.

If you type 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.
Managing File Systems (Overview)

Managing file systems is one of your most important system administration tasks.

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

- “Where to Find File System Management Tasks” on page 273
- “Overview of File Systems” on page 274
- “Default Oracle Solaris File Systems” on page 279
- “Overview of Mounting and Unmounting File Systems” on page 280
- “Determining a File System’s Type” on page 284

Where to Find File System Management Tasks

Use these references to find step-by-step instructions for managing file systems.

<table>
<thead>
<tr>
<th>File System Management Task</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connect and configure new disk devices.</td>
<td>Chapter 7, &quot;Managing Disks (Overview)&quot;</td>
</tr>
<tr>
<td>Create and mount new file systems.</td>
<td>Chapter 15, &quot;Creating and Mounting File Systems (Tasks)&quot;</td>
</tr>
<tr>
<td>Make remote files available to users.</td>
<td>Chapter 5, &quot;Network File System Administration (Tasks),&quot; in System Administration Guide: Network Services</td>
</tr>
</tbody>
</table>
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, beginning 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 know from the context which meaning is intended.

The Oracle Solaris OS 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. The VFS architecture also makes it easier to add new file systems.

Types of Oracle Solaris File Systems

The Oracle Solaris OS 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 284.

Oracle Solaris Disk-Based File Systems

Disk-based file systems are stored on physical media such as hard disks and DVDs. Disk-based file systems can be written in different formats. The available formats are described in the following table.

<table>
<thead>
<tr>
<th>Disk-Based File System</th>
<th>Format Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZFS</td>
<td>ZFS is a disk-based file system. For more information, see the Oracle Solaris ZFS Administration Guide.</td>
</tr>
<tr>
<td>UFS</td>
<td>Legacy UNIX file system (based on the BSD Fat Fast File system that was provided in the 4.3 Tahoe release).</td>
</tr>
</tbody>
</table>
Disk-Based File System  Format Description

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. Oracle Solaris HSFS supports Rock Ridge extensions to ISO 9660. When present on a CD-ROM, these extensions provide all 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.

UDFS  The Universal Disk Format (UDFS) 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:

- ZFS or UFS with hard disk
- HSFS with CD-ROM
- PCFS with USB diskette
- UDF with DVD

However, these associations are not restrictive. For example, DVDs can have ZFS or UFS file systems created on them.

The Universal Disk Format (UDFS) File System

For information about creating a UDFS file system on removable media, see “How to Create a File System on Removable Media” on page 26.

The UDF file system is the industry-standard format for storing information on 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 Oracle Solaris UDF file system works with supported ATAPI and SCSI DVD drives, CD-ROM devices, and disk drives. In addition, the Oracle 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, and 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), with either the sequential disk-at-once recording and incremental recording

The UDF file system requires the following:

- Supported SPARC or x86 platform
- Supported CD-ROM or DVD-ROM device

The Oracle Solaris UDF file system implementation provides the following:

- Support for industry-standard read/write UDF version 1.50
- Fully internationalized file system utilities

**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 the NFS service, you can provide distributed resources (files or directories) by sharing them from a server and mounting them on individual clients. For more information, see “The NFS Environment” on page 282.

**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. Also, some virtual file systems, such as the temporary file system (TMPFS), use the swap space on a disk.

**Temporary File System**

The temporary file system (TMPFS) uses local memory for file system reads and writes. 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. The OS generates a much 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. These files 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 Oracle Solaris OS. You can copy or move files into or out of the /tmp directory, just as you would in a ZFS or 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 15, "Creating and Mounting File Systems (Tasks)." For information about increasing swap space, see Chapter 16, "Configuring Additional Swap Space (Tasks)."

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 the root (/) directory on /tmp/newroot. This loopback mounts make the entire file system hierarchy appear as if 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 15, "Creating and Mounting File Systems (Tasks)."

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 such as ps. Debuggers and other development tools can also access the address space of the processes by using file system calls.

Caution – Do not delete files in the /proc directory. The deletion of processes from the /proc directory does not kill them. /proc files do not use disk space, so there is no 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.
CTFS (the contract filesystem) is the interface for creating, controlling, and observing contracts. A contract enhances the relationship between a process and the system resources it depends on by providing richer error reporting and (optionally) a means of delaying the removal of a resource.

The service management facility (SMF) uses process contracts (a type of contract) to track the processes which compose a service, so that a failure in a part of a multi-process service can be identified as a failure of that service.

**FIFOFS (first-in first-out)**
Named pipe files that give processes common access to data

**FDFS (file descriptors)**
Provides explicit names for opening files by 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

**OBJFS**
The OBJFS (object) file system describes the state of all modules currently loaded by the kernel. This file system is used by debuggers to access information about kernel symbols without having to access the kernel directly.

**SHAREFS**
Provides read-only access to the table of shared file systems for the local system

**SPECFS (special)**
Provides access to character special devices and block devices

**SWAPFS**
Used by the kernel for swapping

### Extended File Attributes

The ZFS, UFS, NFS, and TMPFS file systems have been enhanced to include extended file attributes. Extended file attributes enable application developers to associate specific attributes to a file. For example, a developer of an application used to manage a windowing system 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 namespace. This namespace 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 the file.
$ runat filea ls -l

For more information, see the runat(1) man page.

Many Oracle Solaris file system commands have been modified to support file system attributes by providing an attribute-aware option. Use this option to query, copy, or find file attributes. For more information, see the specific man page for each file system command.

**Swap Space**

The Oracle Solaris OS 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 for 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 16, “Configuring Additional Swap Space (Tasks).”

**Default Oracle Solaris File Systems**

The ZFS file system is hierarchical, starting with the root directory (/) and continuing downwards through a number of directories. The Oracle 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.

For a brief overview of Oracle Solaris file systems and directories, see filesystem(5).

The following table provides a summary of the default Oracle Solaris file systems.

<table>
<thead>
<tr>
<th>Table 14–1  The Default Oracle Solaris File Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>File System or Directory</strong></td>
</tr>
<tr>
<td>root (/)</td>
</tr>
<tr>
<td>/usr</td>
</tr>
<tr>
<td>/export/home or /home</td>
</tr>
</tbody>
</table>
Overview of 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.

Most file systems are automatically mounted by SMF services at system boot time. Generally, you do not need to mount or unmount file systems manually. For more information about mounting different file system types, see “Mounting and Unmounting Oracle Solaris File Systems” on page 286.

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. 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 step-by-step instructions on how to mount file systems, see “Mounting and Unmounting Oracle Solaris File Systems” on page 286.
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 by using the cat or more commands. However, you cannot edit this file. Here is an example of an /etc/mnttab file:

```
$ more /etc/mnttab
rpool/ROOT/zfsBE / zfs dev=3390002 0
/devices /devices devfs dev=85c0000 1337114941
/proc /proc ctsfs dev=8600000 1337114941
/mnttab /etc/mnttab mntfs dev=86c0001 1337114941
swap /system/volatile tmpfs xattr,dev=8700001 1337114941
/objfs /system/object objfs dev=8740001 1337114941
sharefs /etc/dfs/sharetab sharefs dev=8780001 1337114941
/usr/lib/libc/libc_hwcap2.so.1 /lib/libc.so.1 lofs dev=3390002 1337114941
/rpool/ROOT/zfsBE/var /var zfs rw,devices,setuid,nonbmand,exec,
pseudo /dev/fd fd rw,dev=8800001 1337114969
```

The Virtual File System Table

Most file systems are mounted automatically by an SMF service at system boot time.

You might need to edit the /etc/vfstab file to mount legacy or remote file systems or to make changes to the ZFS swap volume. For information about changing a ZFS swap volume, see Chapter 16, “Configuring Additional Swap Space (Tasks).”

To add an entry for mounting a legacy or remote file system, the information you need to specify is as follows:

- The device or the NFS server where the file system resides
- The file system mount point
- File system type
- Whether you want the file system to mount automatically when the system boots (by using the mountall command)
- Any mount options

The following vfstab example is from a system that has a ZFS root file system. In addition, this system is mounting a remote file system, /users/data, from the NFS server, neo.
ZFS file systems are mounted automatically by the SMF service at boot time. You can mount ZFS file systems from the /etc/vfstab by using the legacy mount feature. For more information, see Oracle Solaris ZFS Administration Guide.

For descriptions of each /etc/vfstab field and information on how to edit and use the file, see vfstab(4).

### 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 files.

For more information, see Chapter 4, “Managing Network File Systems (Overview),” in System Administration Guide: Network Services.

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:

- Create a ZFS share by setting the ZFS property. For example:
  
  # cat /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/zvol/dsk/rpool/swap  -  -  swap  -  no  -
  /devices  -  /devices  devfs  -  no  -
  sharefs  -  /etc/dfs/sharetab/sharefs  -  no  -
  ctfs  -  /system/contract ctfs  -  no  -
  objfs  -  /system/object objfs  -  no  -
  swap  -  /tmp  tmpfs  -  yes  -
  neo:/users/data  -  /data  nfs  -  yes  -

- Create a legacy share by using the share command.

  # share -F nfs /ufsfs

NFS Version 4

Oracle's implementation of the NFS version 4 distributed file access protocol is included in the Oracle Solaris release.

NFS version 4 integrates file access, file locking, and mount protocols into a single, unified protocol to ease traversal through a firewall and improve security. The Oracle Solaris implementation of NFS version 4 is fully integrated with Kerberos V5, also known as SEAM, thus providing authentication, integrity, and privacy. NFS version 4 also enables the negotiation of security flavors to be used between the client and the server. With NFS version 4, a server can offer different security flavors for different file systems.

For more information about NFS Version 4 features, see “What's New With the NFS Service” in System Administration Guide: Network Services.

Automounting (autofs)

You can mount NFS file system resources by using a client-side service called automounting (or autofs). The autofs service 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 within that directory. If the resource is not accessed for a certain period of time, it is automatically unmounted.

The autofs service 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 because NFS resources are mounted only 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 NFS file systems on an as-needed basis. By default, the /home file system is mounted by the automount daemon.

With autofs, you can specify multiple servers to provide the same file system. This way, if one of these servers is down, autofs can try to mount the file system from another machine.

For complete information on how to set up and administer autofs, see Chapter 5, “Network File System Administration (Tasks),” in System Administration Guide: Network Services.
Determining a File System's Type

You can determine a file system's type by using one of the following:

- The FS type field in the virtual file system table (the /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

These commands work whether or not the file system is mounted.

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 14-1   Determining a File System's Type**

The following example uses the `fstyp` command to determine the file system type.

```
# fstyp /dev/rdsk/c0t0d0s0
zfs
```

The following example uses the `/etc/vfstab` file to determine the file system type for the `/legacy` file system.

```
$ grep /legacy /etc/vfstab
/dev/dsk/c0t3d0s6  /dev/rdsk/c0t3d0s6  /legacy ufs 2  yes  -
```
Creating Oracle Solaris File Systems

This section provides an overview of creating Oracle Solaris file systems.

Creating ZFS File Systems

A ZFS file system is not directly tied to a specific disk partition. A ZFS file system is contained with a ZFS storage pool that can contain many devices. Either whole disks or disk slices can be added to a ZFS storage pool. Within a pool, you will probably want to create additional file systems. File systems provide points of administration that allow you to manage different sets of data within the same pool.

A ZFS file system is created by using the `zfs create` command. A ZFS file system is automatically mounted when it is created. For more information, see "How to Create a ZFS File System" on page 290.

For more information about creating ZFS storage pools and file systems, see Oracle Solaris ZFS Administration Guide.
Creating a Temporary File System

A temporary file system (TMPFS) uses local memory for file system reads and writes. 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 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 about creating and mounting a TMPFS file system, see “How to Create and Mount a TMPFS File System” on page 292.

For general information, see tmpfs(7FS).

Creating a LOFS File System

An 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 about creating and mounting a LOFS file system, see “How to Create and Mount an LOFS File System” on page 293.

For general information, see lofs(7FS).

Note – 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.

Mounting and Unmounting Oracle Solaris File Systems

ZFS file systems are mounted and unmounted automatically. You can make a legacy UFS 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.

<table>
<thead>
<tr>
<th>Mount Type Needed</th>
<th>Suggested Mount Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local or remote file systems that need to be mounted infrequently.</td>
<td>The mount command that you type manually from the command line.</td>
</tr>
</tbody>
</table>
Mount Type Needed | Suggested Mount Method
--- | ---
Local UFS file systems that need to be mounted frequently. Local ZFS file systems are automatically mounted by an SMF service. | The `/etc/vfstab` file, which mounts the file system automatically when the system is booted in multiuser state.
Remote legacy UFS file systems, such as home directories, that need to be mounted frequently. | ■ The `/etc/vfstab` file, which automatically mounts the file system when the system is booted in multiuser state.
■ `autofs`, which automatically mounts the file system when you access it or unmounts the file system when you change to another directory.

For more information on mounting removable media, see "Managing Removable Media (Overview)" on page 19.

You can determine which file systems are already mounted by using the `mount` command:

```bash
$ mount [ -v ]
```

The `-v` displays the list of mounted file systems in verbose mode.

**EXAMPLE 15–1 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.

```bash
$ mount
/ on rpool/ROOT/zfsBE read/write/setuid/devices/rstchown/dev=3390002 on Tue ...
devices on /dev read/write/setuid/devices/rstchown/dev=8580000 on Tue May 15 ...
dev on /dev read/write/setuid/devices/rstchown/dev=85c0000 on Tue May 15 14:49:01 2012
/system/contract on cfts read/write/setuid/devices/rstchown/dev=8600001 on Tue May 15 ...
/proc on proc read/write/setuid/devices/rstchown/dev=3390002 on Tue May 15 14:49:01 2012
/etc/mnttab on mnttab read/write/setuid/devices/rstchown/dev=86c0001 on Tue May 15 14:49:01 ...
/system/volatile on swap read/write/setuid/devices/rstchown/xattr/dev=8700001 on Tue May 15 ...
/system/object on objfs read/write/setuid/devices/rstchown/dev=8740001 on Tue May 15 ...
/etc/dfs/sharetab on sharefs read/write/setuid/devices/rstchown/xattr/dev=8780001 on Tue May 15 ...
/lib/libc.so.1 on /usr/lib/libc/libc_hwcap2.so.1 read/write/setuid/devices/rstchown/dev ...
/dev/fd on fd read/write/setuid/devices/rstchown/dev=8880001 on Tue May 15 14:49:29 2012
/var on rpool/ROOT/zfsBE/var read/write/setuid/devices/rstchown/nonbmand/exec/xattr/atime/ ...
/tmp on swap read/write/setuid/devices/rstchown/xattr/dev=8700002 on Tue May 15 14:49:29 2012
/var/share on rpool/VARSHARE read/write/setuid/devices/rstchown/nonbmand/exec/xattr/atime/ ...
/home/rimmer on pluto:/export/home/rimmer remote/read/write/setuid/exec/xattr/...
```

This example shows how to use the `zfs mount` command to display information about ZFS file systems that are currently mounted.

```bash
$ zfs mount
rpool/ROOT/zfsBE /
rpool/ROOT/zfsBE /var
rpool/VARSHARE /var/share
```
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>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>device to mount</td>
<td>This field identifies one of the following:</td>
</tr>
<tr>
<td></td>
<td>▪ The block device name for a local legacy UFS filesystem (for example, /dev/dsk/c8t1d0s7).</td>
</tr>
<tr>
<td></td>
<td>▪ The resource name for a remote file system (for example, myserver:/export/home).</td>
</tr>
<tr>
<td></td>
<td>After you add an entry for a remote system resource, be sure that the following service is enabled.</td>
</tr>
<tr>
<td></td>
<td># svc -a</td>
</tr>
<tr>
<td></td>
<td>disabled May 14 svc:/network/nfs/client:default</td>
</tr>
<tr>
<td></td>
<td># svcadm enable svc:/network/nfs/client:default</td>
</tr>
<tr>
<td></td>
<td>For more information about NFS, see System Administration Guide: Network Services.</td>
</tr>
<tr>
<td></td>
<td>▪ The swap volume (for example, /dev/zvol/dsk/rpool/swap).</td>
</tr>
<tr>
<td></td>
<td>▪ A directory for a virtual file system.</td>
</tr>
<tr>
<td>device to fsck</td>
<td>The raw (character) device name that corresponds to the legacy UFS file system identified by the device to mount field (for example, /dev/dsk/c8t1d0s7). 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.</td>
</tr>
<tr>
<td>mount point</td>
<td>Identifies where to mount the legacy or remote file system (for example, /data).</td>
</tr>
<tr>
<td>FS type</td>
<td>Identifies the type of file system.</td>
</tr>
</tbody>
</table>
### FIELD DESCRIPTIONS FOR THE /etc/vfstab FILE (CONTINUED)

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>fsck pass</strong></td>
<td>The pass number used by the <code>fsck</code> command to decide whether to check a legacy UFS file system. When the field contains a dash (<code>-</code>), the file system is not checked. Currently, the <code>fsck pass</code> value in the <code>/etc/vfstab</code> file is ignored during the boot process. When the field contains a zero, legacy UFS file systems are not checked. When the field contains a value greater than zero, the UFS file system is always checked. All legacy UFS file systems with a value of 1 in this field are checked one at a time in the order they appear in the <code>vfstab</code> file. When the <code>fsck</code> command is run on multiple UFS file systems that have <code>fsck pass</code> values greater than 1 and the <code>-p</code> option is used, the <code>fsck</code> 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.</td>
</tr>
<tr>
<td><strong>mount at boot</strong></td>
<td>Set to <code>yes</code> or <code>no</code> for whether the file system should be automatically mounted by the <code>mountall</code> command when the system is booted. Note that this field has nothing to do with <code>autofs</code>. This field should always be set to <code>no</code> for virtual file systems such as <code>/proc</code> and <code>/dev/fd</code>.</td>
</tr>
<tr>
<td><strong>mount options</strong></td>
<td>A list of comma-separated options (with no spaces) that are used for mounting the file system. Use a dash (<code>-</code>) to indicate no options. For more information, see <code>vfstab(4)</code>.</td>
</tr>
</tbody>
</table>

*Note* – You must have an entry in each field in the `/etc/vfstab` file. If there is no value for a field, be sure to specify a dash (`-`). Otherwise, the system might not boot successfully. Similarly, white space should not be used as a field value.

## Prerequisites for Unmounting Oracle Solaris File Systems

The prerequisites for unmounting file systems include the following:

- You must be an administrator.
- 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 the file system 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 That Are Accessing a File System” on page 298.

Notify users if you need to unmount a file system that they are using.

- Unsharing the file system.
  - For example:
    ```
    # zfs set share.nfs=off tank/fs
    ```
  - Use the legacy unshare method. For information, see `unshare(1M)`.

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
```

### Creating and Mounting Oracle Solaris File Systems

This section provides examples of creating and mounting Oracle Solaris file systems.

#### How to Create an ZFS File System

1. **Become superuser.**

2. **Create a ZFS storage pool.**

   The following example illustrates how to create a simple mirrored storage pool named `tank` and a ZFS file system named `tank` in one command. Assume that the whole disks `/dev/dsk/c1t0d0` and `/dev/dsk/c2t0d0` are available for use.

   ```
   # zpool create tank mirror c1t0d0 c2t0d0
   ```

3. **Create a ZFS file system.**

   ```
   # zfs create tank/fs
   ```

   The new ZFS file system, `tank/fs`, can use as much of the disk space as needed, and is automatically mounted at `/tank/fs`.

4. **Confirm that the file system is created.**

   ```
   # zfs list -r tank
   NAME USED AVAIL REFER MOUNTPOINT
   tank  117K 268G 21K /tank
   tank/fs  21K 268G 21K /tank/fs
   ```
How to Create and Mount a Legacy UFS File System

Before You Begin

Ensure that you have met the following prerequisites:

- The disk must be formatted and divided into slices.
- If you are recreating an existing legacy UFS file system, unmount it.
- 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 9, “Administering Disks (Tasks).”

For information on formatting disks and dividing disks into slices, see Chapter 7, “Managing Disks (Overview).”

1 Become superuser.

2 Create a legacy UFS file system.

   # newfs [-N] [-b size] [-i bytes] /dev/rdsk/device-name

   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 legacy UFS file system, check the new file system.

   # fsck /dev/rdsk/device-name

   where 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 fsck(1M).

4 Mount the legacy UFS file system.

   # mkdir /directory-name

   # mount /dev/dsk/device-name /directory-name

Example 15–2 Creating and Mounting a Legacy UFS File System

The following example shows how to create and mount a UFS file system /dev/rdsk/c0t1d0s0 on /legacy.

   # newfs /dev/rdsk/c0t1d0s0

   newfs: construct a new file system /dev/rdsk/c0t1d0s0: (y/n)? y

   /dev/rdsk/c0t1d0s0: 286722656 sectors in 46668 cylinders of 48 tracks, 128 sectors
Creating and Mounting Oracle Solaris File Systems

14001.3MB in 2917 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
32, 98464, 196896, 295328, 393760, 492192, 590624, 689056, 787488, 885920,
Initializing cylinder groups:
..........................................................
super-block backups for last 10 cylinder groups at:
285773216, 285871648, 285970080, 286068512, 286166944, 286265376, 286363808,
286462240, 286560672, 286659104
# fsck /dev/rdsk/c0t1d0s0
# mkdir /legacy
# mount /dev/dsk/c0t1d0s0 /legacy

More Information

After You Create a Legacy UFS File System ...

To mount the legacy UFS file system automatically at boot time, go to “How to Add an Entry to the /etc/vfstab File” on page 294.

▼ How to Create and Mount a TMPFS File System

1 Become superuser.

2 Create the directory that you want to mount as the TMPFS file system, if necessary.
   # mkdir /mount-point
   where 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 MB.
   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 at boot time, see Example 15–4.

4 Verify that the TMPFS file system has been created.
   # mount -v

Example 15–3 Creating and Mounting 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 MB.

   # mkdir /export/reports
   # chmod 777 /export/reports
   # mount -F tmpfs -o size=50m swap /export/reports
   # mount -v

292 System Administration Guide: Devices and File Systems • June 2013
You can set up the system to automatically mount a TMPFS file system at boot time 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 at boot time. Because 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 -
```

### How to Create and Mount an LOFS File System

1. **Become superuser.**

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 15–5 Creating and Mounting an LOFS File System

The following example shows 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
```
Mounting an LOFS FileSystem at Boot Time

You can set up the system to automatically mount an LOFS file system at boot time 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 -
```

Ensure that 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.

How to Add an Entry to the `/etc/vfstab` File

Use this procedure to mount non-ZFS file systems at boot time unless legacy mount behavior is needed for some ZFS file systems. For more information about mounting ZFS file systems, see Oracle Solaris ZFS Administration Guide.

1. Become superuser.

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. Ensure that you do the following:

   a. Separate each field with white space (a space or a tab).
   
   b. Specify a dash (`-`) if a field has no contents.
   
   c. Save the changes.

   **Note** – Because 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.
Adding an Entry to the /etc/vfstab File

The following example shows how to mount the disk slice /dev/dsk/c0t3d0s7 as a legacy UFS file system to the mount point /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. Neither 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
```

How to Mount a File System (/etc/vfstab File)

1. Become superuser.

2. Mount a file system listed in the /etc/vfstab file.

   ```
   # mount /mount-point
   
   where /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 15–8 Mounting a File System (/etc/vfstab File)

The following example shows how to mount the local /legacy file system that is listed in the /etc/vfstab file.

```
# mount /legacy
```

Example 15–9 Mounting All File Systems (/etc/vfstab File)

The following example shows the messages that are displayed when you use the mountall command and the file systems are already mounted.

```
# mountall
mount: /tmp is already mounted or swap is busy
```
The following example shows how to mount all the local systems that are listed in the /etc/vfstab file.

```
# mountall -l
```

The following example shows how to mount all available ZFS file systems.

```
# zfs mount -a
```

The following example shows how to mount all the remote file systems that are listed in the /etc/vfstab file.

```
# mountall -r
```

▶ **How to Mount an NFS File System (mount Command)**

1. Become superuser.

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. Ensure that 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: Network Services*.

4. Mount the NFS file system.
   ```
   # mount -F nfs [-o mount-options] server:/directory /mount-point
   ```

**Example 15–10 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.

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.

Example 15–11 x86: 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 read-only the first logical drive in the extended DOS slice on the `/mnt` directory.

```
# mount -F pcfs -o ro /dev/dsk/c0t0d0p0:2 /mnt
```
How to Stop All Processes That Are Accessing a File System

1. Become superuser.

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 no processes are accessing the file system.
   
   ```
   # fuser -c /mount-point
   ```

Example 15–12 Stopping All Processes That Are 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.

1. Ensure that you have met the prerequisites listed in “Prerequisites for Unmounting Oracle Solaris File Systems” on page 289.
2 Unmount the file system.

    # umount /mount-point

where /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 an LOFS file system

Example 15–13 Unmounting a File System

The following example shows how to unmount a legacy UFS file system:

    # umount /legacy

The following example shows how to forcibly unmount the UFS /legacy file system:

    # umount -f /legacy
    #

The following example shows to unmount all ZFS file systems:

    # zfs umount -a

All file systems are unmounted, except for those file systems that are busy.
Configuring Additional Swap Space (Tasks)

This chapter provides guidelines and step-by-step instructions for configuring additional swap space for a ZFS root file system after the Oracle Solaris OS is installed.

This is a list of the information in this chapter:

- “About Swap Space” on page 301
- “How Do I Know If I Need More Swap Space?” on page 303
- "How Swap Space Is Allocated” on page 304
- “Planning for Swap Space” on page 305
- “Monitoring Swap Resources” on page 306
- “Adding or Changing Swap Space in an Oracle Solaris ZFS Root Environment” on page 308

About Swap Space

You should understand the features of the swap mechanism in Oracle Solaris to determine the following:

- Swap space requirements
- The relationship between swap space and the TMPFS file system
- How to recover from error messages related to swap space

Swap Space and Virtual Memory

Oracle Solaris OS software and application software can use some disk space for temporary storage rather than for file systems. The reserved area of the disk is called swap space. Swap space is used as virtual memory storage areas when the system does not have enough physical memory to handle current processes. In a ZFS root file system, the disk space reserved for swap is a ZFS volume.
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 no identity is assigned to the disk space that is backing the memory.

The Oracle Solaris OS 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 Oracle 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.

Do 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 and Dynamic Reconfiguration

A good practice is to allocate enough swap space to support a failing CPU or system board during dynamic reconfiguration. Otherwise, a CPU or system board failure might result in your host or domain rebooting with less memory.

Without having this additional swap space available, one or more of your applications might fail to start due to insufficient memory. This problem would require manual intervention either to add additional swap space or to reconfigure the memory usage of these applications.

If you have allocated additional swap space to handle a potential loss of memory on reboot, all of your intensive applications might start as usual. This means the system will be available to the users, perhaps possibly slower due to some additional swapping.

For more information, see your hardware dynamic reconfiguration guide.

Configuring Swap Space in a SAN Environment

Review the following points to determine whether you might configure swap space on a network-connected disk, such as in a SAN environment:

- Diagnosing swap space issues on a locally-attached disk is easier than diagnosing swap space issues on a network-connected disk.
- The performance of swap space over a SAN should be comparable to swap space configured on a locally-attached disk.
- Adding more memory to a system with performance issues, after analyzing performance data, might resolve a swap over SAN performance problem better than moving the swap to a locally-attached disk.

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 swaps 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.
Swap-Related Error Messages

These messages indicate that an application was trying to get more anonymous memory. However, no swap space was 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 a file was being written. 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 that 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 Oracle Solaris installation process. In a ZFS root environment, the default swap size is generally in the 512 MB to 2 GB range.

For general guidelines on allocating swap space, see "Planning for Swap Space" on page 305.

Swap Areas and the /etc/vfstab File

After the system is installed, swap areas 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 on system with a UFS root file system or a swap volume path name on a system with a ZFS root file system
- File system type of the swap slice or swap file
The filesystem that contains a swap file must be mounted before the swap file is activated. So, in
the /etc/vfstab file, ensure 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 MB of swap space.

Consult your application vendors for swap space requirements for their applications.

If you are unable to determine swap space requirements from your application vendors, use the
following general guidelines based on your system type to allocate swap space.

**Note** – Crash dump content is compressed so the dump device does not have to be the same size
as physical memory. By default, the dump content value is set to kernel pages. However, if the
dump content value is set to dump all memory pages, then consider increasing the dump size to
half the size of physical memory or more.

**TABLE 16–1** Swap and Dump Volume Sizes for ZFS File Systems

<table>
<thead>
<tr>
<th>System Type</th>
<th>Swap Volume Size</th>
<th>Dump Volume Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>System with about 4 GB of physical memory</td>
<td>1 GB</td>
<td>2 GB</td>
</tr>
<tr>
<td>Mid-range server with about 8 GB of physical memory</td>
<td>2 GB</td>
<td>4 GB</td>
</tr>
<tr>
<td>High-end server with about 16 to 128 GB of physical memory</td>
<td>4 GB</td>
<td>8–64 GB</td>
</tr>
<tr>
<td>High-end server with more than 128 GB of physical memory</td>
<td>1/4 of physical memory size</td>
<td>1/2 of physical memory size</td>
</tr>
</tbody>
</table>
Note – A busy system with many active ZFS file systems might use 1/2 to 3/4 the size of physical memory for the size of the dump device.

On a system with a ZFS root file system, if you attempt to designate a dump device that is too small to hold a system crash system with the dumpadm -d command, you will see a message similar to the following:

dumpadm: dump device /dev/zvol/dsk/rpool/dump is too small to hold a system dump dump size 43467329536 bytes, device size 42949672960 bytes

Allocating Swap Space for ZFS-Based Systems

During an initial installation of a ZFS root file system, a swap area is automatically created on a ZFS volume in the ZFS root pool.

In a ZFS root pool, swap devices are not pre-allocated to fixed-size slices, so it is fairly easy to modify the swap size later.

After you assess the swap requirements of your applications, you can use the default swap size or adjust the swap volume size during an initial installation or after the installation, if necessary.

During an initial installation, the default dump volume size is calculated by the kernel based on dumpadm information and the size of physical memory.

In a ZFS environment, file systems consume space from the pool so the/var/crash directory consumes what it needs depending on how many crash dumps are saved.

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. For example:

```
# swap -l
swapfile    dev   swaplo blocks   free
/dev/dsk/c0t0d0s1  136,1  16 1638608 1600528
```

On a system with a ZFS root file system, the swap -l command identifies similar output except that it identifies the ZFS volume path name. For example:

```
# swap -l
swapfile    dev   swaplo blocks   free
/dev/zvol/dsk/rpool/swap 256,1  16 1058800 1058800
```
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 determine how much swap space is available. When the performance of a system slows down, check the amount of available swap space to determine 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. The `swap -s` command displays swap space in 1024-byte blocks. If you add up the blocks from `swap -l` and convert them to KB, the result is less than `used + available` (in the `swap -s` output). The reason is that `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>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bytes allocated</strong></td>
<td>The total amount of swap space in 1024-byte blocks that is currently allocated as backing store (disk-backed swap space).</td>
</tr>
<tr>
<td><strong>reserved</strong></td>
<td>The total amount of swap space in 1024-byte blocks that is not currently allocated, but claimed by memory for possible future use.</td>
</tr>
<tr>
<td><strong>used</strong></td>
<td>The total amount of swap space in 1024-byte blocks that is either allocated or reserved.</td>
</tr>
<tr>
<td><strong>available</strong></td>
<td>The total amount of swap space in 1024-byte blocks that is currently available for future reservation and allocation.</td>
</tr>
</tbody>
</table>
Adding or Changing Swap Space in an Oracle Solaris ZFS Root Environment

The following section describes how to add or change swap space in a ZFS root environment. See the previous sections to determine if your system or applications need additional swap space.

For information about customizing ZFS swap or dump devices, see “Customizing ZFS Swap and Dump Volumes” in Oracle Solaris ZFS Administration Guide.

How to Add Swap Space in an Oracle Solaris ZFS Root Environment

1. Become superuser.

2. Identify the current swap volume.

   A swap volume cannot be removed if it is in use. You can tell if the current swap volume is in use by comparing the blocks identified in the blocks column and blocks identified in the free column. If the blocks in the two columns are equal, the swap area is not busy. For example:

   ```
   # swap -l
   swapfile dev swaplo blocks free
   /dev/zvol/dsk/rpool/swap 256,1 16 1058800 1058800
   ```

   3. Select one of the following to resize the swap volume.

   a. If the current swap area is not in use, you can resize the size of the current swap volume, but you must reboot the system to see the increased swap space.

   For example:

   ```
   # zfs get volsizex rpool/swap
   NAME PROPERTY VALUE SOURCE
   rpool/swap volsizex 517M -
   # zfs set volsizex=2G rpool/swap
   # zfs get volsizex rpool/swap
   NAME PROPERTY VALUE SOURCE
   rpool/swap volsizex 2G -
   # init 6
   ```

   b. If the system cannot be rebooted, add another swap volume to increase your total swap space.

   For example:

   ```
   # zfs create -V 2G rpool/swap2
   ```
Then, activate the second swap volume.

```
# swap -a /dev/zvol/dsk/rpool/swap2
# swap -l
```

```
swapfile         dev swaplo  blocks  free
/dev/zvol/dsk/rpool/swap  256,1  16  1058800  1058800
/dev/zvol/dsk/rpool/swap2  256,3  16  4194288  4194288
```

4 If necessary, add an entry for the second swap volume in the `/etc/vfstab` file.

For example:

```
/dev/zvol/dsk/rpool/swap2  -  -  swap  -  no  -
```
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 Check the UFS root (/), /usr, or /var File Systems From an Alternate Boot Device” on page 326
- “How to Check Other UFS File Systems (Not root (/), /usr, or /var)” on page 328
- “How to Preen a UFS File System” on page 330
- “How to Restore a Bad Superblock (Solaris 8, 9, and 10 Releases)” on page 335
- “How to Restore a Bad UFS Superblock (Solaris 10 6/06 Release)” on page 332

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

- “UFS File System Consistency” on page 312
- “How the fsck File System State Is Recorded” on page 318
- “What the fsck Command Checks and Tries to Repair” on page 319
- “Interactively Checking and Repairing a UFS File System” on page 325
- “Restoring a Bad UFS Superblock” on page 332
- “Syntax and Options for the fsck Command” on page 337

For information about fsck error messages, see Chapter 20, “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 18, “UFS File System (Reference).”
UFS 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 from the following:

- 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). Often, 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. An 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 340.

Enhancements to UFS File System Utilities (`fsck`, `mkfs`, and `newfs`)

The file system check utility, `fsck`, has been enhanced to include features from the FreeBSD 4.9 version of the `fsck` program, as well as other enhancements.

The `fsck` utility in this Oracle Solaris release includes the following improvements:

- Checks and repairs file systems more thoroughly and provides improved error messages. For example, in some scenarios, `fsck` determines what structures are missing and replaces them appropriately.
- Automatically searches for backup superblocks.
- Reports when `fsck` needs to be rerun.
- When clearing directories, `fsck` now attempts to recover directory contents immediately and therefore, reduces the time spent rerunning this utility.
- If `fsck` finds duplicate blocks, and not all files that reference the duplicate blocks were cleared, `fsck` reports the inode numbers at the end of the `fsck` run. Then, you can use the `find` command to review the inodes that are damaged.
Improved error messages regarding the status of extended attributes and other special files, such as device files and ACL entries, are included.

Includes a `-v` option to enable more verbose messages.

In addition, the `newfs` and `mkfs` commands have been updated to include new options for displaying a file system’s superblock information in text or dumping the superblock information in binary format.

```
newfs [ -S or -B ] /dev/rdsk/...
-S Displays the file system’s superblock in text
-B Dumps the file system’s superblock in binary
```

```
mkfs [ -o calcsb or -o calcbinsb ] /dev/rdsk/... size
-o calcsb Displays the file system’s superblock in text
-o calcbinsb Dumps the file system’s superblock in binary
```

The `fsck` utility uses this superblock information to search for backup superblocks.

The following sections describe specific `fsck` enhancements and their corresponding error messages. For step-by-step instructions on using the `fsck` utility to repair a damaged superblock, see "How to Restore a Bad UFS Superblock (Solaris 10 6/06 Release)" on page 332.

**Automatic Search for Backup Superblocks**

The following `fsck` error message examples illustrate the automatic backup superblock discovery feature.

---

**Caution** – If a file system has a damaged superblock and it was created with `newfs` or `mkfs` customized parameters, such as `ntrack` or `nsect`, using `fsck`’s automatically discovered superblock for the repair process could damage your file system.

In the case of a file system that was created with customized parameters and it has a bad superblock, `fsck` provides the prompt to cancel the `fsck` session:

**CANCEL FILESYSTEM CHECK?**

- If the file system was created with the `newfs` command and `fsck` responds that just the primary superblocks are corrupted, then consider letting `fsck` restore the superblock.

```
# fsck /dev/dsk/c1t2d0s0
** /dev/rdsk/c1t2d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED
LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? no
```
LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? yes

FOUND ALTERNATE SUPERBLOCK 32 WITH NEWFS

USE ALTERNATE SUPERBLOCK? yes

FOUND ALTERNATE SUPERBLOCK AT 32 USING NEWFS
If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? no

** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups
CORRECT GLOBAL SUMMARY
SALVAGE? y

UPDATE STANDARD SUPERBLOCK? y

81 files, 3609 used, 244678 free (6 frags, 30584 blocks, 0.0%
fragmentation)

***** FILE SYSTEM WAS MODIFIED *****

- If the file system was created with the mkfs command and fsck responds that just the
primary superblocks are corrupted, then consider letting fsck restore the superblock.

# fsck /dev/dsk/c1t2d0s0
** /dev/dsk/c1t2d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes

FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS

USE ALTERNATE SUPERBLOCK? yes

FOUND ALTERNATE SUPERBLOCK AT 32 USING MKFS
If filesystem was created with manually-specified geometry, using
auto-discovered superblock may result in irrecoverable damage to
filesystem and user data.

CANCEL FILESYSTEM CHECK? no

** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups
CORRECT GLOBAL SUMMARY
SALVAGE? y

UPDATE STANDARD SUPERBLOCK? y

81 files, 360 used, 243605 free (117 frags, 30436 blocks, 0.0% fragmentation)

***** FILE SYSTEM WAS MODIFIED *****

The following example illustrates what would happen if you specified fsck’s -y option in a damaged superblock scenario. You are automatically dropped out of the fsck session. A message is displayed to rerun it with the alternate superblock.

# fsck -y /dev/dsk/c1t2d0s0
#
** /dev/dsk/c1t2d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes

LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? yes

SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

USE GENERIC SUPERBLOCK FROM MKFS? yes

CALCULATED GENERIC SUPERBLOCK WITH MKFS
If filesystem was created with manually-specified geometry, using auto-discovered superblock may result in irrecoverable damage to filesystem and user data.

CANCEL FILESYSTEM CHECK? yes

Please verify that the indicated block contains a proper superblock for the filesystem (see fsdb(1M)).

FSCK was running in YES mode. If you wish to run in that mode using the alternate superblock, run ‘fsck -y -o b=453920 /dev/dsk/c1t2d0s0’.

The following fsck error message scenario illustrates the new prompts for the backup superblock, but the fsck run is not canceled, in this example. Canceling the fsck session would be an appropriate response if this file system was created with customized parameters or if there is some other concern about running fsck on this file system.

The various superblock error conditions are provided in italics as follows:

# fsck /dev/dsk/c0t1d0s0
** /dev/dsk/c0t1d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED
BAD SUPERBLOCK AT BLOCK 16: NUMBER OF DATA BLOCKS OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: INODES PER GROUP OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: MAGIC NUMBER WRONG
BAD SUPERBLOCK AT BLOCK 16: BAD VALUES IN SUPER BLOCK
BAD SUPERBLOCK AT BLOCK 16: NCG OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: CPG OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: NCYL IS INCONSISTENT WITH NCG*CPG
BAD SUPERBLOCK AT BLOCK 16: SIZE OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: NUMBER OF DIRECTORIES OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: ROTATIONAL POSITION TABLE SIZE OUT OF RANGE
BAD SUPERBLOCK AT BLOCK 16: SIZE OF CYLINDER GROUP SUMMARY AREA WRONG
BAD SUPERBLOCK AT BLOCK 16: INOPB NONSSENSICAL RELATIVE TO BSIZE

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes

FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS

USE ALTERNATE SUPERBLOCK? yes

FOUND ALTERNATE SUPERBLOCK AT 32 USING MKFS

If filesystem was created with manually-specified geometry, using auto-discovered superblock may result in irrecoverable damage to filesystem and user data.

CANCEL FILESYSTEM CHECK? no

** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2a - Check Duplicated Names
** Phase 2b - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups

SALVAGE? yes

UPDATE STANDARD SUPERBLOCK? yes

82 files, 3649 used, 244894 free (6 frags, 30611 blocks, 0.0% fragmentation)

***** FILE SYSTEM WAS MODIFIED *****

**fsck** Reports When it Needs to be Rerun

Better reporting by **fsck** about when it needs to be rerun should alleviate the time and necessity of running it multiple times, which can be particularly time consuming on large file systems.

The following new messages prompt you to rerun the **fsck** utility at the end of an error scenario:

***** PLEASE RERUN FSCK *****
Or:

Please rerun fsck(1M) to correct this.

These new prompts resolve the previous difficulty in determining whether fsck should be rerun or not.

Unless you are prompted to rerun fsck as in the above messages, there is no need to run fsck, even after you see the following message:

***** FILE SYSTEM WAS MODIFIED *****

However, it doesn't harm the file system to rerun fsck after this message. This message is just informational about fsck's corrective actions.

**New fsck Messages Regarding Extended Attributes**

New fsck messages are included that report on and repair files with extended attributes. For example:

BAD ATTRIBUTE REFERENCE TO I=1 FROM I=96

Attribute directory I=97 not attached to file I=96
I=96  OWNER=root  MODE=40755
SIZE=512  MTIME=Jul 12 10:55:10 2010
DIR= <xattr>

FIX?  yes

ZERO LENGTH ATTR DIR I=12  OWNER=root  MODE=160755
SIZE=0  MTIME=Jul 12 10:56:10 2010
CLEAR?  yes

File should BE marked as extended attribute
I=22  OWNER=root  MODE=100644
SIZE=0  MTIME=Jul 12 10:57:10 2010
FILE= <xattr>

FIX?  yes

UNREF ATTR DIR I=106  OWNER=root  MODE=160755
SIZE=512  MTIME=Jul 12 10:58:10 2010
RECONNECT?  yes

File I=107 should NOT be marked as extended attribute
I=107  OWNER=root  MODE=100644
SIZE=0  MTIME=Jul 12 10:59:10 2010
FILE=?/attfsdir-7-att

FIX?  yes
DIR I=106 CONNECTED.
Better Handling of Duplicate Blocks or Fragments

The fsck error messages now reports information about blocks, fragments, or a LFNs, which are the logical fragment numbers from the start of the file. For example, you might see output similar to the following:

** Phase 1 - Check Blocks and Sizes
FRAGMENT 784 DUP I=38 LFN 0
FRAGMENT 785 DUP I=38 LFN 1
FRAGMENT 786 DUP I=38 LFN 2
.
.
.

fsck processes objects as fragments, but in previous Solaris releases, only reported object information as blocks. It now correctly reports as fragments.

If fsck finds error conditions that involve duplicate blocks or fragments, fsck offers to display the uncleared files at end of the fsck output. For example, you might see output similar to the following:

LIST REMAINING DUPS? yes

Some blocks that were found to be in multiple files are still assigned to file(s).
Fragments sorted by inode and logical offsets:
Inode 38:
 Logical Offset 0x00000000 Physical Fragment 784
 Logical Offset 0x00000800 Physical Fragment 786
 Logical Offset 0x00010000 Physical Fragment 788
 Logical Offset 0x00018000 Physical Fragment 790

Then, you can use the find -i inode-number command to identify the name of inode 38, in this example.

How the UFS 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 “Superblock” on page 340.

The possible state flag values are described in the following table.
### TABLE 17–1 Values of UFS File System State Flags

<table>
<thead>
<tr>
<th>State Flag Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSACTIVE</td>
<td>Indicates a mounted file system that has modified data in memory. A mounted file system with this state flag indicates that user data or metadata would be lost if power to the system is interrupted.</td>
</tr>
<tr>
<td>FSBAD</td>
<td>Indicates that the file system contains inconsistent file system data.</td>
</tr>
<tr>
<td>FSCLEAN</td>
<td>Indicates an undamaged, cleanly unmounted file system.</td>
</tr>
<tr>
<td>FSLOG</td>
<td>Indicates that the file system has logging enabled. A file system with this flag set is either mounted or unmounted. If a file system has logging enabled, the only flags that it can have are FSLOG or FSBAD. A file system that has logging disable can have FSACTIVE, FSSTABLE, or FSCLEAN.</td>
</tr>
<tr>
<td>FSSTABLE</td>
<td>Indicates an idle mounted file system. A mounted file system with this state flag indicates that neither user data nor metadata would be lost if power to the system is interrupted.</td>
</tr>
</tbody>
</table>

## 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 UFS File System 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 `fsfs flush` 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 or problems with the disk or controller firmware. Blocks can become damaged on a disk drive at any time. Or, a disk controller can stop functioning correctly.

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 "Structure of Cylinder Groups for UFS File Systems" on page 339.

Superblock Checks

The superblock stores summary information, which is the most commonly corrupted component in a UFS file system. Each change to 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. 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 "Structure of Cylinder Groups for UFS File Systems" on page 339.
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 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

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:
When the file system is created, a fixed number of inodes are set aside. However, these inodes 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 one 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. However, 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 duplicate blocks are found, the fsck command makes a second pass of the inode list to find the other inode that claims each duplicate block. The fsck command cannot determine with certainty which inode is in error. So, the fsck command prompts you to choose which inode should be kept and which inode should be cleared. Note that a large number of duplicate blocks in an inode might be caused by an indirect block not being written to the file system.
Bad Block Number Checks
The fsck command checks each block number claimed by an inode to determine whether its value is higher than the value 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 uses the number of characters shown in the size field to calculate how many blocks should be associated with the inode, and then compares that number 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 direct 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 shut down 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 that is not linked to the file system is found, 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. However, 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.
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 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder 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 343.

---

### 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 inconsistencies while in use

When an in-use file system develops inconsistencies, error messages might be displayed in the console window or the system messages file. Or, the system might crash. For example, the system messages file, `/var/adm/messages`, might include messages similar to the following:

```
Sep 5 13:42:40 hostname ufs: [ID 879645 kern.notice] NOTICE: /: unexpected free inode 630916, run fsck(1M)
```

`hostname` is the system reporting the error.
Before using the `fsck` command, you might want to refer to these references for information on resolving `fsck` error messages:

- “Syntax and Options for the `fsck` Command” on page 337
- Chapter 20, “Resolving UFS File System Inconsistencies (Tasks),” in System Administration Guide: Advanced Administration

Keep the following points in mind when running the `fsck` command to check UFS file systems:

- A file system *should* be inactive when you use `fsck` to check a file system. File system changes waiting to be flushed to disk or file system changes that occur during the `fsck` checking process can be interpreted as file system corruption. These issues may not be a reliable indication of a problem.

- A file system *must* be inactive when you use `fsck` to repair that file system. File system changes waiting to be flushed to disk or file system changes that occur during the `fsck` repairing process might cause the file system to become corrupted. Or, they might cause the system to crash.

- Unmount a file system before you use `fsck` on that file system. Doing so ensures that the file system data structures are consistent as possible. The only exceptions are for the active root (`/`), `/usr`, and `/var` file systems because they must be mounted to run `fsck`.

- If you need to repair the root (`/`), `/usr`, and `/var` file systems, boot the system from an alternate device, if possible, so that these file systems are unmounted and inactive. For step-by-step instructions on running `fsck` on the root (`/`), `/usr`, or `/var` file systems, see “How to Check the UFS root (`/`), `/usr`, or `/var` File Systems From an Alternate Boot Device” on page 326.

How to Check the UFS root (`/`), `/usr`, or `/var` File Systems From an Alternate Boot Device

There is no need to rerun `fsck` if you see the following message:

***** FILE SYSTEM WAS MODIFIED *****

However, it doesn’t harm the file system to rerun `fsck` after this message. This message is just informational about `fsck`’s corrective actions.

This procedure assumes that a local Solaris DVD or network boot server is available so that you can boot the system from an alternate device.

For information on restoring a bad superblock, see “How to Restore a Bad UFS Superblock (Solaris 10 6/06 Release)” on page 332 or “How to Restore a Bad Superblock (Solaris 8, 9, and 10 Releases)” on page 335.
1. Become superuser or assume an equivalent role.

2. For systems with mirrored root (/) file systems only: Detach the root (/) mirror before booting from the alternate device, or you risk corrupting the file system.
   For information on detaching the root (/) mirror, see “Working With Submirrors” in Solaris Volume Manager Administration Guide.

3. Identify the device, such as /dev/dsk/c0t0d0s0, of the root (/), /usr, or /var file system that needs to be checked.
   You'll need to supply this device name when booted from an alternate device. Identifying this device when you are already booted from the alternate device is more difficult.

4. Boot the system with the root (/), /usr, or /var file system that needs to be checked from an alternate device, such as a local Solaris DVD or the network, in single-user mode.
   Doing so ensures that there is no activity on these file systems.
   For example:
   ```
   # init 0
   ok boot net -s
   .
   .
   .
   #
   ```

5. Check the device that contains the root (/), /usr, or /var file system as identified in Step 3.
   If the hardware for the file system to be checked or repaired has changed, the device names might have changed. Check that the fsck -n message Last Mounted on ... indicates the expected device for the file system.
   In this example, the root (/) file system to be checked is /dev/dsk/c0t0d0s0.
   ```
   # fsck -n /dev/rdsk/c0t0d0s0
   ** /dev/rdsk/c0t0d0s0 (NO WRITE)
   ** Last Mounted on /
   .
   .
   
   fsck /dev/rdsk/c0t0d0s0
   ** /dev/rdsk/c0t0d0s0
   ** Last Mounted on /
   ** Phase 1 - Check Blocks and Sizes
   ** Phase 2 - Check Pathnames
   .
   .
   ```
6 Correct any reported fsck errors.
   For information on how to respond to the error message prompts while you interactively check
   one or more UFS file systems, see Chapter 20, "Resolving UFS File System Inconsistencies
   (Tasks)," in System Administration Guide: Advanced Administration.

7 If fsck cannot repair all of the problems after running it, see “Fixing a UFS File System That the
   fsck Command Cannot Repair” on page 331.

8 Mount the repaired file system to determine if any files exist 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. Try to use the
   grep command to match phrases within individual files and the file command to identify file
   types.
   Eventually, remove unidentifiable files or directories left in the lost+found directory so that it
   doesn’t fill up unnecessarily.

9 Bring the system back to multiuser mode.
   # init 6

10 For systems with mirrored root (/) file systems only: Reattach the root (/) mirror.

▼ How to Check Other UFS File Systems (Not root (/),
    /usr, or /var)

   For new information about fsck in the Solaris 10 6/06 release, see “Enhancements to UFS File
   System Utilities (fsck, mksf, and newfs)” on page 312. There is no need to rerun fsck if you see
   the following message:

   ***** FILE SYSTEM WAS MODIFIED *****

   However, it doesn’t harm the file system to rerun fsck after this message. This message is just
   informational about fsck’s corrective actions.

   This procedure assumes that the file system to be checked is unmounted.

   For information on restoring a bad superblock, see “How to Restore a Bad UFS Superblock
   (Solaris 10 6/06 Release)” on page 332 or “How to Restore a Bad Superblock (Solaris 8, 9, and 10
   Releases)” on page 335.

1 Become superuser or assume an equivalent role.
2 Unmount the local file system to ensure that there is no activity on the file system.
   Specify the mount point directory or /dev/dsk/device-name as arguments to the fsck command. Any inconsistency messages are displayed.
   For example:
   ```
   # umount /export/home
   # fsck /dev/rdsk/c0t0d0s7
   ** /dev/dsk/c0t0d0s7
   ** Last Mounted on /export/home
   .
   .
   .
   ```

3 Correct any reported fsck errors.
   For information on how to respond to the error message prompts while you interactively check one or more UFS file systems, see Chapter 20, “Resolving UFS File System Inconsistencies (Tasks),” in System Administration Guide: Advanced Administration.

4 If fsck cannot repair all of the problems after running it, see “Fixing a UFS File System That the fsck Command Cannot Repair” on page 331.

5 Mount the repaired file system to determine if there are any files in the lost+found directory.
   Individual files put in the lost+found directory by the fsck command are renamed with their inode numbers.

6 Rename and move any files put in the lost+found directory.
   If possible, rename the files and move them where they belong. Try to use the grep command to match phrases within individual files and the file command to identify file types.
   Eventually, remove unidentifiable files or directories left in the lost+found directory so that it doesn’t fill up unnecessarily.

Example 17–1 Interactively Checking UFS Non-root (/) or Non-/usr File Systems

The following example shows how to check the /dev/rdsk/c0t0d0s6 file system and correct the incorrect block count. This example assumes that the file system is unmounted.

```bash
# fsck /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%)
```

Interactively Checking and Repairing a UFS File System
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

This procedure assumes that the file system is unmounted or inactive.

1. Become superuser or assume an equivalent role.
2. Unmount the UFS file system.
   ```bash
   # umount /mount-point
   ```
3. Check the UFS file system with the preen option.
   ```bash
   # 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 17–2 Preening a UFS File System

The following example shows how to preen the `/export/home` file system.

```bash
# fsck -o p /export/home
```
Fixing a UFS File System That the fsck Command Cannot Repair

The `fsck` command operates in several passes, and a problem corrected in a later pass can expose other problems that are only detected by earlier passes. Therefore, it is sometimes necessary to run `fsck` until it no longer reports any problems. Doing so ensures that all errors have been found and repaired.

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, try to use the `ff`, `clri`, and `ncheck` commands to figure out and fix what is wrong. For information about how to use these commands, see the following references:

- `fsdb(1M)`
- `ff(1M)`
- `clri(1M)`
- `ncheck(1M)`

Ultimately, you might need to re-create the file system and restore its contents from backup media.

For information about restoring complete file systems, see "Restoring UFS Files and File System Backups (Task Map)" on page 369.

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 repartition the disk again before re-creating and restoring file systems. Check that the device cables and connectors are functional before replacing the disk device. 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. However, if the disk is too severely damaged, 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 10, "SPARC: Setting Up Disks (Tasks)," or Chapter 11, "x86: Setting Up Disks (Tasks)."
Restoring a Bad UFS 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 these copies or use `fsck`'s automatic search for backup superblocks feature. For more information about this feature, see "Automatic Search for Backup Superblocks" on page 313.

For more information about the superblock, see "Superblock" on page 340.

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 Solaris DVD, and attempt the following steps. If these steps fail, recreate the root (/) file system by using the `newfs` command and restore it from a backup copy.

How to Restore a Bad UFS Superblock (Solaris 10 6/06 Release)

If your file system has a bad UFS superblock, `fsck` automatically calculates an alternative superblock as seen in the following messages:

BAD SUPERBLOCK AT . . .

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS?
LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS?

Caution – If a file system with a damaged superblock was created with `newfs` or `mkfs` customized parameters, such as `ntrack` or `nsect`, using `fsck`'s automatically calculated superblock for the repair process could irreparably damage your file system.

In the case of a file system that was created with customized parameters and it has a bad superblock, `fsck` provides the following prompt to cancel the `fsck` session:

CANCEL FILESYSTEM CHECK?

Canceling the `fsck` session would be an appropriate response if this file system was created with customized parameters or if there is some other concern about running `fsck` on this file system.
1 Become superuser or assume an equivalent role.

2 Check the file system with the suspected bad superblock.
   
   # fsck /dev/rdsk/c0t1d0s0
   
   ** /dev/rdsk/c0t1d0s0
   
   BAD SUPERBLOCK at ... 

3 Determine how the file system was created and select one of the following:
   
   ▪ The file system was created with the newfs command.
     
     ▪ fsck responds that all superblocks are corrupt and it must use a generic superblock.
       Answer the fsck prompts as described in the example below.

   ** Caution – Do not use this option if the file system was created with customized parameters. This option should only be used as a last resort. Be prepared to restore the file system from a backup copy.**

   # fsck /dev/dsk/c1t2d0s0
   
   ** /dev/dsk/c1t2d0s0
   
   BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

   LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? no

   LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? yes

   SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.

   USE GENERIC SUPERBLOCK FROM MKFS? no

   USE GENERIC SUPERBLOCK FROM NEWFS? yes

   CALCULATED GENERIC SUPERBLOCK WITH NEWFS
   
   If filesystem was created with manually-specified geometry, using auto-discovered superblock may result in irrecoverable damage to filesystem and user data.

   CANCEL FILESYSTEM CHECK? no

   ** Last Mounted on
   ** Phase 1 - Check Blocks and Sizes
   ** Phase 2 - Check Pathnames
   ** Phase 3a - Check Connectivity
   ** Phase 3b - Verify Shadows/ACLs
   ** Phase 4 - Check Reference Counts
   ** Phase 5 - Check Cylinder Groups
   CORRECT GLOBAL SUMMARY
   SALVAGE? y
UPDATE STANDARD SUPERBLOCK? y
81 files, 3609 used, 244678 free (6 frags, 30584 blocks, 0.0% fragmentation)
***** FILE SYSTEM WAS MODIFIED *****

- fsck responds that it found an alternate superblock with a message similar to the following:

FOUND ALTERNATE SUPERBLOCK 32 WITH NEWFS

With this fsck scenario, follow the prompts as shown in "Automatic Search for Backup Superblocks" on page 313.

- The file system was created with the mkfs command.
- fsck responds that all superblocks are corrupt and must use a generic superblock. Answer the fsck prompts as described in the example below.

Caution – Do not use this option if the file system was created with customized parameters. This option should only be used as a last resort. Be prepared to restore the file system from a backup copy.

```
# fsck /dev/dsk/c1t2d0s0
** /dev/rdsk/c1t2d0s0
BAD SUPERBLOCK AT BLOCK 16: BLOCK SIZE LARGER THAN MAXIMUM SUPPORTED

LOOK FOR ALTERNATE SUPERBLOCKS WITH MKFS? yes

LOOK FOR ALTERNATE SUPERBLOCKS WITH NEWFS? no
SEARCH FOR ALTERNATE SUPERBLOCKS FAILED.
USE GENERIC SUPERBLOCK FROM MKFS? yes
CALCULATED GENERIC SUPERBLOCK WITH MKFS
If filesystem was created with manually-specified geometry, using auto-discovered superblock may result in irrecoverable damage to filesystem and user data.

CANCEL FILESYSTEM CHECK? no

** Last Mounted on
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3a - Check Connectivity
** Phase 3b - Verify Shadows/ACLs
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cylinder Groups
CORRECT GLOBAL SUMMARY
SALVAGE? y

UPDATE STANDARD SUPERBLOCK? y
```
fsck responds that it found an alternate superblock with a message similar to the following:

```
FOUND ALTERNATE SUPERBLOCK 32 WITH MKFS
```

With this fsck scenario, follow the prompts as shown in “Automatic Search for Backup Superblocks” on page 313.

4 Answer the prompts to salvage and restore the superblock.
There is no need to rerun fsck when you see the following message:
```
***** FILE SYSTEM WAS MODIFIED *****
```
However, it doesn’t harm the file system to rerun fsck after this message. This message is just informational about fsck’s corrective actions.

▼ How to Restore a Bad Superblock (Solaris 8, 9, and 10 Releases)

1 Become superuser or assume an equivalent role.

2 Determine whether the bad superblock is in the root (/), /usr, or /var file system and select one of the following:
   - If the bad superblock is in either the root (/), /usr, or /var file system, then boot from the network or a locally connected Solaris DVD.
     From a locally-connected DVD, use the following command:
     ```
     ok boot cdrom -s
     ```
     From the network where a boot or install server is already setup, use the following command:
     ```
     ok boot net -s
     ```
     If you need help stopping the system, see Chapter 12, “Boot an Oracle Solaris System (Tasks),” in Oracle Solaris Administration: Basic Administration or “Booting an x86 Based System by Using GRUB (Task Map)” in Oracle Solaris Administration: Basic Administration.
   - If the bad superblock is not in either the root (/), /usr, /var file system, change to a directory outside the damaged file system and unmount the file system.
     ```
     # umount /mount-point
     ```
3 **Display the superblock values by using the newfs -N command.**

```bash
# newfs -N /dev/rdsk/device-name
```

The command output 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 “Customizing UFS File System Parameters” on page 342.

4 **Provide an alternate superblock by using the fsck command.**

```bash
# 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 17–3 Restoring a Bad Superblock (Solaris 8, 9, and 10 Releases)**

The following example shows how to restore the superblock copy 5264.

```bash
# 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)
***** 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.

<table>
<thead>
<tr>
<th>Command and Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fsck -m</code></td>
<td>Checks whether a file system can be mounted</td>
</tr>
<tr>
<td><code>fsck -y</code></td>
<td>Assumes a yes response for all repairs</td>
</tr>
<tr>
<td><code>fsck -n</code></td>
<td>Assumes a no response for all repairs</td>
</tr>
<tr>
<td><code>fsck -o p</code></td>
<td>Noninteractively preens the file system, fixing all expected (innocuous) inconsistencies, but exits when a serious problem is encountered</td>
</tr>
</tbody>
</table>
This is a list of the reference information in this chapter.
- “Structure of Cylinder Groups for UFS File Systems” on page 339
- “Customizing UFS File System Parameters” on page 342

Structure of Cylinder Groups for UFS File Systems

When you create a UFS file system, the disk slice is divided into cylinder groups. A cylinder group is comprised of one or more consecutive disk cylinders. 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.

<table>
<thead>
<tr>
<th>Block Type</th>
<th>Type of Information Stored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boot block</td>
<td>Information used when the system is booted</td>
</tr>
<tr>
<td>Superblock</td>
<td>Detailed information about the file system</td>
</tr>
<tr>
<td>Inode</td>
<td>All information about a file</td>
</tr>
<tr>
<td>Storage or data block</td>
<td>Data for each file</td>
</tr>
</tbody>
</table>

The following sections provide additional information about the organization and function of these blocks.
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 KB in a slice.

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 the 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

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 inode was changed

The array of 15 disk-block 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 18–1** 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 when a file system is created. By default, data blocks are allocated in two sizes: an 8-KB logical block size, and a 1-KB 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 structure of a typical UFS file system, the following figure shows a series of cylinder groups in a generic UFS file system.

**Customizing UFS File System Parameters**

Before you alter the default file system parameters that are assigned by the `newfs` command, you need to understand them. This section describes these parameters:

- "Logical Block Size" on page 343
- "Fragment Size" on page 343
- "Minimum Free Space" on page 344
- "Rotational Delay" on page 344 (Obsolete)
- "Optimization Type" on page 344
- "Number of Inodes (Files)" on page 345

For a description of the command options that customize these parameters, see `newfs(1M)` and `mkfs_ufs(1M)`.
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 KB) for UFS file systems. The UFS file system supports block sizes of 4096 or 8192 bytes (4 or 8 KB). The recommended logical block size is 8 KB.

**SPARC only** – You can specify only the 8192-byte block size on the sun-4u platform.

To choose the best logical block size for your system, consider both the performance you want and the available space. For most UFS systems, an 8-KB 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 when most of the files are very large. Use a smaller logical block size for file systems when most of the files are very small. You can use the `quot -c filesystem` 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 the best size 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 KB. 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-KB file systems only). The lower bound is actually tied to the disk sector size, typically 512 bytes.

For multiterabyte file systems, the fragment size must be equal to the file system block size.
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, consider 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 when most of the files are large. Use a smaller fragment size for file systems when 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 MB/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 them 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. The value is always set to 0, regardless of the value you specify.

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 sufficient free space exists, allocating disk blocks is relatively easy, 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 Inodes (Files)**

The number of bytes per inode specifies the density of inodes in the file system. The number is divided into the total size of the file system to determine the number of inodes to create. 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 KB) if the file system is less than 1 GB. If the file system is larger than 1 GB, the following formula is used:

<table>
<thead>
<tr>
<th>File System Size</th>
<th>Number of Bytes Per Inode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than or equal to 1 GB</td>
<td>2048</td>
</tr>
<tr>
<td>Less than 2 GB</td>
<td>4096</td>
</tr>
<tr>
<td>Less than 3 GB</td>
<td>6144</td>
</tr>
<tr>
<td>3 GB up to 1 TB</td>
<td>8192</td>
</tr>
<tr>
<td>Greater than 1 TB or created with <code>-T</code> option</td>
<td>1048576</td>
</tr>
</tbody>
</table>

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 about 16 TB of usable space, minus about one percent overhead. A `sparse` file can have a logical size of one terabyte. However, the actual amount of data that can be stored in a file is approximately one percent less than 1 TB 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.
This chapter describes the procedures for backing up file systems by using the `ufsdump` command.

For information on these procedures, see “Backing Up UFS Files and File System (Task Map)” on page 362.

For overview information about performing backups, see “Introduction to Backing Up and Restoring UFS File Systems” on page 347.

For information about backing up individual files to diskettes, see Chapter 21, “Copying Files and File Systems (Tasks).”

For additional information on the `ufsdump` command, see Chapter 23, “UFS Backup and Restore Commands (Reference).”

Introduction to Backing Up and Restoring UFS File Systems

*Backing up* file systems means copying file systems to removable media, such as tape, to safeguard against loss, damage, or corruption. *Restoring* file systems means copying 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 other media.
### TABLE 19–1

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
<th>For More Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back up one or more file systems to a local tape device or a remote tape device.</td>
<td>ufsdump</td>
<td>&quot;Backing Up UFS Files and File System (Task Map)&quot; on page 362 or Chapter 23, &quot;UFS Backup and Restore Commands (Reference)&quot;</td>
</tr>
<tr>
<td>Create read-only copies of file systems.</td>
<td>fssnap</td>
<td>Chapter 20, &quot;Using UFS Snapshots (Tasks)&quot;</td>
</tr>
<tr>
<td>Back up all file systems for systems on a network from a backup server.</td>
<td>Solstice Backup software</td>
<td>Solstice Backup 6.1 Administration Guide</td>
</tr>
<tr>
<td>Back up and restore an NIS+ master server.</td>
<td>nisbackup and nisrestore</td>
<td>System Administration Guide: Naming and Directory Services (NIS+)</td>
</tr>
<tr>
<td>Copy, list, and retrieve files on a tape or diskette.</td>
<td>tar, cpio, or pax</td>
<td>Chapter 21, &quot;Copying Files and File Systems (Tasks)&quot;</td>
</tr>
<tr>
<td>Copy the master disk to a clone disk.</td>
<td>dd</td>
<td>Chapter 21, &quot;Copying Files and File Systems (Tasks)&quot;</td>
</tr>
<tr>
<td>Restore complete file systems or individual files from removable media to a working directory.</td>
<td>ufsrestore</td>
<td>&quot;Restoring UFS Files and File System Backups (Task Map)&quot; on page 369</td>
</tr>
</tbody>
</table>

### 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 types of problems:

- 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 UFS 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 stand-alone systems and servers.
### TABLE 19–2  File Systems to Back Up for Stand-alone Systems

<table>
<thead>
<tr>
<th>File System to Back Up</th>
<th>Description</th>
<th>Back Up Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>root (/) – slice 0</td>
<td>This file system contains the kernel and possibly the /var directory. The /var directory contains temporary files, logging files, or status files, and possibly contains frequently updated system accounting and mail files.</td>
<td>At regular intervals such as weekly or daily</td>
</tr>
<tr>
<td>/usr – slice 6, /opt</td>
<td>The /usr and /opt file systems contain software and executables. The /opt directory is either part of root (/) or is its own file system.</td>
<td>Occasionally</td>
</tr>
<tr>
<td>/export/home – slice 7</td>
<td>This file system can contain the directories and subdirectories of all users on the stand-alone system.</td>
<td>More often than root (/) or /usr, perhaps as often as once a day, depending on your site’s needs</td>
</tr>
<tr>
<td>/export, /var, or other file systems</td>
<td>The /export file system can contain the kernel and executables for diskless clients. The /var directory contains temporary files, logging files, or status files.</td>
<td>As your site requires</td>
</tr>
</tbody>
</table>

### TABLE 19–3  UFS File Systems to Back Up for Servers

<table>
<thead>
<tr>
<th>File System to Back Up</th>
<th>Description</th>
<th>Back Up Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>root (/) – slice 0</td>
<td>This file system contains the kernel and executables.</td>
<td>Once a day to once a month depending on your site’s needs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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 intervals between once a week and once a month.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>File System to Back Up</td>
<td>Description</td>
<td>Back Up Interval</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>/export – slice 3</td>
<td>This file system can contain the kernel and executables for diskless clients.</td>
<td>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.</td>
</tr>
<tr>
<td>/usr – slice 6, /opt</td>
<td>The /usr and /opt file systems contain software and executables. The /opt directory is either part of root (/) or is its own file system.</td>
<td>Once a day to once a month, depending on your site's needs. These file systems are fairly static unless software is added or removed frequently.</td>
</tr>
<tr>
<td>/export/home – slice 7</td>
<td>This file system can contain the home directories of all the users on the system. The files in this file system are volatile.</td>
<td>Once a day to once a week.</td>
</tr>
</tbody>
</table>

### Choosing the Type of Backup

You can perform full or incremental backups by using the `ufs_dump` command. You can create a temporary image of a file system by using the `fssnap` command. The following table lists the differences between these types of backup procedures.

<table>
<thead>
<tr>
<th>Backup Type</th>
<th>Result</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>Copies a complete file system or directory</td>
<td>All data is in one place</td>
<td>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.</td>
</tr>
</tbody>
</table>
### TABLE 19–4 Differences Between Types of Backups (Continued)

<table>
<thead>
<tr>
<th>Backup Type</th>
<th>Result</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snapshot</td>
<td>Creates a temporary image of a file system</td>
<td>System can be in multiuser mode</td>
<td>System performance might degrade while the snapshot is created.</td>
</tr>
<tr>
<td>Incremental</td>
<td>Copies only those files in the specified file system that have changed since a previous backup</td>
<td>Easier to retrieve small changes in file systems</td>
<td>Finding which incremental tape contains a file can take time. You might have to go back to the last full backup.</td>
</tr>
</tbody>
</table>

### Choosing a Tape Device

The following table shows typical tape devices that are used for storing file systems during the backup process. The storage capacity depends on the type of drive and the data being written to the tape. For more information on tape devices, see Chapter 22, "Managing Tape Drives (Tasks)."

### TABLE 19–5 Typical Media for Backing Up File Systems

<table>
<thead>
<tr>
<th>Backup Media</th>
<th>Storage Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2-inch reel tape</td>
<td>140 MB (6250 bpi)</td>
</tr>
<tr>
<td>2.5-GB 1/4-inch cartridge (QIC) tape</td>
<td>2.5 GB</td>
</tr>
<tr>
<td>DDS3 4-mm cartridge tape (DAT)</td>
<td>12–24 GB</td>
</tr>
<tr>
<td>14-GB 8-mm cartridge tape</td>
<td>14 GB</td>
</tr>
<tr>
<td>DLT 7000 1/2-inch cartridge tape</td>
<td>35–70 GB</td>
</tr>
</tbody>
</table>

### High-Level View of Backing Up and Restoring UFS File Systems (Task Map)

Use this task map to identify all the tasks for backing up and restoring file systems. Each task points to a series of additional tasks, such as determining the type of backup to perform.
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 350

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 want to create a snapshot of a 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.
   "Backing Up UFS Files and File System (Task Map)" on page 362
   Chapter 20, "Using UFS Snapshots (Tasks)"
   Chapter 21, "Copying Files and File Systems (Tasks)"

4. (Optional) Restore a file system. 
   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.
   - Restore a file system that was created with the `tar`, `cpio`, or `pax` commands.
   "Restoring UFS Files and File System Backups (Task Map)" on page 369
   Chapter 21, "Copying Files and File Systems (Tasks)"

5. (Optional) Restore the root (/) or /usr file system.
   Restoring the root (/) or /usr file system is more complicated than restoring a noncritical file system. You need to boot from a local Solaris DVD or from the network while these file systems are being restored.
   "How to Restore a UFS root (/) and /usr File System" on page 380

### Considerations for Scheduling Backups

A backup schedule is the schedule that you establish to run the `ufsdump` command. This section identifies considerations to think about when you create a backup schedule. 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
- The time available for doing backups
- The time available for doing a full restore of a damaged file system
- The time available for retrieving individual files that are accidentally deleted
How Often Should You Do Backups?

If you do not need to minimize time requirements and the number 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 so that you can restore files from the last four weeks. This schedule requires at least four sets of tapes, one set for each week. You would then reuse the tapes each month. In addition, you should archive the monthly backups for at least a year. Then, keep yearly backups for a number of years.

Backup Interval Terms and Definitions

The following table describes backup interval terms and definitions.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Snapshot</td>
<td>Creates a temporary image of a file system.</td>
</tr>
<tr>
<td>Full backup</td>
<td>Copies a complete file system or directory.</td>
</tr>
</tbody>
</table>
| Incremental backup  | Copies only those files in the specified file system that have changed since a previous backup. Incremental backup types include the following:  
  Daily, cumulative – Copies a day's worth of file changes on Monday. Then, overwrites Monday's backup with file changes from Tuesday, Wednesday, and so on.  
  Daily, incremental – Copies a day's worth of file changes so that you have distinct tapes of Monday's changes, Tuesday's changes, and so on.  
  Weekly cumulative – Copies the files that have changed during the week and includes the previous week's file changes.  
  Weekly incremental – Copies the files that have changed during the week since the previous weekly backup. |

Guidelines for Scheduling Backups

The following table provides guidelines for scheduling backups. For additional backup schedule considerations, see “Considerations for Scheduling Backups” on page 352.
### Guidelines for Backup Schedules

<table>
<thead>
<tr>
<th>File Restoration Need</th>
<th>Backup Interval</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>To restore different versions of files (for example, file systems that are used for word processing)</td>
<td>Do daily incremental backups every working day. Do not reuse the same tape for daily incremental backups.</td>
<td>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 you might otherwise be unable to restore the needed version of the file. For example, a file that changed on Tuesday, and again on Thursday, goes onto Friday's lower-level backup appearing as 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.</td>
</tr>
<tr>
<td>To quickly restore a complete file system</td>
<td>Do lower-level backups more frequently.</td>
<td>—</td>
</tr>
<tr>
<td>To back up a number of file systems on the same server</td>
<td>Consider staggering the schedule for different file systems.</td>
<td>This way you’re not doing all level 0 backups on the same day.</td>
</tr>
<tr>
<td>To minimize the number of tapes used</td>
<td>Increase the level of incremental backups that are done across the week.</td>
<td>Only changes from day to day are saved on each daily tape.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td>Only changes from week to week (rather than the entire month) are saved on the weekly tapes.</td>
</tr>
<tr>
<td></td>
<td>Put each day’s and week’s incremental backups onto the same tape.</td>
<td>To do so, use the no rewind option of the <code>ufsdump</code> command, such as specifying <code>/dev/rmt/0n</code>.</td>
</tr>
</tbody>
</table>

## 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. A lower dump number always restarts a full or a cumulative backup. The following examples show the flexibility of the incremental dump procedure using levels 1–9.

**Example—Dump Levels for Daily, Cumulative Backups**

Doing daily, cumulative incremental backups is the most commonly used backup schedule and is recommended for most situations. The following example shows a schedule that uses a level 9 dump Monday through Thursday, and a level 5 dump on Friday restarts process.

In the preceding example, you could have used other numbers in the 1–9 range to produce the same results. The key is using the same number Monday through Thursday, with any lower number on Friday. For example, you could have specified levels 4, 4, 4, 4, 2 or 7, 7, 7, 7, 5.

**Example—Dump Levels for Daily, Incremental Backups**

The following example shows a schedule where you capture only a day’s work on different tapes. This type of backup is referred to as a daily, incremental backup. In this case, sequential dump level numbers are used during the week (3, 4, 5, 6) with a lower number (2) on Friday. The lower number on Friday restarts the processing.
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 specified sequence, as described in the examples.

**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 in the `/etc/dumpdates` file.

**Example—Daily Cumulative, Weekly Cumulative Backup Schedule**

Table 19–7 shows the most commonly used incremental backup schedule. This schedule is recommended for most situations. With this schedule, the following occurs:

- Each day, all files that have changed since the lower-level backup at the end of the previous week are saved.
- For each weekday level 9 backup, the previous level 0 or level 5 backup 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 backup for the first week.
- For each Friday level 5 backup, the closest lower-level backup is the level 0 backup done at the beginning of the month. Therefore, each Friday’s tape contains all the files changed during the month up to that point.

**TABLE 19–7  Daily Cumulative/Weekly Cumulative Backup Schedule**

<table>
<thead>
<tr>
<th></th>
<th>Floating</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thurs</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st of Month</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 1</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Week 2</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Week 3</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Week 4</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>
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 19–8  Contents of Tapes for Daily Cumulative/Weekly Cumulative Backup Schedule

<table>
<thead>
<tr>
<th></th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thurs</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>a b</td>
<td>a b c</td>
<td>a b c</td>
<td>a b c d</td>
<td>a b c d e</td>
</tr>
<tr>
<td>Week 2</td>
<td>g</td>
<td>g h</td>
<td>g h i</td>
<td>g h i</td>
<td>a b c d e f g h i j k</td>
</tr>
</tbody>
</table>

**Tape Requirements for the Daily Cumulative, Weekly Cumulative Schedule**

With this schedule, you need six tapes if you want to reuse daily tapes. However, you need nine tapes if you want to use four different daily tapes:

- One tape for the level 0 backup
- Four tapes for Fridays
- One or four daily tapes

If you need to restore a complete file system, you need the following tapes:

- The level 0 tape
- The most recent Friday tape
- The most recent daily tape since the last Friday tape, if any

**Example—Daily Cumulative, Weekly Incremental Backup Schedule**

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 backup for the first week. In addition, each Friday’s tape contains all the files that changed that week.

### TABLE 19–9  Daily Cumulative, Weekly Incremental Backup Schedule

<table>
<thead>
<tr>
<th></th>
<th>Floating</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thurs</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st of Month</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 1</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Week 2</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Week 3</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Week 4</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>6</td>
<td></td>
</tr>
</tbody>
</table>

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 19–10 Contents of Tapes for Daily Cumulative, Weekly Incremental Backup Schedule

<table>
<thead>
<tr>
<th></th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thurs</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>a b</td>
<td>a b c</td>
<td>a b c d</td>
<td>a b c d e</td>
<td>a b c d e f</td>
</tr>
<tr>
<td>Week 2</td>
<td>g</td>
<td>g h</td>
<td>g h i</td>
<td>g h i j</td>
<td>g h i j k</td>
</tr>
</tbody>
</table>

**Tape Requirements for the Daily Cumulative, Weekly Incremental Backup Schedule**

With this schedule, you need six tapes if you want to reuse daily tapes. However, you need nine tapes if you want to use four different daily tapes:

- One tape for the level 0 backup
- Four tapes for Fridays
- One or four daily tapes

If you need to restore a complete file system, you need the following tapes:

- The level 0 tape
- All the Friday tapes
- The most recent daily tape since the last Friday tape, if any

**Example—Daily Incremental, Weekly Cumulative Backup Schedule**

The following table shows a schedule where each weekday tape contains only the files that changed since the previous day. In addition, each Friday’s tape contains all files changed since the initial level 0 backup at the beginning of the month.

TABLE 19–11 Daily Incremental, Weekly Cumulative Backup Schedule

<table>
<thead>
<tr>
<th></th>
<th>Floating</th>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thurs</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st of Month</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Week 1</td>
<td></td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Week 2</td>
<td></td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Week 3</td>
<td></td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>Week 4</td>
<td></td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

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 19–12 Contents of Tapes for Daily Incremental, Weekly Cumulative Backup Schedule

<table>
<thead>
<tr>
<th>Mon</th>
<th>Tues</th>
<th>Wed</th>
<th>Thurs</th>
<th>Fri</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>a b</td>
<td>c d</td>
<td>e f g</td>
<td>h i</td>
</tr>
<tr>
<td>Week 2</td>
<td>j k l</td>
<td>m</td>
<td>n o</td>
<td>p q</td>
</tr>
</tbody>
</table>

### Tape Requirements for Daily Incremental, Weekly Cumulative Schedule

With this schedule, you need at least 9 tapes if you want to reuse daily tapes, which is not recommended. Preferably, you need 21 tapes if you save weekly tapes for a month: one tape for the level 0, 4 tapes for the Fridays, and 4 or 16 daily tapes.

- 1 tape for the level 0 backup
- 4 tapes for all the Friday backups
- 4 or 16 daily tapes

If you need to restore the complete file system, you need the following tapes:

- The level 0 tape
- The most recent Friday tape
- 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 19–13 Example of Monthly Backup Schedule for a Server

<table>
<thead>
<tr>
<th>Directory</th>
<th>Date</th>
<th>Dump Level</th>
<th>Tape Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>root (/)</td>
<td>1st Sunday</td>
<td>0</td>
<td>$n$ tapes</td>
</tr>
<tr>
<td>/usr</td>
<td>1st Sunday</td>
<td>0</td>
<td>$n$ tapes</td>
</tr>
<tr>
<td>/export</td>
<td>1st Sunday</td>
<td>0</td>
<td>$n$ tapes</td>
</tr>
<tr>
<td>/export/home</td>
<td>1st Sunday</td>
<td>0</td>
<td>$n$ tapes</td>
</tr>
<tr>
<td></td>
<td>1st Monday</td>
<td>9</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>1st Tuesday</td>
<td>9</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>1st Wednesday</td>
<td>5</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>1st Thursday</td>
<td>9</td>
<td>D</td>
</tr>
</tbody>
</table>
TABLE 19–13  Example of Monthly Backup Schedule for a Server  (Continued)

<table>
<thead>
<tr>
<th>Directory</th>
<th>Date</th>
<th>Dump Level</th>
<th>Tape Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Friday</td>
<td>9</td>
<td>E</td>
<td></td>
</tr>
<tr>
<td>1st Saturday</td>
<td>5</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>root (/)</td>
<td>2nd Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td>/usr</td>
<td>2nd Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td>/export</td>
<td>2nd Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td>/export/home</td>
<td>2nd Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td></td>
<td>2nd Monday</td>
<td>9</td>
<td>G</td>
</tr>
<tr>
<td></td>
<td>2nd Tuesday</td>
<td>9</td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>2nd Wednesday</td>
<td>5</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>2nd Thursday</td>
<td>9</td>
<td>J</td>
</tr>
<tr>
<td></td>
<td>2nd Friday</td>
<td>9</td>
<td>K</td>
</tr>
<tr>
<td></td>
<td>2nd Saturday</td>
<td>5</td>
<td>L</td>
</tr>
<tr>
<td>root (/)</td>
<td>3rd Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td>/usr</td>
<td>3rd Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td>/export</td>
<td>3rd Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td>/export/home</td>
<td>3rd Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td></td>
<td>3rd Monday</td>
<td>9</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>3rd Tuesday</td>
<td>9</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>3rd Wednesday</td>
<td>5</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>3rd Thursday</td>
<td>9</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>3rd Friday</td>
<td>9</td>
<td>Q</td>
</tr>
<tr>
<td></td>
<td>3rd Saturday</td>
<td>5</td>
<td>R</td>
</tr>
<tr>
<td>root (/)</td>
<td>4th Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td>/usr</td>
<td>4th Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td>/export</td>
<td>4th Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td>/export/home</td>
<td>4th Sunday</td>
<td>0</td>
<td>n tapes</td>
</tr>
<tr>
<td></td>
<td>4th Monday</td>
<td>9</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>4th Tuesday</td>
<td>9</td>
<td>T</td>
</tr>
</tbody>
</table>
TABLE 19–13  Example of Monthly Backup Schedule for a Server  (Continued)

<table>
<thead>
<tr>
<th>Directory</th>
<th>Date</th>
<th>Dump Level</th>
<th>Tape Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>4th Wednesday</td>
<td>5</td>
<td>U</td>
<td></td>
</tr>
<tr>
<td>4th Thursday</td>
<td>9</td>
<td>V</td>
<td></td>
</tr>
<tr>
<td>4th Friday</td>
<td>9</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>4th Saturday</td>
<td>5</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

With this schedule, you use 4n tapes, the number of tapes needed for 4 full backups of the root (/), /usr, /export, and /export/home file systems. Also, you need 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 three 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 previous lower-level backup is 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 of the month, use tape C to do a level 5 backup of the /export/home file system. The ufsdump command copies all files that changed since Sunday.

5. Do the Thursday and Friday level 9 backups of the /export/home file system 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, use tape F to do a level 5 backup of /export/home. The ufsdump command 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 four-week period, when you use them again.

7. Repeat steps 1–6 for the next three weeks, using tapes G–L and 4n tapes for the level 0 backup on Sunday, and so on.

8. For each four-week period, repeat steps 1–7, using a new set of tapes for the level 0 backups and reusing tapes A–X for the incremental backups. The level 0 tapes could be reused after three 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.
Preparing for File System Backups

The preparation for backing up file systems begins with planning, which is described in "Introduction to Backing Up and Restoring UFS File Systems" on page 347 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

For more information, see “Introduction to Backing Up and Restoring UFS File Systems” on page 347.

This section describes two other tasks you might need to perform before you back up file systems:

- Finding the names of file systems to back up
- Determining the number of tapes that are needed for a full backup

How to Find UFS File System Names

1. Display the contents of the `/etc/vfstab` file.

   `$ more /etc/vfstab`
Look in the \texttt{mount point} column for the name of the file system.

Use the directory name listed in the \texttt{mount point} column when you back up the file system.

\textbf{Example 19–1} Finding File System Names

In this example, the file systems to be backed up are \texttt{/}, \texttt{/usr}, and \texttt{/export/home}.

```
# more /etc/vfstab
#device device mount FS fsck mount mount
#to mount to fsck point type pass at boot options
#
fdd - /dev/fd fd - no -
/proc - /proc proc - no -
/dev/dsk/c0t0d0s1 - - swap - no -
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 /usr ufs 1 no -
/dev/dsk/c0t0d0s6 /dev/rdsk/c0t0d0s6 /export/home ufs 2 yes -
/devices - /devices devfs - no -
sharefs - /etc/dfs/sharetab sharefs - no -
ctfs - /system/contract ctfs - no -
objs - /system/object objs - no -
swap - /tmp tmpfs - yes -
```

\section*{How to Determine the Number of Tapes Needed for a Full Backup}

\begin{enumerate}
\item Become superuser or assume an equivalent role.
\item Estimate the size of the backup in bytes.
  \begin{verbatim}
  # ufsdump -s file-system
  
  Use the \texttt{-s} option to display the estimated number of bytes that are needed to do the backup if this is the first backup of the file system.
  
  Use the \texttt{--s} option to display the estimated number of bytes that are needed to do the backup if this is not the first backup of the file system.
  \end{verbatim}
\item Divide the estimated size by the capacity of the tape to determine how many tapes you need.
  For a list of tape capacities, see Table 19–5.
\end{enumerate}

\textbf{Example 19–2} Determining the Number of Tapes

In this example, the file system easily fits on a 150-MB tape.

```
# ufsdump s /export/home
178176
```
Backing Up a UFS 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 20, “Using UFS Snapshots (Tasks).”

- Be aware that backing up file systems when directory-level operations (such as creating, removing, and renaming files) and file-level activity are occurring simultaneously 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. In addition, you can direct the output to the system on which the tape device is located. Typically, the tape device 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 device is to pipe the output from the `ufsdump` command to the `dd` command. For information about using the `dd` command, see Chapter 21, “Copying Files and File Systems (Tasks).”

- If you are doing remote backups across the network, the system with the tape device must have entries in its `/rhosts` file for each client that will be using the device. Also, the system that initiates the backup must be included in the `/rhosts` file on each system that it will back up.

▼ How to Back Up a UFS File System to Tape

The following are general steps for backing up file systems by 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).
   For example:
   ```
   # shutdown -g30 -y
   ```

3 (Optional) Check the file system for consistency.
   For example:
   ```
   # fsck -m /dev/rdsk/c0t0d0s7
   ```
   The `fsck` 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 17, “Checking UFS File System Consistency (Tasks).”
If you need to back up file systems to a remote tape drive, follow these steps:

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 write-enabled into the tape drive.

7 Back up file systems.
   
   ```
   # ufsdump options arguments filenames
   ```
   
   You can back up file systems or directories, or files within file systems. For information on backing up individual files, see `tar(1)` or `cpio(1)`.

   The following examples show how to use the most common `ufsdump` options and arguments:
   
   - Example 19–3
   - Example 19–4
   - Example 19–5
   - Example 19–6

   For other `ufsdump` options and arguments, see Chapter 23, “UFS Backup and Restore Commands (Reference).”

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
   ```
Performing a Full Backup of a UFS root (/) Filesystem

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 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 filesystem being backed up.

For example:

```bash
# init 0
ok boot -s
.
.
# ufsdump 0ucf /dev/rmt/0 /
DUMP: Date of this level 0 dump: Mon Jul 12 10:00:50 2010
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsk/c0t0d0s0 (starbug:/) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Writing 63 Kilobyte records
DUMP: Estimated 1843276 blocks (900.04MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 1843252 blocks (900.03MB) on 1 volume at 6602 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 0 dump on Mon Jul 12 10:22:50 MDT 2010
# ufsrestore tf /dev/rmt/0
  2
  3 ./.lost+found
  4 ./.usr
  5 ./.var
  6 ./.export
  7 ./.bin
  8 ./.dev
.
.
# (Press Control-D to bring system to run level 3)
```

Performing an Incremental Backup of a UFS root (/) File System

The following example shows how to do an incremental backup of the root (/) file system in single-user mode. The following ufsdump options are included:

- **9** specifies 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.

```
# init 0
ok boot -s
# ufsdump 9ucf /dev/rmt/0 /
DUMP: Date of this level 9 dump: Mon Jul 12 10:52:50 2010
DUMP: Date of last level 0 dump: Mon Jul 12 10:50:50 2010
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: Writing 63 Kilobyte records
DUMP: Estimated 422 blocks (211KB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 376 blocks (188KB) on 1 volume at 1843 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 9 dump on Mon Jul 12 10:55:50 2010
# ufsrestore tf /dev/rmt/0
2
9 ./etc
304 ./etc/dumpdates
1117 ./devices
1118 ./devices/pseudo
3381 ./devices/pseudo/pts@0:1
.
.
.
```

```
# init 0
ok boot -s
# ufsdump 9ucf /dev/rmt/0 /
DUMP: Date of this level 9 dump: Mon Jul 12 10:33:14 2010
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsk/c2t1d0s0 (v880-brm-03:/root) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Writing 63 Kilobyte records
DUMP: Estimated 120694 blocks (58.93MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 120580 blocks (58.88MB) on 1 volume at 9026 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 9 dump on Mon Jul 12 10:33:14 2010
# ufsrestore tf /dev/rmt/0
2
9 ./etc
304 ./etc/dumpdates
1117 ./devices
1118 ./devices/pseudo
3381 ./devices/pseudo/pts@0:1
.
.
.
```
Example 19–5  Performing a Full Backup of a UFS Home Directory

The following example shows how to do a full backup of the `/export/home/kryten` home 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: Date of this level 0 dump: Wed Jul 10:35:18 2010
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsk/c0t0d0s7 (starbug:/export/home) to /dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Writing 63 Kilobyte records
DUMP: Estimated 105158 blocks (51.35MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 105082 blocks (51.31MB) on 1 volume at 5025 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 9 dump on Mon Jul 12 10:38:14 2010
```

```
# ufsrestore tf /dev/rmt/0
232 ./kryten
233 ./kryten/filea
234 ./kryten/fileb
235 ./kryten/filec
236 ./kryten/letters
237 ./kryten/letters/letter1
238 ./kryten/letters/letter2
239 ./kryten/letters/letter3
240 ./kryten/reports
241 ./kryten/reports/reportA
242 ./kryten/reports/reportB
243 ./kryten/reports/reportC
```

Example 19–6  Performing a Full Backup to a Remote System (Solaris 10 Data to Solaris 10 System)

The following example shows how to do a full backup of a local `/export/home` file system on a Solaris 10 system (`mars`) to a tape device on a remote Solaris 10 system (`earth`) in single-user mode. The systems must be set up so that remote access is enabled. The following `ufsdump` options are included:

- `0` specifies 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: Date of this level 0 dump: Wed Jul 10:35:18 2010
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/rdsk/c0t0d0s7 (starbug:/export/home) to earth:/dev/rmt/0.
DUMP: Mapping (Pass I) [regular files]
DUMP: Mapping (Pass II) [directories]
DUMP: Writing 63 Kilobyte records
DUMP: Estimated 105158 blocks (51.35MB).
DUMP: Dumping (Pass III) [directories]
DUMP: Dumping (Pass IV) [regular files]
DUMP: 105082 blocks (51.31MB) on 1 volume at 5025 KB/sec
DUMP: DUMP IS DONE
DUMP: Level 9 dump on Mon Jul 12 10:38:14 2010
```

```
# ufsrestore tf earth:/dev/rmt/0
232 ./kryten
233 ./kryten/filea
234 ./kryten/fileb
235 ./kryten/filec
236 ./kryten/letters
237 ./kryten/letters/letter1
238 ./kryten/letters/letter2
239 ./kryten/letters/letter3
240 ./kryten/reports
241 ./kryten/reports/reportA
242 ./kryten/reports/reportB
243 ./kryten/reports/reportC
```
Restoring UFS Files and File System Backups (Task Map)

The following task map describes the procedures associated with restoring files and file systems.

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prepare to restore files and file systems.</td>
<td>Identify the file systems or files to be restored, the tape device, and how you will restore them.</td>
<td>“Preparing to Restore UFS Files and File Systems” on page 370</td>
</tr>
<tr>
<td>Determine which tapes to use.</td>
<td>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.</td>
<td>“How to Determine Which Tapes to Use” on page 371</td>
</tr>
<tr>
<td>Restore files.</td>
<td>Choose one of the following restore methods:</td>
<td>“How to Restore UFS Files Interactively” on page 372</td>
</tr>
<tr>
<td></td>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
Preparing to Restore UFS 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. You can also use this command to restore one or more single files from any backup tape. If you run the `ufsrestore` command 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 to restore from
- 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 device

Determining the UFS 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 UFS File System Names” on page 362.

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 device 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 22, “Managing Tape Drives (Tasks).”

Restoring UFS 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, you should restore them to a temporary location, such as the /var/tmp directory. After you verify the files, you can move them to their proper locations. However, 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 to a temporary directory on the same file system where the files will reside. Then, 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. 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 dump level and from most recent to least recent date, unless the user requests otherwise.
If you have online archive files, identify the correct media.

```bash
# 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 from the tape.
- `archive-name` Identifies the online archive file name.
- `./path/filename` Identifies the file name or file names 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)`.

Insert the media that contains the files to be restored in the drive and verify the correct media.

```bash
# ufsrestore tf /dev/rmt/n ./path/filename /path/filename
```

Be sure to use the complete path for each `filename`. If a file is in the backup, its name and inode number are listed. Otherwise, a message states that the file is not on the volume.

If you have multiple backup files on the same tape, position the tape at the backup file you want to use.

```bash
# ufsrestore xfs /dev/rmt/n tape-number
```

### Example 19–7 Determining Which Tapes to Use

The following example shows how to check if the `/etc/passwd` file is in the online archive.

```bash
# ufsrestore ta /var/tmp/root.archive ./etc/passwd
```

The following example shows how to verify that the `/etc/passwd` file is on the backup tape.

```bash
# ufsrestore tf /dev/rmt/0 ./etc/passwd
```

#### How to Restore UFS 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 to 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-name]

   b. Change to a directory.
      ufsrestore> cd directory-name

   c. Create a list of files and directories that you want to restore.
      ufsrestore> add filenames

   d. (Optional) Remove any directory or file from the list of files to be restored, if necessary.
      ufsrestore> delete filename

7 (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 maintain the mode of the current directory, 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 that all the files and directories you specified in the list have been restored.

13 Move the files to the proper directories.

Example 19–8 Restoring UFS 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
.
.:
...:
.SunW/ export/ net/ sbin/ usr/
Sources/ etools/ opt/ scde/ var/
b/ home/ ptools/ set/ vol/
bin kernel/ pkg/ share/
de/ lib/ platform/ shared/
dev/ lost+found/ proc/ src/
/etc/ mnt/ rtools/ 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
# cd etc
# mv passwd /etc
# mv shadow /etc
# ls -l /etc
```
How to Restore Specific UFS Files Noninteractively

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 to 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 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. Press Return.
   ```
   Specify next volume #: 1
   ```
   The file or files are restored to the current working directory.
7. To maintain the mode of the current directory, 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 that all the files and directories you specified in the list have been restored.
9. Move the files to the proper directories.
Example 19–9  Restoring Specific Files Noninteractively

The following example shows how to noninteractively 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 Jul 14 13:36:17 2010
Dumped from: the epoch
Level 0 dump of / on starbug:/dev/dsk/c0t0d0s0
Label: none
Extract directories from tape
Initialize symbol table.
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.
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 19–10  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 by 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 UFS 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
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 a UFS root (/) and /usr File System” on page 380.

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

2. **If necessary, unmount the file system.**
   ```
   # umount /dev/rdsk/device-name
   Or:
   # umount /file-system
   ```

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 that 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 dump level 0 backup is restored. If the backup 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 dump level 0 and continuing until you reach the highest dump 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 has been 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 0ucf /dev/rmt/n /dev/rdsk/device-name
   A level 0 backup is performed. Always immediately do a full backup of a newly created file
   system because the ufsrestore command 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 19–11 Restoring a Complete UFS File System

The following example shows how to restore the /export/home file system.
Restoring UFS Files and File Systems

Chapter 19 • Backing Up and Restoring UFS File Systems (Overview/Tasks) 379

# newfs /dev/rdsk/c0t0d0s7
newfs: /dev/rdsk/c0t0d0s7 last mounted as /export/home
newfs: construct a new file system /dev/rdsk/c0t0d0s7: (y/n)? y
/dev/rdsk/c0t0d0s7: 143380864 sectors in 23334 cylinders of 48 tracks, 128 sectors
69999.1MB in 1459 cyl groups (16 c/g, 48.00MB/g, 5824 i/g)
super-block backups (for fsck -F ufs -o b=?) at:
32, 98464, 196928, 295392, 393752, 492112, 590472, 688832, 787192, 885552
Initializing cylinder groups:

super-block backups for last 10 cylinder groups at:
142447776, 142546208, 142644640, 142743072, 142841504, 142940036, 143038568,
143137100, 143235632, 143334164
# mount /dev/dsk/c0t0d0s7 /mnt
# cd /mnt
# ufsrestore rvf /dev/rmt/0
Verify volume and initialize maps
Media block size is 126
Dump date: Wed Jul 14 12:45:33 2010
 Dumped from: the epoch
Level 0 dump of /export/home on starbug:/dev/dsk/c0t0d0s7
Label: none
Begin level 0 restore
Initialize symbol table.
Extract directories from tape
Calculate extraction list.
Warning: ./lost+found: File exists
Make node ./rimmer
Make node ./rimmer/wdir
Make node ./lister
Make node ./pmorph
Make node ./inquisitor
Make node ./kryten
Make node ./kryten/letters
Make node ./kryten/reports
Extract new leaves.
Check pointing the restore
extract file ./rimmer/words
extract file ./rimmer/words1
extract file ./rimmer/words2
extract file ./rimmer/words3
extract file ./rimmer/wdir/words
extract file ./rimmer/wdir/words1
extract file ./rimmer/wdir/words2
extract file ./rimmer/wdir/words3
.
.
Add links
Set directory mode, owner, and times.
Check the symbol table.
Check pointing the restore
# rm restoresymtable
# cd /
# umount /mnt
# ufsdump 0ucf /dev/rmt/0 /export/home
# mount /dev/dsk/c0t0d0s7 /export/home
# ls /export/home

▼ How to Restore a UFS root (/) and /usr File System

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 “SPARC: How to Set Up a Disk for a UFS Root File System” on page 200 or “x86: How to Set Up a Disk for a UFS Root File System” on page 218.

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 Create the links for the tape device.
   # tapes

7 Restore the root (/) file system.
   # ufsrestore rvf /dev/rmt/n
   The dump level 0 tape is restored.

8 Remove the tape and load the next level tape in the drive.
   Always restore tapes starting with dump level 0 and continuing from the lowest to highest dump level.

9 Continue restoring as needed.
   # ufsrestore rvf /dev/rmt/n
   The next level tape is restored.

10 Repeat Step 8 and Step 9 for each additional tape.

11 Verify that the file system has been restored.
   # ls
12 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.

13 Change to the root (/) directory.
   # cd /

14 Unmount the newly created file system.
   # umount /mnt

15 Check the new file system.
   # fsck /dev/rdsk/device-name
   The restored file system is checked for consistency.

16 Create the boot blocks on the root partition.
On a SPARC system:
   # 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 Example 19–12.
   On an x86 system:
   # /sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/cwtxdyz
   For more information, see installgrub(1M).
   For an example of using the installgrub command on an x86 based system, see Example 19–13.

17 Insert a new tape in the tape drive.

18 Back up the new file system.
   # ufsdump 0uf /dev/rmt/n /dev/rdsk/device-name
   A dump level 0 backup is performed. Always immediately do a full backup of a newly created file system because the ufsrestore command repositions the files and changes the inode allocation.

19 Repeat steps 5 through 16 for the /usr file system, if necessary.
20 Reboot the system.
   # init 6
   The system is rebooted.

Example 19–12 SPARC: Restoring a UFS 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 Solaris DVD 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/sun4u/lib/fs/ufs/bootblk
   /dev/rdsk/c0t3d0s0
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t3d0s0
# init 6

Example 19–13 x86: Restoring a UFS 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 GRUB failsafe boot session, local Solaris DVD, 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
#/sbin/installgrub /boot/grub/stage1 /boot/grub/stage2 /dev/rdsk/c0t3d0s0
stage1 written to partition 0 sector 0 (abs 2016)
stage2 written to to partition 0, 227 sectors starting 50 (abs 2066)
# ufsdump 0uf /dev/rmt/0 /dev/rdsk/c0t3d0s0
# init 6
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 383.

For overview information about performing backups, see “Introduction to Backing Up and Restoring UFS File Systems” on page 347.

### Using UFS Snapshots (Task Map)

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>For Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Create a UFS snapshot.</td>
<td>Create a read-only copy of a file system by using the <code>fssnap</code> command.</td>
<td>“How to Create a UFS Snapshot” on page 387</td>
</tr>
<tr>
<td>2. Display UFS snapshot information.</td>
<td>Identify UFS snapshot information such as the raw snapshot device.</td>
<td>“How to Display UFS Snapshot Information” on page 388</td>
</tr>
<tr>
<td>3. (Optional) Delete a UFS snapshot.</td>
<td>Delete a snapshot that is already backed up or no longer needed.</td>
<td>“How to Delete a UFS Snapshot” on page 389</td>
</tr>
<tr>
<td>4. (Optional) Back up a UFS snapshot.</td>
<td>Choose one of the following backup methods:&lt;br&gt; Create a full backup of a UFS snapshot by using the <code>ufsdump</code> command.&lt;br&gt; Create an incremental backup of a UFS snapshot by using the <code>ufsdump</code> command.</td>
<td>“How to Create a Full Backup of a UFS Snapshot (<code>ufsdump</code>)” on page 390&lt;br&gt; “How to Create an Incremental Backup of a UFS Snapshot (<code>ufsdump</code>)” on page 391</td>
</tr>
</tbody>
</table>
UFSSnapshotsOverview

You can use the fssnap command to back up file systems while the file system is mounted. This command creates 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 bitmap file that contains copies of pre snapshot data that has been modified since the snapshot was taken.

Keep the following key points in mind when specifying backing-store files:

- The destination path of the backing store files must have enough free space to hold the file system data. The size of the backing store files vary with the amount of activity on the file system.
- The backing store file location must be different from the file system that is being captured in a snapshot.
- The backing-store files can reside on any type of file system, including another UFS file system or an NFS file system.
- Multiple backing-store files are created when you create a snapshot of a UFS file system that is larger than 512 GB.
- Backing-store files are sparse files. The logical size of a sparse file, as reported by the lsz command, is not the same as the amount of space that has been allocated to the sparse file, as reported by the du command.

For more information about creating snapshots for a UFS file system larger than 512 GB, see “Creating a Multiterabyte UFS Snapshot” on page 386.

Why Use UFS Snapshots?

The UFS snapshots feature provides additional availability and convenience for backing up a file system because the file system remains mounted and the system remains in multiuser mode during backups. Then, you can use the tar or cpio commands to back up a UFS snapshot to
tape for more permanent storage. If you use the `ufsdump` command to perform backups, the system should be in single-user mode to keep the file system inactive when you perform backups.

The `fssnap` command gives administrators of non-enterprise-level systems the power of enterprise-level tools, such as Sun StorEdge Instant Image, without the large storage demands.

The UFS snapshots feature is similar to the Instant Image product. 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. 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.

This table describes specific differences between UFS snapshots and Instant Image.

<table>
<thead>
<tr>
<th>UFS Snapshots</th>
<th>Sun StorEdge Instant Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of the backing-store file depends on how much data has changed since the snapshot was taken</td>
<td>Size of the backing-store file equals the size of the entire file system being copied</td>
</tr>
<tr>
<td>Does not persist across system reboots</td>
<td>Persists across system reboots</td>
</tr>
<tr>
<td>Works on UFS file systems</td>
<td>Cannot be used with root (/) or /usr file systems</td>
</tr>
<tr>
<td>Available in the Oracle Solaris release</td>
<td>Part of Sun StorEdge products</td>
</tr>
</tbody>
</table>

**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. However, they 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 initially uses no space, and then it grows quickly, especially on heavily used systems. Make sure that the backing-store file has enough space to expand. 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.

You can also specify a directory for the backing-store path, which means a backing store file is created in the directory specified. For example, if `/var/tmp` is specified for the backing-store path, the following backing-store file is created.

```
/var/tmp/snapshot0
```

If you created one large root (`/`) file system instead of creating separate file systems for `/export/home`, `/usr`, and so on, you will be unable to create a snapshot of those separate file systems. For example, this system does not have a separate file system for `/usr` as indicated under the `Mounted on` column:

```
# df -k /usr
Filesystem  kbytes  used  avail  capacity  Mounted on
/dev/dsk/c0t0d0s0  3618177  2190002  1391994   62%  /
```

If you attempt to create a snapshot for the `/usr` file system, you will see a message similar to the following:

```
# fssnap -F ufs -o bs=/snaps/usr.back.file /usr
snapshot error: Invalid backing file path
```

This message indicates that you cannot have the backing store file on the same file system as the file system being snapped, which is the case for the `/usr` file system, in this example.

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

**Creating a Multiterabyte UFS Snapshot**

Creating a multiterabyte UFS snapshot is identical to creating a snapshot for a smaller UFS file system. The only difference is that multiple backing store files are created for each 512 GB of file system space.

Keep the following key points in mind when creating a snapshot for a file system that is larger than 512 GB:

- Multiple backing store files are created.
  - If you specify a backing store file name when the snapshot is created, then the subsequent backing store file names will be iterated based on the file name that you specify. The subsequent backing-store files will have the same name, but with the suffixes `.2`, `.3`, and so on.
If you only specify a backing store file destination (or directory) and not a backing store file name, then multiple backing store file names will be created and iterated with the suffixes .2, .3, and so on.

The fssnap -i command only reports the first backing store file name even if multiple backing store files have been created. However, the reported backing-store length is the combined sizes of all the backing store files for the snapshot.

Note – Backing-store files are sparse files. The logical size of a sparse file, as reported by the ls command, is not the same as the amount of space that has been allocated to the sparse file, as reported by the du command.

After you have backed up the snapshot or you would just like to remove the snapshot, you will have to remove the backing store files manually if you did not use the unlink option when the snapshot was created.

For an example of creating a snapshot for a file system that is larger than 512 GB, see Example 20–2.

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 captured using UFS snapshots.

5 Verify that the snapshot has been created.
   
   # /usr/lib/fs/ufs/fssnap -i /file-system
Creating a UFS Snapshot

Example 20–1  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. 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 MB.

```
# fssnap -F ufs -o maxsize=500m,bs=/scratch/usr.back.file /usr
/dev/fssnap/1
```

Example 20–2  Creating a Multiterabyte UFS Snapshot

The following example shows how to create a snapshot of a 1.6 TB UFS file system.

```
# fssnap -F ufs -o bs=/var/tmp /datab
# /usr/lib/fs/ufs/fssnap -i /datab
```

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 file system 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 all current snapshots.

For example:

```
# /usr/lib/fs/ufs/fssnap -i
```

```text
Snapshot number : 0
Block Device : /dev/fssnap/0
```
3 Display detailed information about a specific snapshot.

For example:

```bash
# /usr/lib/fs/ufs/fssnap -i /export
Snapshot number : 1
Block Device : /dev/fssnap/1
Raw Device : /dev/rfssnap/1
Mount point : /export
Device state : idle
Back ing store path : /var/tmp/snapshot1
Back ing store size : 0 KB
Maximum backing store size : Unlimited
Snapshot create time : Mon Jul 12 10:37:50 2010
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. An unlinked 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 must manually delete the backing-store file.

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 manually delete the file.

▶ How to Delete a UFS Snapshot

You can delete a snapshot either by rebooting the system or by using the fssnap -d command. When you use this command, you must specify 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.

```bash
# /usr/lib/fs/ufs/fssnap -i
```

3 Delete the snapshot.

```bash
# fssnap -d /file-system
Deleted snapshot 1.
```
If you did not use the `-o unlink` option when you created the snapshot, manually delete the backing-store file.

```
# rm /file-system/backing-store-file
```

### Example 20-3 Deleting a UFS Snapshot

The following example shows how to delete a snapshot and assumes that the `-o unlink` option was not used.

```
# fssnap -i
  0 /export/home
  1 /export
# fssnap -d /export
Deleted snapshot 1.
# rm /var/tmp/snapshot1
```

## Backing Up a UFS Snapshot

You can create a full backup or an incremental backup of a 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. So, 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 procedure 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 : 1
   Block Device : /dev/fssnap/1
   Raw Device : /dev/rfssnap/1
   Mount point : /usr
   Device state : idle
   Backing store path : /var/tmp/snapshot2
   ```
Backing store size : 544 KB
Maximum backing store size : Unlimited
Snapshot create time : Mon Jul 12 10:37:50 2010
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 has been backed up.

For example:

# ufsrestore tf /dev/rmt/0

How to Create an Incremental Backup of a UFS Snapshot (ufsdump)

Back up a UFS snapshot incrementally means that only the files that have been modified since the last snapshot are backed up. Use the ufsdump command with the 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'

In this example, the -o raw option is used 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 (such as the ufsdump command) that require the raw device instead.

3 Verify that the snapshot has been 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 a file system from the backup, restore as if you had taken the backup directly from the original file system. Such a restore uses the `ufsrestore` command. For information on using the `ufsrestore` command to restore a file or file system, see "Restoring UFS Files and File System Backups (Task Map)" on page 369.
This chapter describes how to copy 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 Directories Between File Systems (cpio)” on page 395
- “How to Copy Files to a Tape (tar)” on page 397
- “How to List the Files on a Tape (tar)” on page 398
- “How to Retrieve Files From a Tape (tar)” on page 398
- “Copying Files to a Tape With the pax Command” on page 399
- “How to Copy All Files in a Directory to a Tape (cpio)” on page 400
- “How to List the Files on a Tape (cpio)” on page 401
- “How to Retrieve All Files From a Tape (cpio)” on page 402
- “How to Retrieve Specific Files From a Tape (cpio)” on page 403
- “How to Copy Files to a Remote Tape Device (tar and dd)” on page 404
- “How to Extract Files From a Remote Tape Device” on page 405

**Commands for Copying File Systems**

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.

The following table describes various backup and restore commands that are available in the Oracle Solaris release. For enterprise environments, consider using a enterprise-level backup product. Information about enterprise-level backup products is available on the Oracle Technical Network.

The following table describes the advantages and disadvantages of some of these commands.
### Advantages and Disadvantages of tar, pax, and cpio Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tar</strong></td>
<td>Use to copy files and directory subtrees to a single tape.</td>
<td>- Available on most UNIX operating systems&lt;br&gt;- Public domain versions are readily available</td>
<td>- Is not aware of file system boundaries&lt;br&gt;- Length of full path name cannot exceed 255 characters&lt;br&gt;- Cannot be used to create multiple tape volumes</td>
</tr>
<tr>
<td><strong>pax</strong></td>
<td>Use to copy files, special files, or file systems that require multiple tape volumes. Or, use when you want to copy files to and from POSIX-compliant systems.</td>
<td>- Better portability than the tar or cpio commands for POSIX-compliant systems&lt;br&gt;- Multiple vendor support</td>
<td>Same disadvantages as the tar command, except that the pax command can create multiple tape volumes.</td>
</tr>
<tr>
<td><strong>cpio</strong></td>
<td>Use to copy files, special files, or file systems that require multiple tape volumes. Or, use when you want to copy files from systems running current Oracle Solaris releases systems to systems running older Solaris releases.</td>
<td>- Packs data onto tape more efficiently than the tar command&lt;br&gt;- Skips over any bad spots in a tape when restoring&lt;br&gt;- Provides options for writing files with different header formats, such as (tar, ustar, crc, odc, bar), for portability between different system types&lt;br&gt;- Creates multiple tape volumes</td>
<td>The command syntax is more difficult than the tar or pax commands.</td>
</tr>
</tbody>
</table>

The following sections describes step-by-step instructions and examples of how to use these commands.
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 to pipe the output to the `cpio` command.

**How to Copy Directories Between File Systems (cpio)**

1. Change to the appropriate directory.
   ```
   # cd filesystem1
   ```

2. 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 from the bottom 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. The 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.
Verify that the copy was successful by displaying the contents of the destination directory.

```bash
# cd filesystem2
# ls
```

If appropriate, remove the source directory.

```bash
# rm -rf filesystem1
```

### Example 21–1 Copying Directories Between File Systems (cpio)

```bash
# 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.

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 407.

### Copying Files to Tape (tar Command)

Here is information 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 that 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/0 filenames
   
   c Indicates that you want to create an archive.
   v Displays the name of each file as it is archived.
   f /dev/rmt/0 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. 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/0

   For more information on listing files on a tar tape, see "How to List the Files on a Tape (tar)" on page 398.

Example 21–2 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/0
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/
   
   t List 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/ Indicates the tape device.
   ```

Example 21-3  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
drwxr-xr-x 0/0 0 Jul 14 13:50 2010 reports/
-r--r--r-- 0/0 206663 Jul 14 13:50 2010 reports/reportC
-r--r--r-- 0/0 206663 Jul 14 13:50 2010 reports/reportB
-r--r--r-- 0/0 206663 Jul 14 13:50 2010 reports/reportA
```

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/ [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/ 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 have been copied.
   
   ```
   $ ls -l
   ```
Retrieving Files on a Tape (\texttt{tar})

The following example shows how to retrieve all the files from the tape in drive 0.

\begin{verbatim}
$ 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
\end{verbatim}

Troubleshooting

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 (\texttt{tar})” on page 398.

Copying Files to a Tape With the \texttt{pax} Command

\section*{How to Copy Files to a Tape (\texttt{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.

\begin{verbatim}
$ pax -w -f /dev/rmt/n filenames
\end{verbatim}

\texttt{-w} Enables the write mode.

\texttt{-f /dev/rmt/n} Identifies the tape drive.

\texttt{filenames} Indicates the files and directories that you want to copy. Separate multiple files with spaces.

For more information, see \texttt{pax(1)}.

4. Verify that the files have been copied to tape.

\begin{verbatim}
$ pax -f /dev/rmt/n
\end{verbatim}

5. Remove the tape from the drive. Write the names of the files on the tape label.
Example 21–5  Copying Files to a Tape (pax)

The following 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 write-enabled tape into the tape drive.

3  Copy the files to 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). These options ensure portability to other vendors' 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 have been copied to 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 the command is 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.

5  Remove the tape from the drive. Write the names of the files on the tape label.

Example 21–6  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
1280 blocks
$ cpio -civt < /dev/rmt/0
-r--r--r-- 1 kryten staff 206663 Jul 14 13:52 2010, filea
-r--r--r-- 1 kryten staff 206663 Jul 14 13:52 2010, fileb
-r--r--r-- 1 kryten staff 206663 Jul 14 13:52 2010, filec
drwxr-xr-x 2 kryten staff 0 Jul 14 13:52 2010, letters
drwxr-xr-x 2 kryten staff 0 Jul 14 13:52 2010, reports
1280 blocks

▼ How to List the Files on a Tape (cpio)

Note – Listing the table of contents on a tape 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

Example 21–7  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
-r--r--r-- 1 kryten staff 206663 Jul 14 13:52 2010, filea
-r--r--r-- 1 kryten staff 206663 Jul 14 13:52 2010, fileb
-r--r--r-- 1 kryten staff 206663 Jul 14 13:52 2010, filec
drwxr-xr-x 2 kryten staff 0 Jul 14 13:52 2010, letters
drwxr-xr-x 2 kryten staff 0 Jul 14 13:52 2010, reports
1280 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 recreate 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/
   `-i` Extracts files from standard input.
   `-c` Specifies that the `cpio` command 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/` Specifies the output file.

4. Verify that the files were copied.

   `$ ls -l

Example 21–8 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.
   
   ```bash
   $ cpio -icv "*file" < /dev/rmt/0
   `` 
   
   - `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/0` Specifies the input file.

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

4. Verify that the files were copied.
   
   ```bash
   $ ls -l
   ```

Example 21–9 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.

```bash
$ 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. Configure ssh on the remote system so that you can access the tape drive. See “Configuring Secure Shell (Tasks)” in System Administration Guide: Security Services.

2. Change to the directory where you want to put the files.

3. Insert the tape into the tape drive.

4. Copy the files to a remote tape drive.

   ```
   $ tar cvf - filenames | ssh remote-host dd of=/dev/rmt/ n obs=block-size
   ```

   - `tar cvf` creates a tape archive, lists the files as they are archived, and specifies the tape device.
   - `-v` provides additional information about the tar file entries.
   - `- (Hyphen)` represents a placeholder for the tape device.
   - `filenames` identifies the files to be copied. Separate multiple files with spaces.
   - `ssh | remote-host` pipes the tar command's output to a remote system.
   - `dd of=/dev/rmt/n` represents the output device.
   - `obs=block-size` represents the blocking factor.

5. Remove the tape from the drive. Write the names of the files on the tape label.

Example 21–10 Copying Files to a Remote Tape Drive (tar and dd)

```bash
# tar cvf - * | ssh mercury dd of=/dev/rmt/0 obs=126b
password:
  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.**
   
   ```bash
   cd /var/tmp
   ```

3. **Extract the files from a remote tape device.**
   
   ```bash
   ssh remote-host dd if=/dev/rmt/0 | tar xvBpf -
   ```

   - `ssh remote-host` Indicates a secure shell that is started to extract the files from the tape device by using the `dd` command.
   - `dd if=/dev/rmt/0` Indicates the input device.
   - `| tar xvBpf -` Pipes the output of the `dd` command to the `tar` command, which is used to restore the files.

4. **Verify that the files have been extracted.**
   
   ```bash
   ls -l
   ```

**Example 21–11 Extracting Files From a Remote Tape Drive**

```bash
$ cd /var/tmp
$ ssh mercury dd if=/dev/rmt/0 | tar xvBpf -
password:
x answers/, 0 bytes, 0 tape blocks
x answers/test299, 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.0089, 77 bytes, 1 tape blocks
x tests/, 0 bytes, 0 tape blocks
x tests/test131, 84 bytes, 1 tape blocks
$ ls -l
```
Managing Tape Drives (Tasks)

This chapter describes how to manage tape drives in the Oracle Solaris OS.

This is a list of the information in this chapter:

- “Choosing Which Media to Use” on page 407
- “Backup Device Names” on page 408
- “Displaying Tape Drive Status” on page 410
- “Handling Magnetic Tape Cartridges” on page 411
- “Guidelines for Drive Maintenance and Media Handling” on page 411

This is a list of the step-by-step instructions in this chapter.

- “How to Display Tape Drive Status” on page 410
- “Retensioning a Magnetic Tape Cartridge” on page 411
- “Rewinding a Magnetic Tape Cartridge” on page 411

Choosing Which Media to Use

You typically back up Oracle 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)

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 tape devices that are used for backing up file systems. The storage capacity for each device depends on the type of drive and the data being written to the tape.
### TABLE 22–1  Media Storage Capacities

<table>
<thead>
<tr>
<th>Backup Media</th>
<th>Storage Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2-inch reel tape</td>
<td>140 MB (6250 bpi)</td>
</tr>
<tr>
<td>2.5-GB 1/4-inch cartridge (QIC) tape</td>
<td>2.5 GB</td>
</tr>
<tr>
<td>DDS3 4-mm cartridge tape (DAT)</td>
<td>12–24 GB</td>
</tr>
<tr>
<td>14-GB 8-mm cartridge tape</td>
<td>14 GB</td>
</tr>
<tr>
<td>DLT 7000 1/2-inch cartridge tape</td>
<td>35–70 GB</td>
</tr>
</tbody>
</table>

### Backup Device Names

You specify a tape or device 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 convention.

#### TABLE 22–2  Basic Device Names for Backup Devices

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tape</td>
<td>/dev/rmt/n</td>
</tr>
</tbody>
</table>

In general, you specify a tape device as shown in the following figure.

#### FIGURE 22–1  Tape Drive Device Names

```
/dev/rmt/XAbn
```

- Optional no-rewind \( (n) \) omit for re-wind
- BSD compatible behavior
- Optional density
  - 1 low
  - 2 medium
  - 3 high
  - 4 ultra
  - 5 compressed
- Drive number (0-n)
- Raw magnetic tape device directory
- Devices directory
If you don't specify the density, a tape drive typically writes at its “preferred” density. The preferred density usually means the highest density the tape drive supports. Most SCSI drives can automatically detect 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 a no-rewind option.

<table>
<thead>
<tr>
<th>Drive and Rewind Value</th>
<th>Use This Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>First drive, rewind</td>
<td><code>/dev/rmt/0</code></td>
</tr>
<tr>
<td>First drive, no rewind</td>
<td><code>/dev/rmt/0n</code></td>
</tr>
<tr>
<td>Second drive, rewind</td>
<td><code>/dev/rmt/1</code></td>
</tr>
<tr>
<td>Second drive, no rewind</td>
<td><code>/dev/rmt/1n</code></td>
</tr>
</tbody>
</table>

**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>
<thead>
<tr>
<th>Drive, Density, and Rewind Value</th>
<th>Use This Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>First drive, low density, rewind</td>
<td><code>/dev/rmt/0l</code></td>
</tr>
<tr>
<td>First drive, low density, no rewind</td>
<td><code>/dev/rmt/0ln</code></td>
</tr>
<tr>
<td>Second drive, medium density, rewind</td>
<td><code>/dev/rmt/1m</code></td>
</tr>
<tr>
<td>Second drive, medium density, no rewind</td>
<td><code>/dev/rmt/1mn</code></td>
</tr>
</tbody>
</table>
The additional density values are shown in "Backup Device Names" on page 408.

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 the tape drive status.
   
   # mt -f /dev/rmt/n status

3. Repeat steps 1–2, substituting tape drive numbers 0, 1, 2, 3, and so on to display information about all available tape drives.

Example 22–1 Displaying Tape Drive Status

The following example shows the status for a QIC-150 tape drive (/dev/rmt/0):

```
$ 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
```

The following example shows the status for an Exabyte tape drive (/dev/rmt/1):

```
$ 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
```
Handling Magnetic Tape Cartridges

If errors occur when a tape is being read, you can retension the tape, clean the tape drive, and then try again.

Retensioning 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.

Rewinding 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, periodically clean and check your tape drives to ensure correct operation. See your hardware manuals for instructions on procedures for cleaning a tape drive. You can check your tape hardware by copying some files to the tape, reading the files back, and then comparing the original files with the copied files.

Be aware that hardware can fail in ways that the system does not report.
Always label your tapes after a backup. This label should never change. Every time you do a backup, make another tape label that contains the following information:

- The backup date
- The name of the machine and file system that is backed up
- The backup level
- The tape number (1 of n, if the backup spans multiple volumes)
- 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.
This chapter contains reference information on the `ufsdump` and `ufsrestore` commands. This is a list of the information in this chapter:

- “How the `ufsdump` Command Works” on page 413
- “Specifying `ufsdump` Command Options and Arguments” on page 417
- “The `ufsdump` Command and Security Issues” on page 418
- “Specifying `ufsrestore` Options and Arguments” on page 418

For overview information about performing backups, see "Introduction to Backing Up and Restoring UFS File Systems" on page 347.

For information about backup tasks, see Chapter 19, "Backing Up and Restoring UFS File Systems (Overview/Tasks)."

**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 backup media.

**Determining Device Characteristics**

The `ufsdump` command needs to know only an appropriate tape 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.

You need to use the end-of-media options when the ufsdump command does not understand the way the device detects the end-of-media.

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 the ufsdump Command

The ufsdump command copies data only from the raw disk slice. If the file system is still active, any data in memory buffers is probably not copied. The backup done by the ufsdump command 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.

Purpose 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 following information:

- The file system backed up
- The dump level of the last backup
- The day, date, and time of the backup

For example:

```
# cat /etc/dumpdates
/dev/rdsk/c0t0d0s0 0 Wed Jul 7 13:36:17 2010
/dev/rdsk/c0t0d0s7 0 Thu Jul 8 12:53:12 2010
/dev/rdsk/c0t0d0s7 9 Thu Jul 8 13:41:48 2010
```
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 dump 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 to restore the entire file system.

Note – The /etc/dumpdates file is a text file that can be edited. However, edit it only at your own risk. If you make changes to the file that do not match your archive tapes, you might be unable 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. The destination can be one of the following:

- Local tape drive
- Local diskette drive
- Remote tape drive
- Remote 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 the dump-file argument.

Note – The dump-file argument can also point to a file on a local disk or on a remote disk. If done by mistake, this usage 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 a 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:
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/0m

User 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 408.

Remote Tape or Diskette Drive

You specify a remote tape device or a remote diskette by using the syntax host:device. The ufsdump command writes to the remote device when superuser on the local system has access to the remote system. If you usually run the ufsdump command as superuser, 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.

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 standard output with the ufsdump command and reading from standard input with the ufsrestore command. For 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/c0t0d0s7

You can specify the file system by its mount point directory (for example, /export/home), as long as an entry for it exists in the /etc/vfstab file.
For a complete description of device-naming conventions, see "Backup Device Names" on page 408.

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 the 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 how much space 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.

Specifying ufsdump Command Options and Arguments

This section describes how to specify 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 consist of multiple strings. The option letters and their associated arguments must be in the same order.
- **filenames** Identifies the files to back up. These arguments must always come last, each separated by a space.
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.

For a description of the ufsdump options, see `ufsdump(1M)`.

The ufsdump Command and Security Issues

If you are concerned about security, you should do the following:

- Require superuser access for the ufsdump command.
- Ensure superuser access entries are removed from `/.rhosts` files on clients and servers if you are doing centralized backups.

For general information on security, see `System Administration Guide: Security Services`.

Specifying ufsrestore Options and Arguments

The syntax of the ufsrestore command is as follows:

```
/usr/sbin/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`. For a description of the ufsrestore options, see `ufsrestore(1M)`.
- `arguments` follows the option string with the arguments that match the options. The option letters and their associated arguments must be in the same order.
- `filenames` specifies the file or files to be restored as arguments to the `x` or `t` options. These arguments must always come last, separated by spaces.
Index

Numbers and Symbols
/export/home directory, 279
1394 (FireWire) support, description, 113
4.3 Tahoe file system, 274
9660 DVD format, 31

A
accessing
disk devices, 62
iSCSI disks (how to), 241
removable media (how to), 32
tape devices, 65
adding
a SCSI device to a SCSI bus (how to), 78
a USB camera (how to), 115
a USB mass storage device (how to), 114
adding a new removable media drive (how to), 31
entry to /etc/vfstab file (how to), 294
PCI adapter card (how to), 88
swap to vfstab, 304
USB audio devices (how to), 129
VPPA communication service (how to), 147
allocated inodes, 321
autoconfiguration process, 52
autofs, 283
automounting, and /home, 283

B
backing up
and restoring file systems
commands for, 347
definition, 347
choosing file systems to, 348
full and incremental, definition, 350
preparing for (overview), 362–363
reasons for, 348
types of, 350
backup
device names, 408–410
record of incremental, 415
backup schedules
daily cumulative, weekly cumulative backups, 356
daily cumulative, weekly incremental backups, 357
daily incremental, weekly cumulative backups, 358
examples, 356, 361
for a server, 359–361
guidelines, 353
guidelines for, 352
using dump levels for, 354
bad block numbers, 323
bad inode number, 324
bad superblock, 332
block disk device interface
definition, 63
when to use, 63
blocks
bad, 323
boot, 340
data, 341–342
blocks (Continued)
directory data, 323
duplicate, 322
free, 342
indirect, 323
logical size, 343
regular data, 324
special inodes, 321
boot block, 340
BSD Fat Fast Filesystem, 274
bus-oriented disk controllers, 64
bytes (number per inode), 345

C
causes of file system damage, 318
cdrw command
description, 37
restricting access to (how to), 40
writing data CDs and DVDs and audio CDs
(overview), 39
CDs, names, 30
cfgadm
PCI hot-plugging (overview), 68
SCSI hot-plugging (overview), 68
changing, primary USB audio device (how to), 130
character special inodes, 321
checking
a file system on removable media (how to), 27
and repairing file systems, 325
file system size, 320
format and type of inodes, 321
free blocks, 321
free inodes, 321
inode list for consistency, 320
configuring
a SCSI controller (how to), 75
a SCSI device (how to), 76
a USB device (how to), 134
IB Port, HCA_SVC, or a VPPA device (how to), 144
IB pseudo device (how to), 145
IOC device (how to), 143
iSCSI target discovery (how to), 240
SATA device (how to), 91
configuring (Continued)
unidirectional or bidirectional CHAP authentication for iSCSI (how to), 244
connecting
a SCSI controller (how to), 78
a USB device, logically (how to), 135
copying
directories between file systems with cpio command
(overview), 395
groups of files with cpio command (overview), 395
individual files with cpio command (overview), 395
core files, finding and deleting, 179
cpio command
(overview), 395
copying directories between file systems (how to), 395
extract all files from tape (how to), 402
listing files on tape (how to), 401
creating
a data CD or DVD file system (how to), 42
a disk slice for a ZFS root file system
SPARC, 210
a file system on a DVD-RAM (how to), 27
a file system on a USB mass storage device (how to), 118
a file system on removable media (how to), 26
a full backup of UFS snapshot information (how to), 390
a Solaris partition and modifying the slices on a USB
mass storage device (how to), 122
a UFS snapshot
(example of), 388
a UFS snapshot (how to), 387
disk slice for a ZFS root file system (how to)
x86, 225
loopback file system (overview), 286
CTFS file system, 278
custom parameters for file systems, 342–345
cylinder group, 339–342

daily discrete backups, 355
damage to file systems, 318
data block, 324, 341–342
data directory blocks, 323
datadm command, 151
default
  file system for /tmp (TMPFS), 277
SunOS file system, 279
deleting
  core files, 179
  finding and deleting old/inactive files, 176
temporary files, 179
UFS snapshot information
  (example of), 390
detecting end of media
  cpio command, 395
  ufsdump command, 414
determining
  file system types, 284
  mounted file systems, 287
tape device name, 371
type of tape drive, 370
/dev/dsk directory, 63
/dev/rdsk directory, 63
devfsadm command, 61
device driver
  adding, 60
  definition, 51
device names
  backup, 408–410
  finding a file system name, 370
  finding tape, 371
devices, accessing, 61
df command
  -h option, 170
  -t option (total blocks), 170
  overview, 168
dfstab file, configuring for shared local removable media (how to), 34
directories
  copying between file systems with cpio command
    (overview), 395
displaying information about, 171, 172, 174, 176
  inodes, 321
  /proc, 277
  size of, 174, 176
directories (Continued)
  temporary, clearing out, 177, 179
  /tmp, 277
  unallocated blocks, 324
disabling
  removable media services, 20
  removable media services (how to), 31
  write protection on removable media (how to), 29
disconnect
  a USB device, logically (how to), 134
  disconnecting
    a SCSI controller (how to), 77
    a USB device subtree, logically (how to), 135
disk
  formatting a (overview), 161
  setting up for UFS file systems (overview)
    x86, 218
  setting up for ZFS file systems (overview)
    x86, 223–230
  when to format (overview), 184
disk-based file systems, 274
disk controllers, 64
disk for a ZFS file system
  setting up (how to)
    SPARC, 214
disk for UFS file systems
  setting up (how to)
    SPARC, 201
disk label
  creating (overview), 189
  description, 155
disk setup for a ZFS root file system
  connecting (how to)
    SPARC, 209
    x86, 224
disk slice
  creating for a ZFS root file system
    SPARC, 210
  for a ZFS root file system (how to)
    x86, 225
disk slices
  definition, 159
displaying information about (overview), 186
Index

disk space
  displaying information about
directory sizes, 174, 176
disk space owned per user, 176
  file sizes, 171, 172, 174
  finding and deleting old/inactive files, 176, 180
  finding files exceeding a size limit, 174
  finding large files, 172, 173
diskettes
  creating a file system on (how to), 26
  formatting with rmformat (how to), 25
  loading with volume management (how to), 24
disks
  creating disk slices and labeling a disk (how to)
  SPARC, 202
  creating disk slices for a UFS root file system
    (example of)
  SPARC, 203
determining if formatted (how to), 185
recovering a corrupted disk label (how to), 195
recovery of a corrupted disk label (overview), 195
setting up for a UFS file system (example of)
  SPARC, 204
  setting up for UFS file systems (overview)
  SPARC, 200
  setting up for ZFS file systems (overview)
  SPARC, 208
displaying
  directory information, 171, 172, 174
  disk slice information (overview), 186
file information
  file size, 171, 172
  listing newest, 177
  using the du command, 174
file system information, 176
IB communication services (how to), 147
InfiniBand device information (how to), 141
information about SCSI devices, 74
kernel IB clients of an HCA (how to), 145
PCI slot configuration information (how to), 85
removable media user (how to), 32
SATA devices (how to), 90
size of files, 171–172
swap space, 306–307
displaying (Continued)
  system configuration information, 52, 56
  USB bus information (how to), 132
  USB device information (how to), 116
DOS, file system, 275
driver not attached message, 53
du command, 174, 176
dump levels
  daily, incremental backups, 355
definition, 354
dumpadm command, 64
duplicate blocks, 322
DVD-RAM, creating a file system on (how to), 27
DVD-ROM, 276
dVDs, ISO 9660 format, 31
dynamic reconfiguration, InfiniBand devices, 140
dynamic reconfiguration (overview), 68

E
EFI label
  (overview), 156
  comparison with VTOC label, 157
  restrictions of, 157
troubleshooting problems, 158
eject command, removable media (how to), 33
ejecting, removable media (how to), 33
enabling
  removable media services (how to), 31
  uDAPL, 150
  write protection on removable media (how to), 29
end-of-media detection
cpio command, 395
ufsdump command, 414
error messages, iSCSI, 258
/etc/dfs/dfstab file, configuring for shared
removable media (how to), 34
/etc/dumpdates file, 414–415

F
FDFS file system, 278
FIFO inodes, 321
FIFOFS file system, 278
file system name, 370
file systems
  4.3 Tahoe, 274
  BSD Fat Fast, 274
checking and repairing, 325
creating (overview)
    loopback (LOFS), 286
CTFS, 278
custom parameters, 342–345
cylinder group structure, 339–342
damage to, 318
default SunOS, 279
disk-based, 274
DOS, 275
/export/home, 279
FDFS, 278
FIFOFS, 278
finding types, 284
fixing, 331
High Sierra, 275
ISO 9660, 275
making available (overview), 286
MNTFS, 280
mount table, 281
NAMEFS, 278
network-based, 276
OBJFS, 278
/opt, 280
PCFS, 275
preening, 330
/proc, 280
process (overview), 277
PROCFS (overview), 277
pseudo (overview), 276
reasons for inconsistencies, 319
SHAREFS, 278
sharing, 282–283
SPECFS, 278
stopping all processes accessing (how to), 298
SWAPFS, 278
TMPFS, 276
types of, 274
file systems (Continued)
  UFS, 274
  UNIX, 274
/var, 280
  which to back up, 348
  why you backup, 348
ZFS, 274
files
deleting
  See deleting
displaying information about
    listing, 171, 172
    size, 171, 172, 174, 176
displaying size of, 171–172
/etc/default/fs, 284
/etc/dfs/fstypes, 284
finding files exceeding a size limit, 174
in the /proc directory, 277
retrieving from tape with the tar command (how to), 398
sharing, 282–283
size of, 171, 172, 174, 176
find command
core files, 179
finding files exceeding a size limit, 174
old/inactive files, 176, 177
finding
  and deleting old/inactive files
    See deleting
file system name, 370
files exceeding a size limit, 174
large files, 172, 173
tape device name, 371
type of file system, 284
fixing inconsistent file systems, 331
format of inodes, 321
format utility
  (overview), 159
  analyze menu, 267
  creating a Solaris fdisk partition (how to), 231
  creating disk slices (how to)
    SPARC, 202
    creating disk slices for a UFS file systems (how to)
      x86, 220
format utility (Continued)
defect menu, 269
determining if a disk is formatted (how to), 184
displaying disk slice information (example of), 187
entering command names (how to), 270
fdisk menu, 266
features and benefits, 159
guidelines for using, 161
identifying disks on a system (examples of), 183
identifying disks on a system (how to), 182
input to, 269, 270
labeling a disk
  example of, 190
main menu, 263, 264
partition menu, 265, 266
recovering corrupted disk label (how to), 195
using help facility, 271
when to use, 160
formatting, diskettes with rmformat (how to), 25
formatting a disk, (overview), 161
fragment size, 343–344
free blocks, 321, 342
free hog slice, See donor slice
free inodes, 321
free space (minimum), 344
fsck command
  checking
    free blocks, 321
    free inodes, 321
    inode list size, 320
    superblock, 320
  conditions to repair, 319
  FACTIVE state flag, 318
  FSBAD state flag, 318
  FSCLEAN state flag, 318
  FSSTABLE state flag, 318
  preening, 330
  state flags, 318
  syntax and options, 337
  using interactively, 325
fsnap command, creating a UFS snapshot (how to), 387
fstypes file, 284

full backup
  (example of), 366, 368
definition, 350
fuser command
  finding if removable media is in use (how to), 32
  killing processes accessing removable media (how to), 32

H
High Sierra file system, 275
/home (automounted), 283
hot-plugging
  (overview), 68
  adding a SCSI device to a SCSI bus (how to), 78
  adding PCI adapter card (how to), 88
  configuring a SCSI controller (how to), 75
  configuring a SCSI device (how to), 76
  configuring a USB device (how to), 134
  connecting a SCSI controller (how to), 78
  disconnecting a SCSI controller with cfgadm command (how to), 77
  logically connecting a USB device (how to), 135
  logically disconnecting a USB device (how to), 134
  logically disconnecting a USB device subtree (how to), 135
  PCI devices (overview), 84
  removing a SCSI device (how to), 81
  removing PCI adapter card (how to), 86
  replacing an identical SCSI device on a SCSI controller (how to), 80
  unconfiguring a SCSI device (how to), 75
  unconfiguring a USB device (how to), 133
HSFS, See High Sierra file system

I
identifying
  devices, 54
  disks on a system (how to), 182
  primary USB audio device (how to), 129
  inconsistencies in file systems, 319
  incorrect . and .. entries, 324
incremental backup, 351, 415
  (example of), 366
indirect blocks, 323
InfiniBand devices
  adding a VPPA communication service (how to), 147
  configuring an IB Port, HCA_SVC, or a VPPA device (how to), 144
  configuring an IB pseudo device (how to), 145
  configuring an IOC device (how to), 143
  displaying (how to), 141
  displaying IB communication services (how to), 147
  displaying kernel IB clients of an HCA (how to), 145
dynamic reconfiguration (overview), 140
  overview, 137
removing an existing IB port, HCA_SVC, or a VPPA communication service (how to), 148
unconfiguring an IB Port, HCA_SVC, or a VPPA (how to), 143
unconfiguring an IB pseudo device (how to), 144
unconfiguring an IOC device (how to), 142
unconfiguring IB devices connected an HCA (how to), 146
  updating an IOC configuration (how to), 149
  updating the IP p_key tables (how to), 147
inode list size, 320
inode states, 321
inodes, 340–341
  bad number, 324
  block special, 321
  character special, 321
  checking format and type, 321
directory, 321
FIFO, 321
  link count, 322
  number of bytes per, 345
  regular, 321
  size, 323
  symbolic link, 321
installboot command, SPARC, 207
installboot command for a ZFS root file system, 213
installgrub command, x86, 222
installgrub command for a ZFS root file system, 228
installing boot blocks for a UFS root file system (how to), SPARC, 207
installing boot blocks for a ZFS root file system (how to), SPARC, 213
iSCSI (overview), 235
  accessing iSCSI disks (how to), 241
  configuring iSCSI target discovery (how to), 240
  configuring unidirectional or bidirectional CHAP authentication for (how to), 244
  general iSCSI error messages, 258
  modifying iSCSI initiator and target parameters (how to), 253
  monitoring your iSCSI configuration (how to), 251
  removing discovered iSCSI targets (how to), 242
  software and hardware requirements, 236
  static and dynamic target discovery, 238
  troubleshooting iSCSI configuration problems (how to), 256
iscsiadm add command, adding static or dynamic targets (example of), 240
iscsiadm list, displaying iSCSI configuration information (example of), 251
iscsiadm modify command
  enabling CHAP (example of), 245
  enabling or disabling static or dynamic targets (example of), 242
iscsiadm remove command, removing static or dynamic targets (example of), 243
ISO 9660 file system, 275
ISO standards, 9660 DVD format, 31

K
/kernel/drv directory, 52
killing
  all processes accessing a file system (how to), 298
  processes accessing removable media (how to), 32

L
large files, 173
level 0 backup, 354
link count of inodes, 322
listing
  files and directories, 171, 172, 177
loading, diskettes (how to), 24
log (record of dumps), 414–415
logical block size, 343
logical device name
  definition, 62
disk, 62
tape, 65
logical device names, removable media, 65
loopback file system (LOFS), creating (overview), 286
lost+found directory, 318
ls command
  checking directory sizes, 171
  -t option (size in bytes), 172
  -s option (size in blocks), 172
  -t option (newest files), 177

mounting (Continued)
  removable media
    automatic mounting compared to, 21, 22
    UFS file systems, 295
  mt command, 411

N
NAMEFS file system, 278
network-based file systems, 276
NFS
  description, 282
  server description, 282
  vfstab entry for, 295
nfsd daemon
  starting, 34
  verifying if running, 34
no media was found message, 25

M
maintaining tape drives, 411
maximums, finding files exceeding maximum size, 174
media was found message, 24
memory storage (virtual), definition, 301
minimum free space, 344
mkisofs command, create a data CD or DVD file system (how to), 42
MNTFS file system, 280
mntab file, 281
modifying
  iSCSI initiator and target parameters (how to), 253
  partitions and creating a PCFS file system on a USB mass storage device (how to), 119
monitoring, your iSCSI configuration (how to), 251
mount point, definition, 280
mount table, 281
mounting
  a file system with /etc/vfstab, 295
  a USB mass storage device (how to), 124, 125
  file systems automatically, 283
  NFS file systems, 295
  remote removable media manually (example of), 35

N
NAMEFS file system, 278
network-based file systems, 276
NFS
  description, 282
  server description, 282
  vfstab entry for, 295
nfsd daemon
  starting, 34
  verifying if running, 34
no media was found message, 25

O
OBJFS file system, 278
/opt directory, 280
options, for ufsdump command, 417

P
parameters (file system), 342–345
partition (swap), definition, 301
passwd file, restoring from tape (example of), 376
PCFS file system, 275
PCI devices
  adding PCI adapter card (how to), 88
  displaying PCI slot configuration information (how to), 85
  removing PCI adapter card (how to), 86
  troubleshooting PCI configuration problems, 90
physical device name
  definition, 62
preening file systems, 330
preparing
  for backing up (overview), 362–363
preparing (Continued)
to restore files (overview), 370–371
/proc directory, 277, 280
process file system (PROCFS), 277
PROCFS file system, (overview), 277
prtvtoc command, 64
(example of), 194
pseudo file systems, (overview), 276

Q
quot command, 176

R
raw disk device interface, 63
RCM script
commands for, 93
overview, 92
record of
  dumps, 414–415
  incremental backup, 415
registering, service provider in the DAT static registry
  (how to), 151
regular inodes, 321
removable media
  accessing (how to), 32
  accessing media on other systems (example of), 35
  adding a new removable media drive (how to), 31
  checking a file system on (how to), 27
  creating a file system on (how to), 26
  creating a file system on a DVD-RAM (how to), 27
  disabling or enabling removable media services (how to), 31
  ejecting (how to), 33
  enabling write protection on (how to), 29
  finding out if media is in use (how to), 32
  formatting diskettes with rmformat (how to), 25
  killing processes accessing (how to), 32
  loading diskettes (how to), 24
  management, benefits, 19
  mounting
    manual compared to automatic, 21, 22
  names, 30
  repairing bad blocks on removable media (how to), 28
  services, disabling, 20
removing
  a SCSI device (how to), 81
  discovered iSCSI targets (how to), 242
  existing IB port, HCA_SVC, or a VPPA communication service (how to), 148
  PCI adapter card (how to), 86
  repairing, bad blocks on removable media (how to), 28
  replacing, an identical SCSI device on a SCSI controller (how to), 80
  resetting, a USB device (how to), 135
  resolving, a failed SCSI unconfigure operation (how to), 83
  restoring bad superblock, 332
  restoring file systems
    complete (example of), 378
    complete (how to), 377
    determining which tapes to use (how to), 371
    preparing to (overview), 370–371
    root and /usr (SPARC) (example of), 382
    root or /usr (x86) (example of), 382
    type of tape drive, 370
  restoring files
    interactive restore (example of), 374
    non-interactive restore (example of), 376
  restricting, removable media access (how to), 40
  retrieving, files from tape with tar command (how to), 398
  rm command, 178, 179
Rock Ridge extension (HSFS file system), 275
root (/) file system or directory, 279

S
SATA devices
  configuring SATA device (how to), 91
  displaying SATA device information (how to), 90
  unconfiguring SATA device (how to), 90
scheduling backups, 352
SCSI devices
adding a SCSI device to a SCSI bus (how to), 78
configuring a SCSI controller (how to), 75
configuring a SCSI device (how to), 76
connecting a SCSI controller (how to), 78
disconnecting with \texttt{cfgadm} command (how to), 77
displaying information about (how to), 74
removing a SCSI device (how to), 81
replacing an identical SCSI device on a SCSI controller (how to), 80
resolving a failed SCSI unconfigure operation (how to), 83
troubleshooting SCSI configuration problem, 82
unconfiguring a SCSI controller (how to), 75

SCSI tape drives, 409
setting up
a disk for ZFS file systems (overview)
x86, 223–230
disks for ZFS file systems (overview)
SPARC, 208
setting up for UFS file systems
a disk (overview)
SPARC, 200
x86, 218
share command, 282
making removable media available to other systems (how to), 34
shareall command, 282
SHAREFS file system, 278
sharing, files, 282–283
size
checking file system, 320
directory, 174, 176
file, 171, 172, 174, 176
fragment, 343–344
inode, 323
slice (definition), 159
Solaris \texttt{fdisk} partition, guidelines, 230–231
space optimization type, 344–345
SPECFS file system, 278
specifying a disk slice, 64
starting, \texttt{nfssd} daemon, 34
state flag, \texttt{fsck}, 318
stopping
all processes for a file system (how to), 298
killing processes accessing removable media (how to), 32
storage (virtual memory), definition, 301
storage capacities (media), 351, 407
structure of cylinder groups, 339–342
SunOS default file system, 279
superblock, 320, 332, 340
\texttt{svidm disable}, (example of), 20
swap command, 64
swap file
adding to \texttt{vfstab}, 304
displaying, 306–307
swap partition, definition, 301
\texttt{swapadd} command, 304
SWAPFS file system, 278
symbolic links, 321
syntax, \texttt{fsck} command, 337
\texttt{sysdef} command, 54

T
tape
capacity, 417
capacity, 417
characteristics, 417
retrieving files from with \texttt{tar} command (how to), 398
sizes, 351, 407
storage capacities, 351, 407
tape devices (naming), 65
tape drive
determining type for restore, 370
maintaining, 411
maximum SCSI, 409
rewind, 409
tar command
(overview), 396
copying files to remote tape with \texttt{dd} command (how to), 404
listing files on tape (how to), 398
retrieving files from remote tape with \texttt{dd} command (how to), 405
retrieving files from tape (how to), 398
temporary directories, 177, 179
temporary file system (TMPFS), overview, 276
time (optimization type), 344–345
/tmp directory, 277, 280
TMPFS file system, overview, 276
troubleshooting
  a failed SCSI unconfigure operation, 83
  EFI disk labels, 158
  iSCSI configuration problems (how to), 256
  PCI configuration problems, 90
  SCSI configuration problems, 82
  USB audio device problems, 130
  USB mass storage devices, 126
type of file systems, 274
type of inodes, 321

U
uDAPL
  (overview), 149
  enabling (how to), 150
  registering a service provider in the DAT static registry (how to), 151
  unregistering a service provider in the DAT static registry (how to), 151
  updating the DAT static registry (how to), 151
UDF file system, 275
UFS file system, 274
  mounting, 295
  mounting with /etc/vfstab, 295
UFS file systems, displaying information about, 176
UFS snapshot
  creating (how to), 387
  creating a full backup of (how to), 390
  description, 385
ufsdump command
  end-of-media detection, 414
  full backup (example of), 366, 368
  how data is copied with, 414
  how it works, 413–417
  incremental backup (example of), 366
  limitations, 417
  options and arguments, 417
ufsdump command (overview), 364
ufsrestore command, 418
  determining which tapes to use (how to), 371
  preparing to use (overview), 370
unallocated directory blocks, 324
unallocated inodes, 321
unconfiguring
  a SCSI controller (how to), 75
  a USB device (how to), 133
  IB devices connected an HCA (how to), 146
  IB Port, HCA_SVC, or a VPPA Device (how to), 143
  IB pseudo device (how to), 144
  IOC device (how to), 142
  SATA device (how to), 90
UNIX file system, 274
unmounting
  a USB mass storage device (how to), 124, 125
unregistering, service provider in the DAT static registry (how to), 151
unsupported devices, 52
updating
  DAT static registry (how to), 151
  IOC configuration (how to), 149
  IP p_key tables (how to), 147
USB devices
  (overview), 107
  acronyms, 107
  adding a USB camera (how to), 115
  adding a USB mass storage device (how to), 114
  adding audio devices (how to), 129
  audio
    (overview), 128
    changing the primary device (how to), 130
    device ownership, 131
  bus description, 109
  bus-powered devices, 104
  cables for, 107
  composite device, 110
  compound device, 110
  configuring a USB device (how to), 134
  connect a USB device (how to), 135
  creating a file system on a mass storage device (how to), 118
USB devices (Continued)
creating a Solaris partition and modifying the slices on a mass storage device (how to), 122
device classes, 110
device nodes, 108
diskette devices (overview), 112
displaying bus information (how to), 132
displaying USB device information (how to), 116
drivers, 110
host controller and root hub, 105
hot-plugging (overview), 114
hotpluggable attribute, 113
identifying primary audio device (how to), 129
keyboards and mouse devices, 104
logically disconnecting a USB device (how to), 134
logically disconnecting a USB device subtree (how to), 135
modifying partitions and creating a PCFS file system on a mass storage device (how to), 119
mounting mass storage (how to), 125
mounting or unmounting a mass storage device (how to), 124
names of, 110
Oracle Solaris USB Architecture (USBA), 108
physical device hierarchy, 109
power management, 106
removable mass storage (overview), 111
resetting a USB device (how to), 135
troubleshooting audio device problems, 130
troubleshooting tips for mass storage devices, 126
unconfiguring a device (how to), 133
unmounting mass storage (how to), 125
USB 2.0 support, 104
wheel mouse support, 105
user ownership of disk space, 176

vfstab file (Continued)
entry for LOFS, 294
virtual memory storage, definition, 301

W
writing, data CDs and DVDs and audio CDs (overview), 39

Z
ZFS file system, 274
setting up a disk (how to)
x86, 229

V
/var directory, 280
/var/adm directory, controlling size of, 178
verifying, nfsd daemon is running, 34
vfstab file, 284, 304
adding entries to (how to), 294
adding swap to, 304