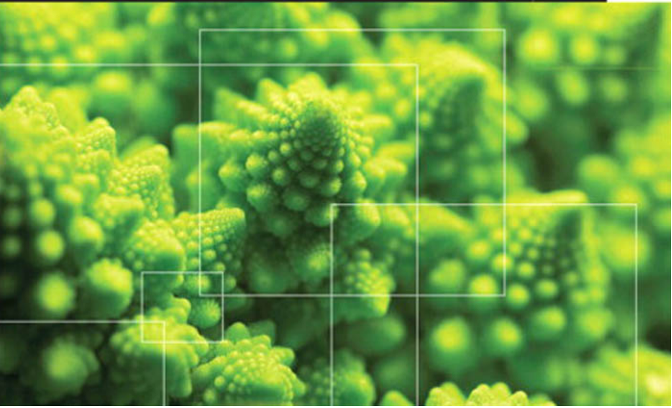
PRENTICE HALL

Oracle Solaris 11
System Administration

Bill Calkins



FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Oracle® Solaris 11 System Administration

Oracle® Solaris 11 System Administration

Bill Calkins

PRENTICE HALL

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Capetown • Sydney • Tokyo • Singapore • Mexico City

The publisher offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales
international@pearson.com

Visit us on the Web: informit.com/ph

Library of Congress Cataloging-in-Publication Data

Calkins, Bill.
Oracle Solaris 11 system administration / Bill Calkins.
pages cm
Includes index.
ISBN 978-0-13-300710-7 (paperback : alkaline paper)
1. Operating systems (Computers) 2. Solaris (Computer file) I. Title.
QA76.774.S65C35 2013
005.4'32—dc23 2013010583

ISBN-13: 978-0-13-300710-7
ISBN-10: 0-13-300710-3

Text printed in the United States on recycled paper at Edwards Brothers Malloy in Ann Arbor, Michigan.
First printing, June 2013

Editor-in-Chief
Mark L. Taub

Acquisitions Editor
Greg Doench

Development Editor
Chris Zahn

Managing Editor
John Fuller

Project Editor
Elizabeth Ryan

Copy Editor
Stephanie Geels

Indexer
Dick Evans, Infodex

Proofreader
Geneil Breeze

Technical Reviewer
John Philcox

Cover Designer
Alan Clements

Compositor
diacriTech

Thank you to my family for putting up with the time I spend writing. Although I was there physically, I was away most of the time mentally. Writing this book took away valuable family time, and I will try to make it up now that I'm finally finished (writing, that is).

Kids, it's ok to come into my office and play your games again!

This page intentionally left blank

Contents

Preface xix
Acknowledgments xxv
About the Author xxvii

Chapter 1 Installing Oracle Solaris 11 1
Preparation for Installing the OS 1
Oracle Solaris Versions 2
Selecting the Platform 2
Methods of Installation 6
Obtaining the Installation Media 9
Using the Live Media Installer 11
Logging In 22
Using the Interactive Text Installer: x86 23
Using the Interactive Text Installer: SPARC 33
Verifying the Operating Environment 42
Upgrading from a Previous Version of Oracle Solaris 55
Changing an Oracle Solaris Instance 56
Reconfigure an Oracle Solaris Instance 57
Unconfigure an Oracle Solaris Instance 60

Updating the OS 61
Summary 61

Chapter 2 Managing and Updating Software with IPS 63
Understanding IPS 64
Package 64
FMRI 65
Publisher 66
Repository 66
Image 67
BE 68
Distribution Constructor 68
Package Lifecycle 68
Package Dependencies 68
Consolidation 68
Incorporation 70
Authority 70
Installing and Managing Packages 70
Package Management Commands (Command Line) 71
BEs 93
Manage BEs 95
List the BEs 96
Create a New BE 98
Change the Default BE 102
Mount an Inactive BE 105
Rename a BE 106
Uninstall a BE 106
Manage Software Packages in an Inactive BE 107
Update the OS 109
Oracle Solaris Version Names 109
Periodic SRUs 111
Configure a Network-Based Support Repository 112
Configure a Local Support Repository 113
Update Oracle Solaris to a New Release 116
Summary 120

Chapter 3 Boot and Shutdown Procedures for SPARC and x86-Based Systems 121

The Boot Process: SPARC 121

Powering On the System 124

The OpenBoot Environment 124

Accessing the OpenBoot Environment 126

System Console 127

OpenBoot Firmware Tasks 133

The OpenBoot Architecture 134

The OpenBoot Interface 135

PROM Device Tree (Full Device Pathnames) 137

OpenBoot Device Aliases 139

OpenBoot NVRAM 142

OpenBoot Security 149

OpenBoot Diagnostics 151

Booting a SPARC System 155

The `boot` Command 157

Booting from a ZFS Boot Disk 162

Booting an x86-Based System 166

The BIOS 166

GRUB 167

`bootadm` Command 172

The Boot Archive 174

The Kernel 177

The `init` Phase 182

SMF 183

The Service Configuration Repository 186

Service FMRI 189

Service Dependencies 190

SMF Command-Line Administration Utilities 191

Displaying Information about Services 191

Modifying the Service Configuration Repository 197

Starting and Stopping Services Using SMF 200

SMF Message Logging 212

Create SMF Manifest and Profiles 213
    Create New Service Scripts 213
    Create an SMF Manifest: `svcbundle` 222
    Legacy Services 223
System Shutdown 229
    Commands to Shut Down the System 230
    Fastboot 234
    Stopping the System for Recovery Purposes: SPARC 236
    Stopping the System for Recovery Purposes: x86 236
    Turning Off the Power to the Hardware 237
`syslog` 237
    Using the `logger` Command 242
`rsyslog` 242
Summary 244

Chapter 4 Administering Storage Devices 245
Device Drivers 245
Physical Device Name 246
Device Autoconfiguration 250
USB Removable Devices 251
Instance Names 255
Major and Minor Device Numbers 257
Logical Device Names 258
    Block and Raw Devices 262
A File System Defined 263
Defining a Disk's Geometry 264
    Disk Controllers 265
    Defect List 265
    Disk Labels 265
    Partition Tables 267
Disk Slices 267
    Displaying Disk Configuration Information 268
    Using the `format` Utility to Create Slices: SPARC 270
    Using the `format` Utility to Create Slices: x86 281

Administering LUNs 285
Summary 287

Chapter 5 Administering ZFS File Systems 289
Introduction to ZFS 290
ZFS Storage Pools 290
ZFS Is Self-Healing 291
Simplified Administration 292
ZFS Terms 292
ZFS Hardware and Software Requirements 294
ZFS System Process 294
ZFS RAID Configurations 294
Creating a Storage Pool 295
ZFS Help 296
Creating a ZFS File System 298
Using the Whole Disk in a Storage Pool 301
Using a Disk Slice in a Storage Pool 302
Dry Run of Storage Pool and File System Creation 303
Renaming a ZFS File System 304
Listing ZFS File Systems and Storage Pools 304
Displaying ZFS Storage Pool Information 304
Listing ZFS File Systems 309
ZFS Versions 310
ZFS Components 312
Using Disks in a ZFS Storage Pool 313
Using Files in a ZFS Storage Pool 313
Encrypting ZFS File Systems 314
Create an Encrypted ZFS File System 315
Change the `keysource` Property 316
Removing a ZFS File System 317
The `fuser` Command 318
Removing a ZFS Storage Pool 320
Adding Storage Space to a ZFS Storage Pool 322
Mirrored Storage Pools 323

RAID-Z Storage Pools 324
Attaching and Detaching Devices in a Storage Pool 327
Converting a Nonredundant Pool to a Mirrored Pool 327
Adding Redundancy to a Mirrored Pool 328
Detaching a Device from a Mirrored Pool 329
Split a Mirrored Pool 330
Taking Devices in a Storage Pool Offline and Online 332
ZFS History 333
ZFS Properties 334
ZFS File System Properties 335
ZFS Storage Pool Properties 339
Setting ZFS File System Properties 340
Setting ZFS Storage Pool Properties 343
Mounting ZFS File Systems 344
Legacy Mountpoints 347
Sharing ZFS File Systems 349
ZFS Snapshots 349
Creating a ZFS Snapshot 350
Listing ZFS Snapshots 351
Snapshot Disk Space Accounting 352
Saving and Restoring a ZFS Snapshot 353
Destroying a ZFS Snapshot 354
Renaming a ZFS Snapshot 354
Rolling Back a ZFS Snapshot 354
Identifying ZFS Snapshot Differences 356
ZFS Clones 357
Destroying a ZFS Clone 358
Replacing a ZFS File System with a ZFS Clone 358
`zpool scrub` 359
Replacing Devices in a Storage Pool 360
Hot Spares 365
The ZFS Root File System 368
Using an EFI-Formatted Boot Disk: x86 370
Creating Root Pool Snapshots 373

Monitoring ZFS 375
Migrate Legacy File Systems to ZFS 378
Shadow Migration 379
Summary 381

Chapter 6 Administering Zones 383
Oracle Solaris 11 Enhancements 385
Consolidation and Resource Management 385
Consolidation 386
Oracle Solaris Zones 386
Types of Zones 387
Zone States 387
Zone Features 388
Non-global Zone Root File System Models 390
Non-global Zone `zonepath` 391
Networking in a Zone Environment 392
Planning the Zone 393
Zone Service Instances and Daemons 394
Configuring a Zone 395
Create a Zone: `zonecfg` 395
Viewing the Zone Configuration 409
Administering Zones: `zoneadm` 411
List a Zone 412
Verify a Zone 413
Install a Zone 414
Boot a Zone 417
Shut Down a Zone 419
Uninstall a Zone 420
Delete a Zone 421
Log In to a Zone: `zlogin` 421
Log In to the Zone Console 423
Initial Zone Login 423
Log In to a Zone 424
Run a Command in a Zone 424

Using ZFS for Oracle Solaris Zones 425
Adding a ZFS Dataset to a Non-global Zone 425
Delegating a ZFS Dataset to a Non-global Zone 426
Zone Resource Controls 429
Monitor a Zone 430
Create an Exclusive-IP Zone 433
Modify an Existing Zone 438
Move a Zone 439
Migrate a Zone 440
Clone a Zone 444
Create a Shared-IP Zone 447
Migrate an Oracle Solaris 10 Global Zone to Oracle Solaris 11 (P2V) 450
Prepare an Oracle Solaris 10 Global Zone for Migration 450
Create a `solaris10` Brand Zone 450
Back Up a Zone 453
Recover a Zone 454
Delegated Zone Administration 455
Summary 455

Chapter 7 User and Security Administration 457
Administering User Accounts 457
Managing User and Group Accounts from the Command Line 460
Adding User Accounts from the Command Line 460
Modifying User Accounts from the Command Line 465
Deleting User Accounts from the Command Line 466
Adding Group Accounts from the Command Line 466
Modifying Group Accounts from the Command Line 467
Deleting Group Accounts from the Command Line 468
Setting Up Shell Initialization Files 469
C Shell Initialization Files 469
Bourne Shell Initialization Files 470

Korn Shell Initialization Files 470
Bash Shell Initialization Files 470
Default Initialization Files 471
Customizing User Initialization Files 472
The Home Directory 475
Projects 476
Name Services 477
System Security 477
Controlling Physical Security 478
Controlling System Access 478
Password Encryption 481
Where User Account Information Is Stored 482
Restricted Shells 486
RBAC 487
Create a Role Using Authorizations 487
Configuring RBAC from the Command Line 489
Create a Role Using a Profile 493
Delegating Zone Management 500
Create a Custom Rights Profile 502
Controlling File Access 503
Effective UIDs and GIDs 505
The Default User Mask 506
ACLs 507
Setting the Correct Path 512
The `setuid` and `setgid` Programs 513
Auditing Users 515
Monitoring Users and System Usage 515
Checking Who Is Logged In 517
Controlling Network Security 522
Securing Network Services 522
The `/etc/default/login` File 525
Trusted Hosts 526

Restricting FTP 527
Securing Superuser Access 531
The Secure Shell (`ssh`) 533
Common-Sense Security Techniques 538
Summary 539

Chapter 8 Managing System Processes 541
`ps` 543
`pgrep` 547
`prstat` 549
`top` 556
`mpstat` 557
`ptree` 559
`time` 559
`svcs` 560
Proc Tools and `/proc` 560
Process Types 562
Using Signals 565
Scheduling Processes 569
Scheduling Priorities 569
Changing the Priority of a Time-Sharing Process with `nice` 570
Process Scheduler 572
Using the Oracle Solaris Batch-Processing Facility 575
Configure `crontab` 575
Scheduling a Single System Event (`at`) 578
Configure System Core Files 581
Configure System Crash Dumps 584
Summary 587

Chapter 9 The Oracle Solaris Network Environment 589
Client/Server Model 589
Hosts 590
Oracle Solaris TCP/IP 590
Network Interface 592

Fixed Network Configuration 592
Identifying the Physical Network Interface 594
Data Link 595
`ipadm`: Administer the IP Interface 598
Reactive Network Configuration 606
Network Configuration Profiles 607
Reactive Network Services 607
Disable the Reactive Network Configuration 608
Enable the Reactive Network Configuration 609
Configuring Additional Network Services 609
Configure and Activate DNS 610
Configure the Name Service Switch 611
Configure the Default Route 612
Reset the Name Service Switch 613
The System Hostname 613
Virtual Networking 615
Configure a Simple Virtual Network 615
The etherstub 617
Create a Private Virtual Network between Zones 618
Checking the Network 621
Network Services 624
RPC Services 627
Summary 629

Chapter 10 Network File Systems 631
NFS Overview 631
NFS Version 4 632
Oracle Solaris 11 Enhancements to NFS v4 633
Servers and Clients 633
NFS Daemons 633
NFS Services 633
Sharing a File System 635
Sharing a ZFS File System: Oracle Solaris 11 11/11 636
Sharing a ZFS File System: Oracle Solaris 11.1 640

ZFS Sharing within a Non-Global Zone 643
Mounting a Remote File System 644
Changing the NFS Version 649
AutoFS 650
AutoFS Maps 653
When to Use `automount` 665
Mirror Mounts 666
Troubleshooting NFS Errors 667
The Stale NFS File Handle Message 667
The RPC: Program Not Registered Error 667
The NFS: Service Not Responding Error 668
The Server Not Responding Error 669
The RPPC: Unknown Host Error 669
The No Such File or Directory Error 669
Summary 669

Index 671

Preface

This book covers all of the topics necessary to effectively install and administer an Oracle Solaris 11 system. When used as a study guide, this book will save you a great deal of time and effort searching for information you will need to administer Oracle Solaris 11 on SPARC and x86-based systems. This book covers each topic in enough detail for inexperienced administrators to learn about Oracle Solaris 11 and apply the knowledge to real-life scenarios. Experienced readers will find the material complete and concise, making it a valuable reference for everyday tasks. Drawing from my years of experience as a senior Oracle Solaris administrator and consultant, you'll find new ideas and see some different approaches to system administration that probably were not covered in your Oracle Solaris administration training courses.

You might be familiar with my Oracle Solaris certification training guides and exam prep books that have been published over the past 15 years. I am a Solaris Subject Matter Expert for Oracle. I participate in the development of the Oracle Solaris 11 certification exams, and I have made certain that the topics covered on the exam are covered in this book. But, rather than stopping at simply an exam prep book, I have gone to great effort to ensure that all tasks that you need to perform are covered beyond what is simply required to pass the exam.

Experienced administrators will find many welcome enhancements in Oracle Solaris 11, but they will also find that Oracle Solaris 11 is vastly different from previous versions of the operating system. Oracle has added new features and, in some cases, has completely redesigned some of the functions you may have used on a daily basis. To eliminate the frustration of learning a new environment, I describe new solutions for performing familiar tasks.

Welcome to the Oracle Solaris 11 community, and don't forget to visit my blog at `http://www.unixed.com/blog` where I answer your questions and discuss various topics related to Oracle Solaris administration.

What's New in Oracle Solaris 11

Oracle Solaris 11 was first introduced in November 2010 as Oracle Solaris 11 Express. With Oracle Solaris 11 Express, Oracle provided a fully tested, fully supported, production-ready package that allowed customers to have access to the latest technology and hundreds of new features. Most administrators used Oracle Solaris 11 Express to test the new features and to provide feedback to the manufacturer.

The first production release of Oracle Solaris 11 was made available in November 2011 and is referred to as Oracle Solaris 11 11/11. In October 2012, Oracle introduced Oracle Solaris 11.1, the first major release of Oracle Solaris 11. This book covers features found in both versions of Oracle Solaris 11. For those of you currently running Oracle Solaris 10, you may be wondering, "What are the real benefits of moving to Oracle Solaris 11?" Throughout this book, I provide unbiased coverage of each feature to help you make an informed decision on the benefits of moving to Oracle Solaris 11.

Chapter 1 describes how to install the Oracle Solaris 11 operating system using the Live Media and text installers, and I guide you through each step of the installation process. During the installation, I help you make decisions that will affect the final installation and help you understand which configuration options may or may not be recommended for a secure production environment.

Chapter 2 discusses the new Image Packaging System (IPS). This new method of installing and updating software packages eliminates the traditional software package and patching commands that many administrators have used for years. IPS can be a complex and confusing topic, and I have created many real-life examples to guide you through the complete software lifecycle, including creating and maintaining an up-to-date IPS repository, installing and managing software packages, and using boot environments when updating the operating system.

Chapter 3 details the entire boot process on both the SPARC and x86 platforms. Where many textbooks skip the hardware-level discussion, I cover everything from powering on the server to accessing the console through the ALOM and ILOM interface and monitoring hardware faults through the fault management architecture. I describe the OpenBoot and GRUB2 environments in detail. The entire boot process, including managing multiple boot environments (BEs), booting a BE, loading the kernel, specifying run levels and milestones, and starting the Service Management Facility (SMF), is described in detail. I am aware that most problems are encountered during the boot process, and I have made an effort to describe every scenario I have encountered on both the SPARC and x86 platforms. You'll also gain a complete understanding of kernel tunables, SMF, and services. You'll learn how to create and manage custom SMF profiles and manifests. You'll learn about managing system messages using the new `rsyslog` feature and the legacy `syslog` facility.

Chapter 4 describes the hardware components and begins with attaching new hardware, identifying hardware components, configuring device drivers, and configuring and formatting storage devices.

ZFS storage pools and file systems are the topic of Chapter 5. You'll find more information on ZFS in this chapter than in entire books that have been published on the topic. I describe key concepts in creating redundant and nonredundant storage pools. I go on to describe how to create ZFS file systems, manage properties to control file system characteristics, encrypt file systems, and back up and restore ZFS file systems. I use examples to describe problems and failures with storage components and finally ZFS monitoring, troubleshooting, and recovery techniques that I have used in production environments.

Chapter 6 discusses the virtualization environment, Oracle Solaris zones. You'll be guided through the creation of the different types of non-global and immutable zones. You'll learn how to boot and manage zones, set resources on a zone, delegate ZFS storage, and monitor, clone, back up, and recover zones.

In Chapter 7, I describe techniques used to harden an Oracle Solaris system beginning with securing and monitoring user accounts, controlling system and file access, delegating administrative tasks, and controlling network security.

Chapter 8 describes how to monitor and manage system processes on a multiprocessor system. You'll also learn how to manage and configure core files and crash dumps.

Chapter 9 is a discussion of the Oracle Solaris 11 network environment. Readers with experience on previous versions of Oracle Solaris will appreciate the detail and the step-by-step examples provided in the explanation of the complex topic of virtual networking. I describe how to configure reactive and fixed network configurations. I outline the new methods used to configure network parameters and services. I illustrate how to create a virtual network between non-global zones. And I describe network monitoring and troubleshooting techniques that I use in production environments.

Chapter 10 describes Network File Systems (NFS) and begins with a thorough explanation of NFS and where NFS is used in a production environment. Again, for those with experience on previous versions of Oracle Solaris, I detail how to configure NFS in Oracle Solaris 11. Using step-by-step examples, I explain the new process of sharing file systems on Oracle Solaris 11/11 and how it changed again in Oracle Solaris 11.1. You'll learn how to mount and manage NFS file systems manually and using AutoFS. I conclude the chapter with techniques I use for monitoring and troubleshooting NFS.

Enjoy the book, keep it close by, and may the material in this book help you better your skills, enhance your career, and achieve your goals.

Conventions Used in This Book

Commands: In the steps and examples, the commands you type are displayed in a special monospaced bold font.

```
# ls -l<cr>
```

`<cr>` indicates pressing the Enter/Return key.

The use of the hash symbol (#), dollar sign ($) or `ok`, when prefixed to the command, indicates the system prompt as shown in the examples:

```
ok boot<cr>
# ls -l<cr>
$ ls -l<cr>
```

The prompt is not to be typed in as part of the command syntax.

Arguments and Options—In command syntax, command options and arguments are enclosed in < >. (The italicized words within the < > symbols stand for what you will actually type. Don't type the "< >.")

```
# ls -l <directoryname><cr>
```

Code Continuation Character—When a line of code is too long to fit on one line, it is broken and continued to the next line. The continuation is preceded by a backslash (\), for example:

```
# useradd -u 3000 -g other -d /export/home/bcalkins -m -s /usr/bin/bash \
-c "Bill Calkins, ext. 2345" bcalkins<cr>
```

The backslash is not to be typed as part of the command syntax. Type the command as one continuous line. When the text gets to the end of the line, it will automatically wrap to the next line.

Commands are case sensitive in Oracle Solaris, so make sure you use upper and lowercase as specified. Make sure that you use spaces, hyphens (-), double quotes (") and single quotes (') exactly as indicated in the examples that are provided.

Audience

This book is designed for anyone who has a basic understanding of UNIX or Linux and wants to learn more about administering an Oracle Solaris 11 system. Whether or not you plan to become certified, this book is the starting point to becoming an Oracle Solaris system administrator. It contains the same training material that I use in my Oracle Solaris System Administration classes. This book covers the basic as well as the advanced system administration topics you need to know before you begin administering the Oracle Solaris 11 operating system. The goal was to present the material in an easy-to-follow format, with text that is easy to read and understand. The only prerequisite is that you have used UNIX or Linux, you have attended a fundamental UNIX or Linux class for users, or you have studied equivalent material so that you understand basic UNIX commands and syntax. Before you begin administering Oracle Solaris, it's important that you have actually used UNIX or Linux.

This book is also intended for experienced system administrators who want to become certified, update their current Oracle Solaris certification, or simply learn about the Oracle Solaris 11 operating environment.

This page intentionally left blank

Acknowledgments

I would like to thank Greg Doench and Prentice Hall for giving me the opportunity to write another book on Oracle Solaris. I appreciate that you recognize the value in having a single author write an entire technical book to keep the material consistent and smooth. I also thank them for letting me take my time to get it right. Oracle Solaris 11 has gone through many changes since the initial release. At the last minute, I wanted to make sure that this book was up to date with Oracle Solaris 11.1, which meant delaying the release of this book from its original publication date.

I also want to thank John Philcox once again for coming on board as a technical editor. John has worked on most of my books, and we are of the same mind when it comes to publishing an Oracle Solaris textbook. I look forward to working with you again, John.

I also want to thank Oracle for allowing me to be part of the certification team and actively involved in the creation of the Oracle Solaris 11 certification exams. Oracle is very particular with the contractors with whom they work, and I feel privileged that they consider me as part of their team of experts.

That's it! I don't want to bore you with a long list of names when it's John, myself, and the team of editors at Prentice Hall who put this book together. A small, tight group is an efficient and successful group, and I would like to extend my sincere thanks to all of you who edited the text, laid out the pages, and shipped the books. My efforts would be lost if it weren't for your hard work.

This page intentionally left blank

About the Author

About the Author

Bill Calkins is an Oracle Solaris 11 Certified Professional, works as a Solaris Subject Matter Expert for Oracle Corporation, and participates in the development of the Oracle Solaris 11 certification exams. Bill is the owner and president of UnixEd and Pyramid Consulting Inc., IT training and consulting firms located near Grand Rapids, Michigan, specializing in the implementation and administration of UNIX, Linux, and Microsoft Windows–based systems. He has more than twenty years of experience in UNIX system administration, consulting, and training at more than 300 companies. Bill is known worldwide for the Oracle Solaris textbooks he's authored, which have always been best sellers and used by universities and training organizations worldwide. It began with the *Oracle Solaris 2.6 Training Guide* in 1999 and subsequent books for Oracle Solaris 7, 8, 9, and 10.

Many of you have written with your success stories, suggestions, and comments. Your suggestions are always welcome and are what keep making these textbooks better with each new release. Drop me a note at wcalkins@unixed.com, introduce yourself, and tell me what you think.

About the Technical Editor

John Philcox currently works as a Technical Specialist at Atos IT Services UK. He is based in Cheltenham, Gloucestershire, in the United Kingdom and specializes in

UNIX systems and networks. He has more than 30 years' experience in IT, 25 of those with the SunOS and Oracle Solaris environments. He is a certified Oracle Solaris system and network administrator and has worked in a number of large, multi-vendor networks in both the public and private sectors of business.

John was the author of *Solaris System Management*, published by New Riders, and *Solaris 9 Network Administrator Certification (Exam Cram 2),* published by QUE Publishing. John has acted as technical editor or contributing author to most of Bill Calkins' books.

4

Administering Storage Devices

It's important that you understand how Oracle Solaris views the disk drives and various other hardware components on your system. In particular, you need to understand how the storage devices are configured and named before you can create a file system on them or install the Oracle Solaris operating environment.

Device management in the Oracle Solaris 11 environment includes adding and removing from system peripheral devices such as tape drives, printers, and disk drives. Device management sometimes also involves adding a third-party device driver to support a device if the device driver is not available in Oracle's distribution of the Oracle Solaris operating environment.

System administrators need to know how to specify device names when using commands to manage disks, file systems, and other devices. This chapter describes disk device management in detail. It also describes disk device naming conventions as well as adding, configuring, and displaying information about disk devices attached to your system.

Device Drivers

A computer typically uses a wide range of peripheral and mass-storage devices such as a serial attached SCSI disk drive, a keyboard, a mouse, and some kind of magnetic backup medium. Other commonly used devices include CD/DVD-ROM drives, printers, and various USB devices. Oracle Solaris communicates with peripheral devices through device files or drivers. A "device driver" is a low-level program that allows the kernel to communicate with a specific piece of hardware. The driver serves as the OS's "interpreter" for that piece of hardware. Before Oracle Solaris can communicate with a device, the device must have a device driver.

When a system is started for the first time, the kernel creates a device hierarchy to represent all of the devices connected to the system. This is the autoconfiguration process, which is described later in this chapter. If a driver is not loaded for a particular peripheral device, that device is not functional. In Oracle Solaris, each disk device is described in three ways, using three distinct naming conventions:

- Physical device name: Represents the full device pathname in the device information hierarchy
- Instance name: Represents the kernel's abbreviation name for every possible device on the system
- Logical device name: Used by system administrators with most file system commands to refer to devices

System administrators need to understand these device names when using commands to manage disks and file systems. We discuss these device names throughout this chapter.

Physical Device Name

Before the OS is loaded, the system locates a particular device through the device tree, also called the full device pathname. Full device pathnames are described in the "PROM Device Tree (Full Device Pathnames)" section of Chapter 3, "Boot and Shutdown Procedures for SPARC and x86-Based Systems." After the kernel is loaded, however, a device is located by its physical device pathname. Physical device names represent the full device pathname for a device. Note that the two names have the same structure. For example, the full device pathname for a SCSI disk at target 0 on a SunFire T2000 system is as follows:

```
/pci@780/pci@0/pci@9/scsi@0/disk@0
```

SAS Disk Drives

SCSI drives come in two types: parallel and serial. The serial attached SCSI (SAS) drive delivers better performance than its parallel predecessor. Both drives show up as SCSI drives in the OS.

Intel-based systems commonly use IDE or Serial AT Attachment (SATA) disk drives. On the x86 platform, the SATA disk (target 0) looks like this:

```
/pci@0,0/pci8086,2829@d/disk@0,0
```

Now let's look at the corresponding physical device name from the OS level. Use the `dmesg` command, described later in this section, to obtain information about devices connected to your system. By viewing information displayed by the `dmesg` command, you'll receive the following information about the SunFire T2000's SAS disk 0:

```
# dmesg |grep scsi<cr>
Jan 23 16:02:42 server rootnex: [ID 349649 kern.info] scsi_vhci0 at root
Jan 23 16:02:42 server genunix: [ID 936769 kern.info] scsi_vhci0 is /scsi_vhci
Jan 23 16:02:45 server scsi: [ID 583861 kern.info] sd0 at ahci0: target 0 lun 0
Jan 23 16:02:45 server scsi: [ID 583861 kern.info] sd1 at ahci0: target 1 lun 0
Jan 23 16:03:04 server rootnex: [ID 349649 kern.info] iscsi0 at root
Jan 23 16:03:04 server genunix: [ID 936769 kern.info] iscsi0 is /iscsi
```

This same information is also available in the `/var/adm/messages` file.

As you can see, the physical device name listed above and the full device name seen at the OpenBoot PROM are the same. The difference is that the full device pathname is simply a path to a particular device. The physical device is the actual driver used by Oracle Solaris to access that device from the OS.

Physical device files are found in the `/devices` directory. The content of the `/devices` directory is controlled by the `devfs` file system. The entries in the `/devices` directory dynamically represent the current state of accessible devices in the kernel and require no administration. New device entries are added when the devices are detected and added to the kernel. The physical device files for SAS disks 0 and 1 connected to the primary SCSI controller would be

```
/devices/pci@780/pci@0/pci@9/scsi@0/sd@0,0:<#>
/devices/pci@780/pci@0/pci@9/scsi@0/sd@1,0:<#>
```

for the block device and

```
/devices/pci@780/pci@0/pci@9/scsi@0/sd@0,0:<#>,raw
/devices/pci@780/pci@0/pci@9/scsi@0/sd@1,0:<#>,raw
```

for the character (raw) device, where `<#>` is a letter representing the disk slice. Block and character devices are described later in this chapter in the section titled "Block and Raw Devices."

Table 4-1 Device Information Commands

Command	Description
`prtconf`	The `prtconf` command displays system configuration information, including the total amount of memory and the device configuration, as described by the system's hierarchy. This useful tool verifies whether a device has been seen by the system.
`sysdef`	The `sysdef` command displays device configuration information, including system hardware, pseudo devices, loadable modules, and selected kernel parameters.
`dmesg`	The `dmesg` command displays system diagnostic messages as well as a list of devices attached to the system since the most recent restart.
`format`	The `format` command displays both physical and logical device names for all available disks.

The system commands used to provide information about physical devices are described in Table 4-1.

`prtconf` Output

The output produced by the `prtconf` command can vary depending on the version of the system's PROM.

Type the `prtconf` command:

```
# prtconf<cr>
System Configuration: Oracle Corporation sun4v
Memory size: 3968 Megabytes
System Peripherals (Software Nodes):

SUNW,Sun-Fire-T200
    scsi_vhci, instance #0
    packages (driver not attached)
        SUNW,builtin-drivers (driver not attached)
        deblocker (driver not attached)
        disk-label (driver not attached)
  <Output has been truncated.>
```

Use the `-v` option to display detailed information about devices.

The `sysdef` command can also be used to list information about hardware devices, pseudo devices, system devices, loadable modules, and selected kernel tunable parameters as follows:

```
# sysdef<cr>
*
* Hostid
```

```
*
  8510e818
*
* sun4v Configuration
*
*
* Devices
*
scsi_vhci, instance #0
packages (driver not attached)
        SUNW,builtin-drivers (driver not attached)
        deblocker (driver not attached)
        disk-label (driver not attached)
        terminal-emulator (driver not attached)
        dropins (driver not attached)
        SUNW,asr (driver not attached)
        kbd-translator (driver not attached)
        obp-tftp (driver not attached)
        zfs-file-system (driver not attached)

*Output has been truncated.
* System Configuration
*
  swap files
swapfile              dev    swaplo  blocks  free
/dev/zvol/dsk/rpool/swap 228,2       16 2097136 2097136
*
* Tunable Parameters
*
81469440        maximum memory allowed in buffer cache (bufhwm)
   30000        maximum number of processes (v.v_proc)
      99        maximum global priority in sys class (MAXCLSYSPRI)
   29995        maximum processes per user id (v.v_maxup)
      30        auto update time limit in seconds (NAUTOUP)
      25        page stealing low water mark (GPGSLO)
       1        fsflush run rate (FSFLUSHR)
      25        minimum resident memory for avoiding deadlock (MINARMEM)
*Output has been truncated
```

Use the output of the `prtconf` command to identify which disk, tape, and CD/DVD-ROM devices are connected to the system. As shown in the preceding `prtconf` and `sysdef` examples, some devices display the `driver not attached` message next to the device instance. This message does not always mean that a driver is unavailable for this device. It means that no driver is currently attached to the device instance because there is no device at this node or the device is not in use. The OS automatically loads drivers when the device is accessed, and it unloads them when it is not in use.

The system determines which devices are attached to it at startup. This is why it is important to have all peripheral devices powered on at startup, even if they are not currently being used. During startup, the kernel configures itself dynamically, loading needed modules into memory. Device drivers are loaded when devices, such as disk and tape devices, are accessed for the first time. This process is called "auto-configuration" because all kernel modules are loaded automatically if needed. As described in Chapter 3, the system administrator can customize the way in which kernel modules are loaded by modifying the `/etc/system` file.

Device Autoconfiguration

Autoconfiguration offers many advantages over the manual configuration method used in earlier versions of SunOS, in which device drivers were manually added to the kernel, the kernel was recompiled, and the system had to be restarted. Now, with autoconfiguration, the administrator simply connects the new device to the system and performs a reconfiguration startup. To perform a reconfiguration startup, follow these steps:

1. Create the `/reconfigure` file with the following command:

   ```
   # touch /reconfigure<cr>
   ```

   The `/reconfigure` file causes the Oracle Solaris software to check for the presence of any newly installed devices the next time you turn on or start up your system.
2. Shut down the system using the shutdown procedure described in Chapter 3.

   If you need to connect the device, turn off power to the system and all peripherals after Oracle Solaris has been properly shut down.
3. After the new device is connected, restore power to the peripherals first and then to the system. Verify that the peripheral device has been added by attempting to access it.

Automatic Removal of /reconfigure

The file named `/reconfigure` automatically gets removed during the bootup process.

An optional method of performing a reconfiguration startup is to type “boot -r” at the OpenBoot prompt.

On an x86-based system, perform a reconfiguration reboot by editing the `boot` command in the GRUB menu as described in Chapter 3.

Specify a Reconfiguration Reboot

As root, you can also issue the `reboot -- -r` command from the shell prompt. The `-- -r` passes the `-r` to the `boot` command.

During a reconfiguration restart, a device hierarchy is created in the `/devices` file system to represent the devices connected to the system. The kernel uses this to associate drivers with their appropriate devices.

Autoconfiguration offers the following benefits:

- Main memory is used more efficiently because modules are loaded as needed.
- There is no need to reconfigure the kernel if new devices are added to the system. When you add devices such as disks or tape drives other than USB and hot-pluggable devices, the system needs to be shut down before you connect the hardware so that no damage is done to the electrical components.
- Drivers can be loaded and tested without having to rebuild the kernel and restart the system.

devfsadm

Another option used to automatically configure devices on systems that must remain running at all times, and one that does not require a reboot, is the `devfsadm` command.

Occasionally, you might install a new device for which Oracle Solaris does not have a supporting device driver. Always check with the manufacturer to make sure any device you plan to add to your system has a supported device driver. If a driver is not included with the standard Oracle Solaris release, the manufacturer should provide the software needed for the device to be properly installed, maintained, and administered.

USB Removable Devices

USB devices were developed to provide a method to attach peripheral devices such as keyboards, printers, cameras, and disk drives using a common connector and interface. Furthermore, USB devices are "hot-pluggable," which means they can be connected or disconnected while the system is running. The OS automatically detects when a USB device has been connected and automatically configures the operating environment to make it available.

The Oracle Solaris 11 operating environment supports USB devices. When hot-plugging a USB device, the device is immediately displayed in the device hierarchy. For example, a full device pathname for a USB thumb drive connected to a SunFire T2000 system would appear as follows:

```
/devices/pci@7c0/pci@0/pci@1/pci@0/usb@6/hub@1/storage@2/disk@0,0
```

A printer would look like this:

```
/pci@1f,4000/usb@5/hub@3/printer@1
```

The steps to add a USB mass storage device are as follows:

1. Insert a USB thumb drive into the USB port on your server. For this example, the device already contains a file system.
2. Verify that the USB device is mounted by entering the `rmformat` command as follows:

```
# rmformat<cr>
Two devices are listed as follows:
Looking for devices ...
     1. Logical Node: /dev/rdsk/c3t0d0s2
        Physical Node: /pci@7c0/pci@0/pci@1/pci@0/ide@8/sd@0,0
        Connected Device: TEAC  DW-224SL-R    1.0B
        Device Type: DVD Reader
        Bus: IDE
        Size: 525.9 MB
        Label: <None>
        Access permissions: Medium is not write protected.
     2. Logical Node: /dev/rdsk/c5t0d0s2
        Physical Node: /pci@7c0/pci@0/pci@1/pci@0/usb@6/hub@1/storage@2/disk@0,0
        Connected Device: Generic Flash Disk    8.07
        Device Type: Removable
        Bus: USB
        Size: 981.0 MB
        Label: <None>
        Access permissions: Medium is not write protected.
```

Note

If the device does not contain a file system, use the `rmformat` command to format the device. The `rmformat` command is described later in this section.

The first device listed is the removable DVD, and the second device listed is the removable USB thumb drive.

3. Verify that the device has been automatically mounted by typing

```
# mount<cr>
```

The mounted device is displayed as follows:

```
/media/NO NAME on /dev/dsk/c5t0d0s2:1 read/write/nosetuid/nodevices/rstchown/
hidden/nofoldcase/clamptime/noatime/timezone=18000/owner=0/group=0/mask=077/
dev=32c102a on Thu Jan 24 12:09:11 2013
```

4. The nickname for the mounted device can also be listed by typing

```
# rmmount -l<cr>
/dev/dsk/c5t0d0s2:1 rmdisk,rmdisk0,THUMBDRIVE,/media/THUMBDRIVE
/dev/dsk/c3t0d0s2  cdrom,cdrom0,cd,cd0,sr,sr0,Oracle_Solaris-11_1-Text-\
SPARC,/media/Oracle_Solaris-11_1-Text-SPARC
```

Notice the path to each device. Access the removable DVD media through this path:

```
/media/Oracle_Solaris-11_1-Text-SPARC
```

Access the USB thumb drive media through this path:

```
/media/THUMBDRIVE
```

The `rmformat` command is used to format, list, eject, partition, and protect removable rewritable media. If the USB device already has a file system, the device is automatically mounted. To unmount the device, type

```
# rmmount -u /dev/dsk/c5t0d0s2<cr>
/dev/dsk/c5t0d0s2 unmounted
```

To format the device, type

```
# rmformat -F quick /dev/rdsk/c5t0d0s2<cr>
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y<cr>
```

The `rmformat` command has three formatting options:

1. `quick`: This option formats the media without certification or with limited certification of certain tracks on the media.
2. `long`: This option completely formats the media.
3. `force`: This option formats completely without user confirmation.

Create a file system on the device as follows:

```
# mkfs -F pcfs -o nofdisk,size=9800 /dev/rdsk/ c5t0d0s2<cr>
```

The `mkfs` command constructs a file system on a raw device. I specified `-F pcfs` to create a file allocation table (FAT) file system.

Be careful when removing USB devices. If the device is being used when it is disconnected, you will get I/O errors and possible data errors. When this happens, you'll need to plug the device back in, stop the application that is using the device, and then unplug the device.

USB mass storage devices and DVD-ROMs can be inserted and automatically mounted by using the removable media services. These services are started by default and can be enabled or disabled as follows.

To prevent removable volumes from automatically mounting, disable the `rmvolmgr` service as follows:

```
# svcadm disable rmvolmgr<cr>
```

To disable all of the volume management media services, disable the `dbus`, `hal`, and `rmvolmgr` services as follows:

```
# svcadm disable rmvolmgr<cr>
# svcadm disable dbus<cr>
# svcadm disable hal<cr>
```

Disabling the volume management services means that you would have to mount all media manually using the `mount` command.

Enable removable media services:

```
# svcadm enable rmvolmgr<cr>
# svcadm enable dbus<cr>
# svcadm enable hal<cr>
```

When disconnecting a USB device such as a USB thumb drive, eject the device as follows:

1. List the removable devices as follows:

```
# rmmount -l<cr>
/dev/dsk/c3t0d0s2 cdrom,cdrom0,cd,cd0,sr,sr0,Oracle_Solaris_Text_SPARC,\
/media/Oracle_Solaris_Text_SPARC
/dev/dsk/c5t0d0s2:1 rmdisk,rmdisk0,NO NAME the devices currently mounted:
```

2. Unmount the device as follows:

```
# rmmount -u rmdisk0<cr>
```

3. Eject the device as follows:

```
# eject rmdisk0<cr>
```

Instance Names

The instance name represents the kernel's abbreviated name for every possible device on the system. A few examples of instance names are

- `sd0`: The instance name for a SCSI disk
- `e1000g`: The instance name for a type of network interface

Instance names are mapped to a physical device name in the `/etc/path_to_inst` file. The following shows the contents of a `path_to_inst` file:

```
# more /etc/path_to_inst<cr>
#
#   Caution! This file contains critical kernel state
#
#
#   Caution! This file contains critical kernel state
#
"/fcoe" 0 "fcoe"
"/iscsi" 0 "iscsi"
"/pseudo" 0 "pseudo"
"/scsi_vhci" 0 "scsi_vhci"
"/options" 0 "options"
"/pci@780" 0 "px"
"/pci@780/pci@0" 0 "pcieb"
"/pci@780/pci@0/pci@1" 1 "pcieb"
"/pci@780/pci@0/pci@1/network@0" 0 "e1000g"
"/pci@780/pci@0/pci@1/network@0,1" 1 "e1000g"
"/pci@780/pci@0/pci@2" 2 "pcieb"
"/pci@780/pci@0/pci@8" 3 "pcieb"
"/pci@780/pci@0/pci@9" 4 "pcieb"
"/pci@780/pci@0/pci@9/scsi@0" 0 "mpt"
"/pci@780/pci@0/pci@9/scsi@0/sd@0,0" 2 "sd"
"/pci@780/pci@0/pci@9/scsi@0/sd@1,0" 3 "sd"
 ... <output has been trunctated> ...
```

Although instance names can be displayed using the commands `dmesg`, `sysdef`, and `prtconf`, the only command that shows the mapping of the instance name to the physical device name is the `dmesg` command. For example, you can determine the mapping of an instance name to a physical device name by looking at the `dmesg` output, as shown in the following example from a T2000 SPARC system:

```
Jan 24 12:31:02 solaris genunix: [ID 936769 kern.info] sd2 is\
 /pci@780/pci@0/pci@9/scsi@0/sd@0,0
Jan 24 12:32:03 solaris genunix: [ID 936769 kern.info] sd3 is\
 /pci@780/pci@0/pci@9/scsi@0/sd@1,0
```

In the first example, sd2 is the instance name and /pci@780/pci@0/pci@9/scsi@0/sd@0,0 is the physical device name. In the second example, sd3 is the instance name and /pci@780/pci@0/pci@9/scsi@0/sd@1,0 is the physical device name. After the instance name has been assigned to a device, it remains mapped to that device. To keep instance numbers consistent across restarts, the system records them in the /etc/path_to_inst file. This file is only read at startup, and it is updated by the devfsadmd daemon described later in this section.

Devices already existing on a system are not rearranged when new devices are added, even if new devices are added to pci slots that are numerically lower than those occupied by existing devices. In other words, the /etc/path_to_inst file is appended to, not rewritten, when new devices are added.

It is generally not necessary for the system administrator to change the path_to_inst file because the system maintains it. The system administrator can, however, change the assignment of instance numbers by editing this file and doing a reconfiguration startup. However, any changes made in this file are lost if the devfsadm command is run before the system is restarted.

Resolving Problems with /etc/path_to_inst

If you can't start up from the startup disk because of a problem with the /etc/path_to_inst file, you should start up from the CD/DVD (boot cdrom -s) and remove the /etc/path_to_inst file from the startup disk. To do this, start up from the DVD using boot cdrom -s at the OpenBoot prompt. Use the rm command to remove the file named /a/etc/path_to_inst. The path_to_inst file will automatically be created the next time the system boots.

You can add new devices to a system without requiring a reboot. It's all handled by the devfsadmd daemon that transparently builds the necessary configuration entries for those devices capable of notifying the kernel when the device is added (such as USB, FC-AL, disks, and so on). An example of when to use the devfsadm command would be if the system had been started but the power to the tape drive was not turned on. During startup, the system did not detect the device; therefore, its drivers were not installed.

To gain access to the device, you could halt the system, turn on power to the tape drive, and start the system back up, or you could simply turn on power to the tape drive and issue the following command at the command prompt:

```
# devfsadm<cr>
```

When used without any options, devfsadm will attempt to load every driver in the system and attach each driver to its respective device instances. You can restrict devfsadm to only look at specific devices using the -c option as follows:

```
# devfsadm -c tape<cr>
```

This restricts the `devfsadm` command to devices of class *tape*.

You can also use the `devfsadm` command to configure only the devices for a specific driver such as “st” by using the `-i` option as follows:

```
# devfsadm -i st<cr>
```

The `devfsadm` command will only configure the devices for the driver named “st.”

Major and Minor Device Numbers

Each device has a major and minor device number assigned to it. These numbers identify the proper device location and device driver to the kernel. This number is used by the OS to key into the proper device driver whenever a physical device file corresponding to one of the devices it manages is opened. The major device number maps to a device driver such as sd, st, or e1000g. The minor device number indicates the specific member within that class of devices. All devices managed by a given device driver contain a unique minor number. Some drivers of pseudo devices (software entities set up to look like devices) create new minor devices on demand. Together, the major and minor numbers uniquely define a device and its device driver.

Physical device files have a unique output when listed with the `ls -l` command, as shown in the following example:

```
# cd /devices/pci@780/pci@0/pci@9/scsi@0<cr>
# ls -l<cr>
```

The system responds with the following:

```
total 4
drwxr-xr-x   2 root     sys          2 Jan 24 13:25 sd@0,0
brw-r-----   1 root     sys    203, 16 Jan 24 12:32 sd@0,0:a
crw-r-----   1 root     sys    203, 16 Jan 24 12:32 sd@0,0:a,raw
brw-r-----   1 root     sys    203, 17 Jan 24 12:32 sd@0,0:b
crw-r-----   1 root     sys    203, 17 Jan 24 12:32 sd@0,0:b,raw
brw-r-----   1 root     sys    203, 18 Jan 24 12:32 sd@0,0:c
crw-r-----   1 root     sys    203, 18 Jan 24 12:32 sd@0,0:c,raw
brw-r-----   1 root     sys    203, 19 Jan 24 12:32 sd@0,0:d
crw-r-----   1 root     sys    203, 19 Jan 24 12:32 sd@0,0:d,raw
brw-r-----   1 root     sys    203, 20 Jan 24 12:32 sd@0,0:e
crw-r-----   1 root     sys    203, 20 Jan 24 12:32 sd@0,0:e,raw
```

continues

```
brw-r-----   1 root     sys    203, 21 Jan 24 12:32 sd@0,0:f
crw-r-----   1 root     sys    203, 21 Jan 24 12:32 sd@0,0:f,raw
brw-r-----   1 root     sys    203, 22 Jan 24 12:32 sd@0,0:g
crw-r-----   1 root     sys    203, 22 Jan 24 12:32 sd@0,0:g,raw
brw-r-----   1 root     sys    203, 23 Jan 24 12:32 sd@0,0:h
crw-r-----   1 root     sys    203, 23 Jan 24 12:32 sd@0,0:h,raw ...
<output has been truncated> ...
```

This long listing includes columns showing major and minor numbers for each device. The sd driver manages all of the devices listed in the previous example that have a major number of 203. Minor numbers are listed after the comma.

During the process of building the /devices directory, major numbers are assigned based on the kernel module attached to the device. Each device is assigned a major device number by using the name-to-number mappings held in the /etc/name_to_major file. This file is maintained by the system and is undocumented. The following is a sample of the /etc/name_to_major file:

```
# more /etc/name_to_major<cr>
 ... <output has been truncated> ...
sckmdrv 199
scsa1394 200
scsa2usb 201
scsi_vhci 202
sd 203
sdpib 204
sdt 205
seeprom 206
ses 207
 ... <output has been truncated> ...
```

To create the minor device entries, the devfsadmd daemon uses the information placed in the dev_info node by the device driver. Permissions and ownership information are kept in the /etc/minor_perm file.

Logical Device Names

The final stage of the autoconfiguration process involves the creation of the logical device name to reflect the new set of devices on the system. Both SPARC and x86 systems use logical device names, but they differ slightly on each platform. To see a list of logical device names for the disks connected to a SPARC system, execute a long listing on the /dev/dsk directory as follows:

```
# ls -l /dev/dsk<cr>
total 96
lrwxrwxrwx   1 root     root          54 Jan 21 07:59 c1t0d0s0 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:a
```

```
lrwxrwxrwx   1 root     root          54 Jan 21 07:59 c1t0d0s1 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:b
lrwxrwxrwx   1 root     root          54 Jan 21 07:59 c1t0d0s2 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:c
lrwxrwxrwx   1 root     root          54 Jan 21 07:59 c1t0d0s3 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:d
 ... <output has been truncated> ...
```

On the second line of output from the `ls -l` command, notice that the logical device name `c1t0d0s0` is linked to the physical device name, as shown in the following:

```
../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:a
```

On SPARC systems, you'll see an eight-string logical device name (`c#t#d#s#`) for each disk slice that contains the following:

- Controller number (`c#`): Identifies the host bus adapter, which controls communications between the system and disk unit. The controller number is assigned in sequential order, such as c0, c1, c2, and so on.
- Target number (`t#`): Target numbers, such as t0, t1, t2, and t3 correspond to a unique hardware address that is assigned to each disk, tape, or DVD-ROM. Some external disk drives have an address switch located on the rear panel. Some internal disks have address pins that are jumpered to assign that disk's target number.
- Disk number (`d#`): The disk number is also known as the logical unit number (LUN). This number reflects the number of disks at the target location. The disk number is always set to 0 on embedded SCSI controllers.
- Slice number (`s#`): A slice number ranging from 0 to 7.

On the x86 platform, IDE and SATA disk drives do not use target controllers. Device names of these types of disks represent the controller (`c3`), disk (`d#`), and slice (`s#`). Because IDE disks do not use target controllers, these disks use a t# value to represent the identity of the disks on its primary and secondary IDE buses. Target values on these systems are as follows:

- t0: Master device on the primary IDE bus
- t1: Slave device on the primary IDE bus
- t2: Master device on the secondary IDE bus
- t3: Slave device on the secondary IDE bus

The following is an example of IDE disks on an x86-based server:

```
# ls -l /dev/dsk<cr>
total 48
lrwxrwxrwx    1 root     root          45 Jan 23 18:11 c0t0d0s0 ->\
 ../../devices/pci@1f,0/pci@1,1/ide@d/sd@0,0:a
lrwxrwxrwx    1 root     root          45 Jan 23 18:11 c0t0d0s1 ->\
 ../../devices/pci@1f,0/pci@1,1/ide@d/sd@0,0:b
lrwxrwxrwx    1 root     root          44 Jan 23 18:11 c1t1d0s0 ->\
 ../../devices/pci@1f,0/pci@1/scsi@8/sd@1,0:a
lrwxrwxrwx    1 root     root          44 Jan 23 18:11 c1t1d0s1 ->\
 ../../devices/pci@1f,0/pci@1/scsi@8/sd@1,0:b
```

x86-based Oracle Solaris systems have a different disk-naming convention, but before describing the logical device name for a disk on an x86-based system, it's worth pointing out a fundamental difference between disk slicing on a SPARC system and disk slicing on an x86-based system. Disk partitioning on Oracle Solaris for the x86 platform has one more level than that of Oracle Solaris for SPARC. On Oracle Solaris for SPARC, slices and partitions are one and the same; on Oracle Solaris for x86, slices are "subpartitions" of a fixed disk (`fdisk`) partition table. This was done to allow Oracle Solaris to coexist with other x86-based OSs, such as for dual-boot configurations.

This difference in slicing brings some differences in the naming of disk devices on an x86-based system. Slices are created in the first partition on a drive and, for SCSI disks, are named the same as with Oracle Solaris for SPARC (`c#t#d0s#`). However, because slices are within an `fdisk` partition table, the x86 partitions have their own device names. The entire drive is named `c#t#d0p0`, and the `fdisk` partitions (maximum of 4) are `c#t#d0p1` through `c#t#d0p4`. To support the x86 environment, the `format` utility also has an added command called `fdisk` to deal with the `fdisk` partitions.

Oracle Solaris x86-based systems have 16 slices (numbered 0-15) versus 8 for SPARC. On the x86 system, slice 8 is used to hold boot code and contains the GRUB `stage1` program in sector 0, the disk label, the VTOC in sectors 1 and 2, and GRUB `stage2` program beginning at sector 50. GRUB is described in Chapter 3. Slice 8 also occupies the first cylinder (cylinder 0) of the `fdisk` partition.

On IDE and SATA disk drives, slice 9 is used for alternate sectors and contains blocks used to store bad block information. Higher slices are available for use but are not supported by `format` at this time, and the `format` utility will only allow you to modify slices 0–7. The major differences between the logical device names used on SPARC-based systems versus x86-based systems are as follows:

`c` is the controller number.
`t` is the SCSI target number.
`s` is the slice number ranging from 0 to 15.

p represents the `fdisk` partition (not slice partition). This number ranges from `p0` to `p4`. `p0` represents the entire disk.
d is the LUN or IDE drive number.

If an IDE drive is used, `d` is used to determine MASTER or SLAVE, and the `t` is not used for IDE drives. For example, two controllers are installed on an x86 PC:

c0 is an IDE controller.
c1 is a SCSI controller.

On an x86-based system, the following devices are listed in the `/dev/dsk` directory for a SATA disk, target 0:

```
c1t0d0p0       c1t0d0s1       c1t0d0s15      c1t0d0s7
c1t0d0p1       c1t0d0s10      c1t0d0s2       c1t0d0s8
c1t0d0p2       c1t0d0s11      c1t0d0s3       c1t0d0s9
c1t0d0p3       c1t0d0s12      c1t0d0s4
c1t0d0p4       c1t0d0s13      c1t0d0s5
c1t0d0s0       c1t0d0s14      c1t0d0s6
```

Examples of logical device names are the following:

- `c1t0d0s0`: A SCSI, SAS, or SATA disk device name that specifies controller 1, target 0, disk 0, and slice 0
- `c1d0p0`: An IDE disk name on an x86-based system that specifies controller 1, disk 0, and `fdisk` partition 0
- `c1d0s0`: An IDE disk name that specifies controller 1, disk 0, and slice 0
- `c2t11d0p0`: A SCSI, SAS, or SATA disk device name on an x86 system that specifies controller 2, target 11, disk 0, and `fdisk` partition 0
- `c2t11d0s0`: A SCSI, SAS, or SATA disk device name that specifies controller 2, target 11, disk 0, and slice 0
- `c3t266000C0FFF7C140d31s2`: A Fibre Channel attached LUN name that specifies controller 3, WWN 266000C0FFF7C140, LUN 31, and slice 2

On both SPARC-based and x86-based systems, the logical device name is the name that the system administrator uses to refer to a particular device when running various file system commands.

For example, if running the `mount` command, use the logical device name `/dev/dsk/c0t0d0s7` to mount the file system `/home`:

```
# mount /dev/dsk/c0t0d0s7 /home<cr>
```

Table 4-2 Device Directories

Directory	Description of Contents
/dev/dsk	Block interface to disk devices
/dev/rdsk	Raw or character interface to disk devices
/dev/rmt	Tape devices
/dev/term	Serial line devices
/dev/cua	Dial-out modems
/dev/pts	Pseudo terminals
/dev/fbs	Frame buffers
/dev/sad	STREAMS administrative driver
/dev/md	Metadevices managed by Oracle Solaris Volume Manager (SVM)

Logical device files in the `/dev` directory are symbolically linked to physical device files in the `/devices` directory. Logical device names are used to access disk devices if you do any of the following:

- Add a new disk to the system.
- Move a disk from one system to another.
- Access (or mount) a file system residing on a local disk.
- Back up a local file system.
- Repair a file system.

Logical devices are organized in subdirectories under the `/dev` directory by their device types, as shown in Table 4-2.

Block and Raw Devices

Disk drives have an entry under both the `/dev/dsk` and `/dev/rdsk` directories. The `/dsk` directory refers to the block or buffered device file, and the `/rdsk` directory refers to the character or raw device file. The "r" in `rdsk` stands for "raw." You may even hear these devices referred to as "cooked" and "uncooked" devices.

The `/dev/dsk` directory contains the disk entries for the block device nodes in `/devices`, as shown in the following command output:

```
# ls -l /dev/dsk<cr>
total 96
lrwxrwxrwx  1 root      root           54 Jan 21 07:59 c1t0d0s0 ->\
```

```
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:a
lrwxrwxrwx   1 root    root          54 Jan 21 07:59 c1t0d0s1 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:b
lrwxrwxrwx   1 root    root          54 Jan 21 07:59 c1t0d0s2 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:c
lrwxrwxrwx   1 root    root          54 Jan 21 07:59 c1t0d0s3 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:d
 ... <Output has been truncated> ...
```

The /dev/rdsk directory contains the disk entries for the character device nodes in /devices, as shown in the following command:

```
# ls -l /dev/rdsk<cr>
total 96
lrwxrwxrwx   1 root     root          58 Jan 21 07:59 c1t0d0s0 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:a,raw
lrwxrwxrwx   1 root     root          58 Jan 21 07:59 c1t0d0s1 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:b,raw
lrwxrwxrwx   1 root     root          58 Jan 21 07:59 c1t0d0s2 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:c,raw
lrwxrwxrwx   1 root     root          58 Jan 21 07:59 c1t0d0s3 ->\
 ../../devices/pci@0,600000/pci@0/pci@0/scsi@0/sd@0,0:d,raw

 ... <Output has been truncated> ...
```

A File System Defined

A file system is a collection of files and directories stored on disk in a standard UNIX file system (UFS) format. All disk-based computer systems have a file system. In UNIX, file systems have two basic components: files and directories. A file is the actual information as it is stored on the disk, and a directory is a list of the filenames. In addition to keeping track of filenames, the file system must keep track of a file's access date, permissions, and ownership. Managing file systems is one of the system administrator's most important tasks. Administration of the file system involves the following:

- Ensuring that users have access to data. This means that systems are up and operational, file permissions are set up properly, and data is accessible.
- Protecting file systems against file corruption and hardware failures. This is accomplished by checking the file system regularly and maintaining proper system backups.
- Securing file systems against unauthorized access. Only authorized users should have access to files.
- Providing users with adequate space for their files.

- Keeping the file system clean. In other words, data in the file system must be relevant and not wasteful of disk space. Procedures are needed to make sure that users follow proper naming conventions and that data is stored in an organized manner.

You'll see the term "file system" used in several ways. Usually, "file system" describes a particular type of file system (disk based, network based, or virtual). It might also describe the entire file tree from the root directory downward. In another context, the term "file system" might be used to describe the structure of a disk slice, which is described later in this chapter.

Creating and administering ZFS file systems is described in the next chapter.

Defining a Disk's Geometry

Before creating a file system on a disk, you need to understand the basic geometry of a disk drive. Disks come in many shapes and sizes. The number of heads, tracks, and sectors and the disk capacity vary from one model to another. Basic disk terminology is described in Table 4-3.

A hard disk consists of several separate disk platters mounted on a common spindle. Data stored on each platter surface is written and read by disk heads. The circular path that a disk head traces over a spinning disk platter is called a "track."

Table 4-3 Disk Terminology

Disk Term	Description
Track	A concentric ring on a disk that passes under a single stationary disk head as the disk rotates.
Cylinder	The set of tracks with the same nominal distance from the axis about which the disk rotates.
Sector	Section of each disk platter. A sector holds 512 bytes.
Block	A data storage area on a disk. A disk block is 512 bytes.
Disk controller	A chip and its associated circuitry that control the disk drive.
Disk label	The first sector of a disk (block 0) that contains disk geometry and partition information. Also referred to as the VTOC. To label a disk means to write slice information onto the disk. You usually label a disk after you change its slices using the `format` command.
Device driver	A kernel module that controls a hardware or virtual device.

Each track is made up of a number of sectors laid end to end. A sector consists of a header, a trailer, and 512 bytes of data. The header and trailer contain error-checking information to help ensure the accuracy of the data. Taken together, the set of tracks traced across all the individual disk platter surfaces for a single position of the heads is called a "cylinder."

Disk Controllers

Associated with every disk is a controller, an intelligent device responsible for organizing data on the disk. Some disk controllers are located on a separate circuit board, such as SCSI. Other controller types are integrated with the disk drive, such as SATA and IDE.

Defect List

Disks might contain areas where data cannot be written and retrieved reliably. These areas are called "defects." The controller uses the error-checking information in each disk block's trailer to determine whether a defect is present in that block. When a block is found to be defective, the controller can be instructed to add it to a defect list and avoid using that block in the future. The last two cylinders of a disk are set aside for diagnostic use and for storing the disk defect list.

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 or VTOC.

To label a disk means to write slice information onto the disk. You usually label a disk after defining its slices. If you fail to label a disk after creating slices, the slices will be unavailable because the OS has no way of knowing about them.

Oracle Solaris supports two types of disk labels:

- SMI: The traditional VTOC disk label used for boot disks and disks smaller than two terabytes (2TB)
- EFI: The Extensible Firmware Interface label for disks larger than 2TB

The advantages of the EFI disk label over the SMI disk label are as follows:

- Provides support for disks greater than 2TB in size.
- Provides usable slices 0–6, where slice 2 is just another slice.

- Partitions (or slices) cannot overlap with the primary or backup label, nor with any other partitions. The size of the EFI label is usually 34 sectors, so partitions start at sector 34. This feature means that no partition can start at sector zero (0).
- No cylinder, head, or sector information is stored in the EFI label. Sizes are reported in blocks.
- Information that was stored in the alternate cylinders area, the last two cylinders of the disk, is now stored in slice 8.
- If you use the `format` utility to change partition sizes, the `unassigned` partition tag is assigned to partitions with sizes equal to zero. By default, the `format` utility assigns the `usr` partition tag to any partition with a size greater than zero. You can use the partition change menu to reassign partition tags after the partitions are changed.
- Oracle Solaris ZFS file systems use EFI labels by default when the entire disk is selected.

The following are restrictions of the EFI disk label:

- A disk with an EFI label may not be recognized on systems running older releases.
- Up until Oracle Solaris 11.1, the x86- and SPARC-based systems could not boot from a disk with an EFI disk label. In Oracle Solaris 11.1, x86-based systems can now boot to an EFI (GPT) labeled disk using GRUB2 (an updated version of GRUB). As of this writing, this feature is not currently available on SPARC-based systems. A boot disk on a SPARC-based system is installed with a legacy VTOC (SMI) label.
- The EFI specification prohibits overlapping slices. The entire disk is represented by c#t#d#.
- 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 on disks with EFI labels:
  - The `save` option is not supported because disks with EFI labels do not need an entry in the `format.dat` file.
  - The `backup` option is not applicable because the disk driver finds the primary label and writes it back to the disk.

It may be necessary to change a disk label from SMI to EFI or vice versa. Use the `format` command with the `-e` option as described in the section titled, "Using the `format` Utility to Create Slices: SPARC" later in this chapter.

Partition Tables

An important part of the disk label is the partition table, which identifies a disk's slices, the slice boundaries (in cylinders), and the total size of the slices. A disk's partition table can be displayed by using the `format` utility described in the "Disk Slices" section that follows.

Disk Slices

Disks are divided into regions called "disk slices" or "disk partitions." A slice is composed of a single range of contiguous blocks. It is a physical subset of the disk (except for slice 2, which represents the entire disk). A UFS or the swap area is built within these disk slices. The boundaries of a disk slice are defined when a disk is partitioned using the `format` utility, and the slice information for a particular disk can be viewed by using the `prtvtoc` command. Each disk slice appears to the OS (and to the system administrator) as though it were a separate disk drive.

> **Disk Slices and ZFS**
>
> With ZFS file systems, disk slicing has become unnecessary except for configuring the boot disk. ZFS file systems are described in Chapter 5, "Administering ZFS File Systems." Previous versions of Oracle Solaris used disk slices, and you should be familiar with the concept in case you encounter disk slices on other Oracle Solaris systems.

Disk slicing differs between the SPARC and x86 platforms. On the SPARC platform, the entire disk is devoted to the OS; the disk can be divided into 8 slices, numbered 0 to 7. On the x86 platform, the disk is divided into `fdisk` partitions using the `fdisk` command. The `fdisk` partition is divided into 10 slices, numbered 0 to 9.

> **Slices Versus Partitions**
>
> Oracle Solaris device names use the term "slice" (and the letter "s" in the device name) to refer to the slice number. Slices were called partitions in SunOS 4.x. This book attempts to use the term "slice" whenever possible; however, certain interfaces, such as the `format` and `prtvtoc` commands, refer to slices as partitions.

A physical disk consists of a stack of circular platters. Data is stored on these platters in a cylindrical pattern called "cylinders" as illustrated in Figure 4-1. Cylinders can be grouped and isolated from one another. A group of cylinders is referred to as a slice. A slice is defined with start and end points, starting from the outside of the platters to the center of the stack of platters, which is called the "spindle."

For example, a 73GB SCSI disk has 14,087 cylinders, numbered 0 to 14,086. Cylinder 0 is on the outside edge of the platters, and cylinder 14,086 is the closest to the spindle. Disk slices are defined by an offset and a size in cylinders. The offset is

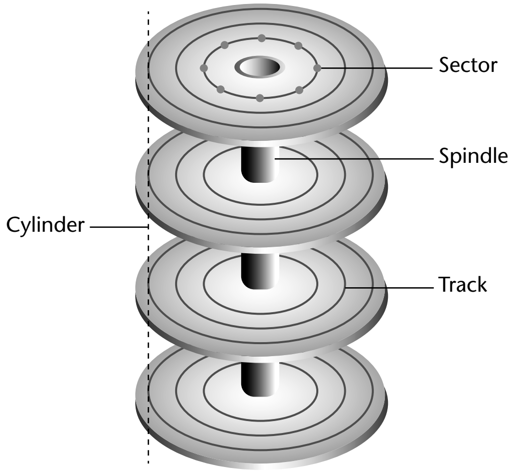


Figure 4-1 Disk platters and cylinders

the distance from cylinder 0. To define a slice, the administrator provides a starting cylinder and an ending cylinder. A slice spanning from cylinder 0 to 14,086 would use the entire disk and is typical of the slicing scheme used on a ZFS boot disk.

When setting up slices, remember these rules:

- Each disk slice holds only one file system.
- No file system can span multiple slices without the use of a volume manager such as ZFS or SVM.
- After a file system is created, its size cannot be increased or decreased without repartitioning and possibly destroying the partition directly before or after it.
- Slices cannot span multiple disks; however, multiple swap slices on separate disks are allowed.

When we discuss ZFS in Chapter 5, you'll learn how to get around some of these limitations in file systems.

Displaying Disk Configuration Information

As described earlier, disk configuration information is stored in the disk label. If you know the disk and slice number, you can display information for a disk by using the print volume table of contents (`prtvtoc`) command. You can specify the volume by specifying any slice defined on the disk (for example, `/dev/rdsk/c0t3d0s2` or `/dev/rdsk/c0t3d0s*`). Regardless of which slice you specify, all slices defined on the disk will be displayed. If you know the target number of the disk but do not know how it is divided into slices, you can show information for the entire disk by specifying either slice 2 or s*. The following steps show how you can examine information stored on a disk's label by using the `prtvtoc` command.

1. Become the superuser.
2. Type the following text and press *Enter*.

```
# prtvtoc /dev/rdsk/c2t0d0s2<cr>
```

The system responds with the following:

```
* /dev/rdsk/c2t0d0s2 partition map
*
* Dimensions:
*     512 bytes/sector
*     424 sectors/track
*      24 tracks/cylinder
*   10176 sectors/cylinder
*   14089 cylinders
*   14087 accessible cylinders
*
* Flags:
*   1: unmountable
*  10: read-only
*
* Unallocated space:
*       First     Sector    Last
*       Sector     Count    Sector
*           0     10176     10175
*
*                           First     Sector     Last
* Partition Tag Flags       Sector     Count     Sector Mount Directory
       0     2   00        10176 143339136 143349311
       2     5   01            0 143349312 143349311
```

The disk described is a SAS disk, target 0 with an SMI VTOC label. The `prtvtoc` command shows the number of cylinders and heads, as well as how the disk's slices are arranged.

The following is an example of running the `prtvtoc` command on a SCSI disk with an EFI label:

```
# prtvtoc /dev/rdsk/c2t1d0s1<cr>
* /dev/rdsk/c2t1d0s1 partition map
```

continues

```
*
* Dimensions:
*     512 bytes/sector
* 8385121 sectors
* 8385054 accessible sectors
*
* Flags:
*   1: unmountable
*  10: read-only
*
*                         First    Sector    Last
* Partition  Tag  Flags  Sector    Count    Sector Mount Directory
       0      2    01        34    41006     41039
       1      2    00     41040  8327663   8368702  /mnt
       8     11    00   8368703    16384   8385086
```

Using the `format` Utility to Create Slices: SPARC

Before you can create a file system on a disk, the disk must be formatted, and you must divide it into slices using the `format` utility. Formatting involves two separate processes:

- Writing format information to the disk
- Completing a surface analysis, which compiles an up-to-date list of disk defects

When a disk is formatted, header and trailer information is superimposed on the disk. When the `format` utility runs a surface analysis, the controller scans the disk for defects. It should be noted that defects and formatting information reduce the total disk space available for data. This is why a new disk usually holds only 90% to 95% of its capacity after formatting. This percentage varies according to disk geometry and decreases as the disk ages and develops more defects.

The need to perform a surface analysis on a disk drive has dropped as more manufacturers ship their disk drives formatted and partitioned. You should not need to perform a surface analysis within the `format` utility when adding a disk drive to an existing system unless you think disk defects are causing problems. The primary reason that you would use `format` is if you want to view or change the partitioning scheme on a disk.

Always Back Up Your Data

Formatting and creating slices is a destructive process, so make sure user data is backed up before you start.

The `format` utility searches your system for all attached disk drives and reports the following information about the disk drives it finds:

- Target location
- Disk geometry
- Whether the disk is formatted
- Whether the disk has mounted partitions

In addition, the `format` utility is used in disk repair operations to do the following:

- Retrieve disk labels
- Repair defective sectors
- Format and analyze disks
- Partition disks
- Label disks (i.e., write the disk name and configuration information to the disk for future retrieval)

The installation program partitions and labels disk drives as part of installing the Oracle Solaris release. However, you might need to use the `format` utility when doing the following:

- Displaying slice information
- Dividing a disk into slices
- Formatting a disk drive when you think disk defects are causing problems
- Repairing a disk drive
- Changing a disk label from EFI to SMI or vice versa

The following example uses the `format` utility to create disk slices on a disk.

1. Become the superuser
2. Type "code."

   The system responds with the following:

```
Searching for disks ... done

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@1f,0/pci@1/scsi@8/sd@0,0
       1. c0t1d0 <SUN36G cyl 24620 alt 2 hd 27 sec 107>
          /pci@1f,0/pci@1/scsi@8/sd@1,0
```

3. Specify the disk (enter its number).

   The system responds with the format main menu:

```
FORMAT MENU:
    disk - select a disk
    type - select (define) a disk type
    partition - select (define) a partition table
    current - describe the current disk
    format - format and analyze the disk
    repair - repair a defective sector
    label - write label to the disk
    analyze - surface analysis
    defect - defect list management
    backup - search for backup labels
    verify - read and display labels
    save - save new disk/partition definitions
    inquiry - show vendor, product and revision
    volname - set 8-character volume name
    !<cmd> - execute <cmd>, then return
    quit
```

   Table 4-4 describes the format main menu items.

4. Type "partition" at the format prompt. The partition menu is displayed.

Using Shortcuts in the `format` Utility

It is unnecessary to type the entire command. After you type the first two characters of a command, the `format` utility recognizes the entire command.

```
format> partition<cr>
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
    !<cmd> - execute <cmd>, then return
    quit
```

5. Type "print" to display the current partition map.

```
partition> print<cr>
```

Table 4-4 Format Main Menu Item Descriptions

Menu Item	Description
disk	Lists all of the system's drives. Also lets you choose the disk you want to use in subsequent operations. This disk is referred to as the current disk.
type	Identifies the manufacturer and model of the current disk. Also displays a list of known drive types. Choose the *Auto configure* option for all SCSI-2 disk drives.
partition	Creates and modifies slices.
current	Describes the current disk (that is, device name, device type, number of cylinders, alternate cylinders, heads, sectors, and physical device name).
format	Formats the current disk using one of these sources of information in this order:
	Information that is found in the format.dat file.
	Information from the automatic configuration process.
	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	x86 platform only: Runs the fdisk program to create an Oracle Solaris fdisk partition.
repair	Used to repair a specific block on the current disk.
label	Writes a new label to the current disk. This is not the same as labeling the disk with volname.
analyze	Runs read, write, and compare tests.
defect	Retrieves and displays defect lists. This feature does not apply to IDE disks. IDE disks manage defects automatically.
backup	Searches for backup labels if the VTOC becomes corrupted or gets deleted.
verify	Displays information about the current disk such as device name, device type, number of cylinders, alternate cylinders, heads, sectors, and partition table.
save	Saves new disk and partition information.
inquiry	SCSI disks only: Displays the vendor, product name, and revision level of the current drive. This will also display the disk's current firmware.
volname	Labels the disk with a new eight-character volume name that you specify. This is not the same as writing the partition table to disk using label.
quit	Exits the format menu. Pressing *Ctrl+D* will also exit the format utility from the main menu or from any submenu.

The system responds with the following:

```
Current partition table (original):
Total disk cylinders available: 24620 + 2 (reserved cylinders)

Part       Tag    Flag     Cylinders        Size          Blocks
  0       root     wm    1418 -  9924      11.72GB  (8507/0/0)  24576723
  1        var     wm    9925 - 13469       4.88GB  (3545/0/0)  10241505
  2     backup     wm       0 - 24619      33.92GB  (24620/0/0) 71127180
  3       swap     wu       0 -  1417      1.95GB  (1418/0/0)  4096602
  4 unassigned     wm   13470 - 14887       1.95GB  (1418/0/0)  4096602
  5 unassigned     wm   14888 - 16112       1.69GB  (1225/0/0)  3539025
  6 unassigned     wm   16113 - 16821    1000.15MB  (709/0/0)  2048301
  7       home     wm   16822 - 23910       9.77GB  (7089/0/0)  20480121
```

The columns displayed with the partition table are

- Part: The slice number (0–7).
- Tag: This is an optional value that indicates how the slice is being used. The value can be any of the following names that best fits the function of the file system you are creating:
  `unassigned`, `boot`, `root`, `swap`, `usr`, `backup`, `stand`, `var`, `home`, `alternates`, `reserved`, `system`, `BIOS_boot`
- Flag: Values in this column can be

  `wm` The disk slice is writable and mountable.
  `wu` The disk slice is writable and unmountable (such as a swap slice).
  `rm` The disk slice is read-only and mountable.
  `ru` The disk slice is read-only and unmountable.
- Cylinders: The starting and ending cylinder number for the disk slice.
- Size: The slice size specified as

  `mb` megabytes
  `gb` gigabytes
  `b` blocks
  `c` cylinders
  `b` Blocks
  `e` Ending cylinder

Wasted Disk Space

Wasted disk space occurs during partitioning when one or more cylinders have not been allocated to a disk slice. This may happen intentionally or accidentally. If there are unallocated slices available, then wasted space can possibly be assigned to a slice at another time.

You can use the `name` and `save` commands in the partition menu to name and save a newly created partition table to a file that can be referenced by name later, when you want to use this same partition scheme on another disk. When issuing the `name` command, you'll provide a unique name for this partition scheme and then issue the `save` command to save the information to the `./format.dat` file. Normally this file is located in the `/etc` directory, so provide the full pathname for `/etc/format.dat` to update the master file.

6. After you partition the disk, you must label it by typing `label` at the partition prompt:

```
partition> label<cr>
```

You are asked for confirmation on labeling the disk as follows:

```
Ready to label disk, continue? y<cr>
```

Enter "Y" to continue.

Label Your Drive

To label a disk means to write slice information onto the disk. If you don't label the drive when exiting the `format` utility, your partition changes will not be retained. It's a good idea to get into the habit of labeling at the partition submenu, but you can also label at the `format` utility main menu as well—you get two chances to remember before exiting the utility.

7. After labeling the disk, type "quit" to exit the partition menu or press *Ctrl+D* to exit the `format` utility:

```
partition> quit<cr>
```

8. Type "quit" again to exit the format utility:

```
format> quit<cr>
```

It's important to point out a few undesirable things that can happen when defining disk partitions with the `format` utility if you're not careful. First, be careful not to waste disk space. Wasted disk space can occur when you decrease the size of one slice and do not adjust the starting cylinder number of the adjoining disk slice.

Second, don't overlap disk slices. Overlapping occurs when one or more cylinders are allocated to more than one disk slice. For example, increasing the size of one slice without decreasing the size of the adjoining slice will create overlapping partitions. The `format` utility will not warn you of wasted disk space or overlapping partitions.

The main reason a system administrator uses the `format` utility is to divide a disk into disk slices. In Oracle Solaris 11 11/11, for a bootable ZFS root pool, the disks in the pool must contain slices and must be labeled with an SMI label. The simplest configuration would be to put the entire disk capacity in slice 0 and use that slice for the root pool. Bootable ZFS root pools are discussed further in Chapter 5.

EFI Boot Disks in Oracle Solaris 11.1

Oracle Solaris 11.1 is the first release to support booting from an EFI (GPT) or SMI formatted disk on the x86-based systems using GRUB2. As of this writing, the SPARC platform still requires the boot disk to contain an SMI label.

I'll describe how to use the SMI label for SPARC-based systems. For example, on a SPARC-based system with a 72GB disk, you would need to have 68GB of usable space located in slice 0. Similarly, on an x86-based system with a 72GB disk, you would also need to allow 68GB of usable space located in slice 0. A small amount of boot information is contained in slice 8. Slice 8 requires no administration and cannot be changed.

Follow these steps to partition a disk (`c2t0d0`) to be used as a ZFS boot disk (bootable ZFS root pool) on a SPARC-based system. If the disk has an EFI label, and the firmware has not been upgraded, you must first convert it to an SMI label.

Use the `prtvtoc` command to verify the disk label as follows:

```
# prtvtoc /dev/rdsk/c2t0d0s2<cr>
```

The system displays

```
Part      Tag     Flag  Cylinders      Size            Blocks
  0       root     wm    1 - 14086    68.35GB     (14086/0/0) 143339136
  1 unassigned     wm    0             0          (0/0/0)             0
  2     backup     wu    0 - 14086    68.35GB     (14087/0/0) 143349312
  3 unassigned     wm    0             0          (0/0/0)             0
  4 unassigned     wm    0             0          (0/0/0)             0
  5 unassigned     wm    0             0          (0/0/0)             0
  6 unassigned     wm    0             0          (0/0/0)             0
  7 unassigned     wm    0             0          (0/0/0)             0
```

Notice that slice 2 is labeled "backup" and the slices are numbered 0-7. This is an SMI label.

The following shows the output that is displayed for a disk with an EFI label:

```
Part      Tag     Flag   First Sector        Size        Last Sector
  0        usr     wm              34      68.36GB         143358320
  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
  7 unassigned     wm               0          0              0
  8   reserved     wm       143358321       8.00MB         143374704
```

Notice there is a slice 8 and slice 2 is NOT labeled "backup." This is an EFI label and would need to be changed. Use the following steps to change the label from an EFI label to an SMI label:

1. As root, use the `format -e` command and select the disk to label as follows:

```
# format -e<cr>
```

The system displays a list of disks. In the example, I selected disk 1 (`c2t1d0`):

```
Searching for disks ... done
AVAILABLE DISK SELECTIONS:
       0. c2t0d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
          /pci@780/pci@0/pci@9/scsi@0/sd@0,0
       1. c2t1d0 <SEAGATE-ST973402SSUN72G-0603-68.37GB>
          /pci@780/pci@0/pci@9/scsi@0/sd@1,0
Specify disk (enter its number): 1<cr>
selecting c2t1d0
[disk formatted]
```

2. The main menu is displayed. Type "label" to `label` the disk:

```
FORMAT MENU:
   disk       - select a disk
   type       - select (define) a disk type
   partition  - select (define) a partition table
   current    - describe the current disk
   format     - format and analyze the disk
   repair     - repair a defective sector
   label      - write label to the disk
   analyze    - surface analysis
   defect     - defect list management
   backup     - search for backup labels
   verify     - read and display labels
   inquiry    - show disk ID
   scsi       - independent SCSI mode selects
   cache      - enable, disable or query SCSI disk cache
   volname    - set 8-character volume name
```

continues

```
    !<cmd>  - execute <cmd>, then return
    quit
format> label<cr>
```

3. Select option 1 to label the disk with an SMI label and press *Enter* when prompted for autoconfiguration:

```
[0] SMI Label
[1] EFI Label
Specify Label type[1]: 0<cr>
Auto configuration via format.dat[no]?<cr>
Auto configuration via generic SCSI-2[no]?<cr>
format>
```

4. Exit the `format` utility.

```
format> quit<cr>
```

To slice the disk so that it can be used as a ZFS boot disk, follow these steps:

1. As root, enter the `format` utility:

```
# format<cr>
Searching for disks ... done
```

   Select the disk that is going to be sliced. In the example, I will select disk 1 (`c2t1d0`):

```
AVAILABLE DISK SELECTIONS:
        0. c2t0d0 <SUN72G cyl 14087 alt 2 hd 24 sec 424>
           /pci@780/pci@0/pci@9/scsi@0/sd@0,0
        1. c2t1d0 <SEAGATE-ST973402SSUN72G-0603-68.37GB>
           /pci@780/pci@0/pci@9/scsi@0/sd@1,0
Specify disk (enter its number): 1<cr>
```

   The system responds with

```
selecting c2t1d0
[disk formatted]
```

2. Type "partition" at the `format` prompt. The partition menu is displayed.

```
format> partition<cr>
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
!<cmd> - execute <cmd>, then return
quit
```

3. Type "print" to display the current partition map.

```
partition> print<cr>
```

The system responds with the following:

```
partition> print<cr>
Current partition table (original):
Total disk cylinders available: 14087 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders       Size            Blocks
  0       root    wm       0 -    25    129.19MB    (26/0/0)       264576
  1       swap    wu      26 -    51    129.19MB    (26/0/0)       264576
  2     backup    wu       0 - 14086     68.35GB    (14087/0/0) 143349312
  3 unassigned    wm       0               0        (0/0/0)             0
  4 unassigned    wm       0               0        (0/0/0)             0
  5 unassigned    wm       0               0        (0/0/0)             0
  6        usr    wm      52 - 14086     68.10GB    (14035/0/0) 142820160
  7 unassigned    wm       0               0        (0/0/0)             0

partition>
```

4. Enter "modify" to change the partition table:

```
partition> modify<cr>
```

5. Select option 1 for "All Free Hog" when prompted:

```
Select partitioning base:
        0. Current partition table (original)
        1. All Free Hog
Choose base (enter number) [0]? 1<cr>

Part      Tag    Flag     Cylinders      Size        Blocks
  0       root    wm       0              0          (0/0/0)             0
```

continues

```
  1       swap    wu       0             0          (0/0/0)             0
  2     backup    wu       0 - 14086    68.35GB    (14087/0/0) 143349312
  3 unassigned    wm       0             0          (0/0/0)             0
  4 unassigned    wm       0             0          (0/0/0)             0
  5 unassigned    wm       0             0          (0/0/0)             0
  6        usr    wm       0             0          (0/0/0)             0
  7 unassigned    wm       0             0          (0/0/0)             0
```

6. Type "yes" when asked whether to continue:

```
Do you wish to continue creating a new partition table based on above table[yes]?
yes<cr>
```

Type "0" for the Free Hog partition:

```
Free Hog partition[6]? 0<cr>
```

The Free Hog Slice

When using the `format` utility to change the size of disk slices, a temporary slice is automatically designated that expands and shrinks to accommodate the slice resizing operations. This temporary slice is referred to as the "free hog," and it represents the unused disk space on a disk drive. If a slice is decreased, the free hog expands. After the `modify` operation is complete, the remaining free hog space is allocated to the slice specified.

7. The system will prompt you to enter a size for each partition. Press *Enter* when prompted as follows, and each slice will be 0MB:

```
Enter size of partition '1' [0b, 0c, 0.00mb, 0.00gb]:<cr>
Enter size of partition '3' [0b, 0c, 0.00mb, 0.00gb]:<cr>
Enter size of partition '4' [0b, 0c, 0.00mb, 0.00gb]:<cr>
Enter size of partition '5' [0b, 0c, 0.00mb, 0.00gb]:<cr>
Enter size of partition '6' [0b, 0c, 0.00mb, 0.00gb]:<cr>
Enter size of partition '7' [0b, 0c, 0.00mb, 0.00gb]:<cr>
```

Because all of the slices have been set to "0," the free hog space is the entire disk. This space will be allocated to slice 0 as specified in step 6.

8. When prompted to make this the current partition table, press *Enter* to use the default value "yes":

```
Okay to make this the current partition table[yes]?<cr>
```

9. When prompted for a table name, enter "rootdisk." This name is not significant and can be any name.

```
Enter table name (remember quotes): rootdisk<cr>
```

Enter "pr" to display the new partition table:

```
partition> pr<cr>
Current partition table (unnamed):
Total disk cylinders available: 14087 + 2 (reserved cylinders)
Part      Tag    Flag     Cylinders       Size       Blocks
  0       root    wm       0 - 14086      68.35GB    (14087/0/0) 143349312
  1       swap    wu       0               0         (0/0/0)             0
  2     backup    wu       0 - 14086      68.35GB    (14087/0/0) 143349312
  3 unassigned    wm       0               0         (0/0/0)             0
  4 unassigned    wm       0               0         (0/0/0)             0
  5 unassigned    wm       0               0         (0/0/0)             0
  6        usr    wm       0               0         (0/0/0)             0
  7 unassigned    wm       0               0         (0/0/0)             0

partition>
```

Notice that slice 0 is the entire disk.

10. Enter "quit" or press *Ctrl+D* to exit the `format` utility.

```
partition> quit<cr>
```

Using the `format` Utility to Create Slices: x86

As described earlier in this chapter, Oracle Solaris on the x86 platform treats disk drives slightly differently than on the SPARC-based systems. Disks on the x86 platform must have an `fdisk` partition table. The x86-based systems use the `fdisk` partition table to identify parts of the disk reserved for different OSs and to identify the partition that the system will boot from. This boot partition is referred to as the "active disk" partition. You can assign one `fdisk` partition on a disk to be used for Oracle Solaris.

On an x86-based system, once a disk drive has been physically installed and verified as working, you'll use the `format` command to slice the disk, but first an `fdisk` partition must be created on the new drive. You can create this `fdisk` partition using the `fdisk` command from the command line or through the `format` utility.

The following steps describe how to create a fixed disk partition table on a disk using the format utility:

1. As root, type "format" to get into the `format` utility.

```
# format<cr>
```

The following menu appears:

```
AVAILABLE DISK SELECTIONS:
       0. c1t0d0 <FUJITSU-M1606S-512-6234 cyl 3455 alt 2 hd 6 sec 102>
           /pci@0,0/pci9004,8178@3/cmdk@0,0
       1. c1t1d0 <IBM-DFHSS1W!e-4141 cyl 4071 alt 2 hd 4 sec 135>
           /pci@0,0/pci9004,8178@3/cmdk@1,0
       2. c1t2d0 <DEFAULT cyl 2928 alt 2 hd 6 sec 120>
           /pci@0,0/pci9004,8178@3/cmdk@2,0
Specify disk (enter its number):
```

2. Enter the number corresponding to the new drive and the following menu will be displayed:

```
FORMAT MENU:
   disk       - select a disk
   type       - select (define) a disk type
   partition  - select (define) a partition table
   current    - describe the current disk
   format     - format and analyze the disk
   fdisk      - run the fdisk program
   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 name5
   quit
format>
```

3. Select the `fdisk` option and the following menu appears:

```
The recommended default partitioning for your disk is:
a 100% "SOLARIS System" partition.

To select this, please type "y". To partition your disk
differently, type "n" and the "fdisk" program will let you select other
partitions.
```

4. If you wish to use the entire drive for Oracle Solaris, enter "Y." This will return you to the format menu. If "N" is entered, the `fdisk` menu will be displayed.

```
        Total disk size is 4073 cylinders
        Cylinder size is 540 (512 byte) blocks
                                      Cylinders
    Partition  Status  Type       Start  End  Length    %
    =========  ======  ======== =====  ===  ======  ===

THERE ARE NO PARTITIONS CURRENTLY DEFINED
SELECT ONE OF THE FOLLOWING:
        1.  Create a partition
        2.  Change Active (Boot from) partition
        3.  Delete a partition
        4.  Exit (Update disk configuration and exit)
        5.  Cancel (Exit without updating disk configuration)
Enter Selection:
```

5. Choose `1` to create an `fdisk` partition. This is not the same as a slice.
6. After creating the partition, choose `4` to exit and save. The format menu will return.
7. Choose `partition` and follow the procedure for formatting a disk on page 272, beginning at step 4.

Disks on x86-based systems can be divided into 10 slices labeled slice 0 through slice 9. On Oracle Solaris 11/11, slices 0 through 7 are used for the same purposes as disk slices found on SPARC-based systems. Slice 2 represents all of the space within the `fdisk` partition. As stated earlier, slices 8 and 9 are used for purposes specific to x86-based hardware. You cannot modify slices 8 and 9 using the `format` utility. Beginning with Oracle Solaris 11.1, the boot disk on an x86-based system can contain an EFI label, and the partition scheme is slightly different than the SPARC system. Slice 0 is reserved for the `BIOS_boot` information. I'll describe this more in the next chapter.

Here's an example of the partition table on an IDE or SATA disk on an x86-based system running Oracle Solaris 11/11, as displayed by the `format` utility:

```
Part  Tag          Flag    Cylinders    Size       Blocks
(output omitted)
8     boot         wu      0 - 0        7.84MB     (1/0/0) 16065
9     alternates   wm      1 - 2        15.69MB    (2/0/0) 32130
```

In the previous example, notice that slice 9 is defined and tagged as the `alternates` slice.

The next example shows the partition table for a SCSI disk attached to an x86-based system. Notice that partition 8 is assigned, but slice 9 is not used:

```
Part  Tag          Flag    Cylinders    Size       Blocks
(output omitted)
8     boot         wu      0 - 0        7.84MB     (1/0/0) 16065
9     unassigned   wm      0            0          (0/0/0) 0
```

One more item of note: On standard UFSs, don't change the size of disk slices that are currently in use. When a disk with existing slices is repartitioned and relabeled, any existing data will be lost. Before repartitioning a disk, first copy all of the data to tape or to another disk.

You can also create the fixed disk partition table on an x86-based system disk from the command line using a single command as follows:

```
# fdisk -B c1t0d0<cr>
```

The -B option creates a single fixed disk partition that spans the entire disk. The following 36GB disk was formatted using the fdisk -B command:

```
Part      Tag  Flag  Cylinders    Size         Blocks
  0 unassigned  wm    0            0      (0/0/0)           0
  1 unassigned  wm    0            0      (0/0/0)           0
  2     backup  wu    0 - 4695    35.97GB (4696/0/0)  75441240
  3 unassigned  wm    0            0      (0/0/0)           0
  4 unassigned  wm    0            0      (0/0/0)           0
  5 unassigned  wm    0            0      (0/0/0)           0
  6 unassigned  wm    0            0      (0/0/0)           0
  7 unassigned  wm    0            0      (0/0/0)           0
  8       boot  wu    0 -  0       7.84MB (1/0/0)       16065
  9 unassigned  wm    0            0      (0/0/0)           0
```

The fdisk -B command can also be used to convert a disk label from an EFI to SMI.

To verify that a disk contains a fixed disk partition table, issue the following command:

```
# fdisk -v -W - /dev/rdsk/c3t0d0p<cr>
```

The system displays the fdisk table for disk c3t0d0:

```
* /dev/rdsk/c3t0d0p0 default fdisk table
* Dimensions:
*  512 bytes/sector
*  63 sectors/track
*  255 tracks/cylinder
*  2088 cylinders
*
* systid:
*     1: DOSOS12
*     2: PCIXOS
*     4: DOSOS16
*     5: EXTDOS
*     6: DOSBIG
*     7: FDISK_IFS
```

```
*     8: FDISK_AIXBOOT
*     9: FDISK_AIXDATA
*    10: FDISK_0S2BOOT
*    11: FDISK_WINDOWS
*    12: FDISK_EXT_WIN
*    14: FDISK_FAT95
*    15: FDISK_EXTLBA
*    18: DIAGPART
*    65: FDISK_LINUX
*    82: FDISK_CPM
*    86: DOSDATA
*    98: OTHEROS
*    99: UNIXOS
*   100: FDISK_NOVELL2
*   101: FDISK_NOVELL3
*   119: FDISK_QNX4
*   120: FDISK_QNX42
*   121: FDISK_QNX43
*   130: SUNIXOS
*   131: FDISK_LINUXNAT
*   134: FDISK_NTFSVOL1
*   135: FDISK_NTFSVOL2
*   165: FDISK_BSD
*   167: FDISK_NEXTSTEP
*   183: FDISK_BSDIFS
*   184: FDISK_BSDISWAP
*   190: X86BOOT
*   191: SUNIXOS2
*   238: EFI_PMBR
*   239: EFI_FS


* Id    Act    Bhead    Bsect    Bcyl    Ehead    Esect    Ecyl    Rsect    Numsect
  191   128    0        1        1       254      63       1023    16065    33527655
  0     0      0        0        0       0        0        0       0        0
  0     0      0        0        0       0        0        0       0        0
  0     0      0        0        0       0        0        0       0        0
```

When there are multiple disks of the same type (manufacturer, model, size, and geometry) to be sliced, you can save time by copying the label from a source disk over to a target disk without going through all of the steps using the `format` utility. Use the `prtvtoc` command to get the partition table from the source disk (`c0t0d0`) and write the table to the target disk (`c0t1d0`) using the `fmthard` command as follows:

```
# prtvtoc /dev/rdsk/c0t0d0s2 | fmthard -s - /dev/rdsk/c0t1d0s2<cr>
```

Administering LUNs

Many modern Oracle Solaris systems are attached directly to an external Storage Area Network (SAN) device containing several disk drives. The drives in this storage device are configured as virtual drives, and each is referred to as a logical unit. Each logical unit is identified by a number, the logical unit number or LUN. The LUN is attached to the server through a Fibre Channel connection. These LUNs appear as disk drives, just like an internal disk, but the device name is different in that it contains the device's World Wide Name (WWN). Here's an example:

```
# format<cr>
Searching for disks ... done


AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
          /pci@0,600000/pci@0/pci@8/pci@0/scsi@1/sd@0,0
       1. c0t1d0 <SUN146G cyl 14087 alt 2 hd 24 sec 848>
          /pci@0,600000/pci@0/pci@8/pci@0/scsi@1/sd@1,0
       2. c3t60A98000572D577465346D3936706348d0 <NETAPP-LUN-7330-200.00GB>
          /scsi_vhci/ssd@g60a98000572d577465346d3936706348
       3. c3t60A9800043346B74635A4F4371304A66d0 <NETAPP-LUN-7330-200.00GB>
          /scsi_vhci/ssd@g60a9800043346b74635a4f4371304a66
       4. c3t60A9800043346B74635A4F4370767969d0 <NETAPP-LUN-7330-200.00GB>
          /scsi_vhci/ssd@g60a9800043346b74635a4f4370767969
       5. c3t60A9800043346B7448344F4371543378d0 <NETAPP-LUN-7330-1.27TB>
          /scsi_vhci/ssd@g60a9800043346b7448344f4371543378
       6. c3t60A9800043346B7448344F4374437666d0 <NETAPP-LUN-7330-1.22TB>
          /scsi_vhci/ssd@g60a9800043346b7448344f4374437666
Specify disk (enter its number):
```

For disks 2–6, the WWN is in place of the target ID field. On a fiber-attached device, the WWN is a unique identifier used to uniquely identify each LUN in a Fibre Channel network. In the previous example, the server is connected to a NetApp data storage system connected via Fibre Channel. The LUNs were identified and dynamically configured during the boot process. They can be displayed using the `format` command as shown in the previous example.

Use the `luxadm` command to scan the devices and present a list of all LUNs and their logical names as follows:

```
# luxadm probe<cr>
No Network Array enclosures found in /dev/es

Found Fibre Channel device(s):
  Node WWN:500a0980894b97b6 Device Type:Disk device
    Logical Path:/dev/rdsk/c3t60A98000572D577465346D3936706348d0s2
  Node WWN:500a0980894b97b6 Device Type:Disk device
    Logical Path:/dev/rdsk/c3t60A9800043346B74635A4F4371304A66d0s2
  Node WWN:500a0980894b97b6 Device Type:Disk device
    Logical Path:/dev/rdsk/c3t60A9800043346B74635A4F4370767969d0s2
  Node WWN:500a0980894b97b6 Device Type:Disk device
    Logical Path:/dev/rdsk/c3t60A9800043346B7448344F4371543378d0s2
  Node WWN:500a0980894b97b6 Device Type:Disk device
    Logical Path:/dev/rdsk/c3t60A9800043346B7448344F4374437666d0s2
```

Choose a logical name and display information about each individual LUN as follows:

```
# luxadm display /dev/rdsk/c3t60A98000572D577465346D3936706348d0s2<cr>
DEVICE PROPERTIES for disk: /dev/rdsk/c3t60A98000572D577465346D3936706348d0s2
  Vendor:                 NETAPP
  Product ID:             LUN
```

```
   Revision:             7330
   Serial Num:           W-Wte4m96pcH
   Unformatted capacity: 204800.000 MBytes
   Read Cache:           Enabled
     Minimum prefetch:    0x0
     Maximum prefetch:    0x0
   Device Type:           Disk device
   Path(s):

   /dev/rdsk/c3t60A98000572D577465346D3936706348d0s2
   /devices/scsi_vhci/ssd@g60a98000572d577465346d3936706348:c,raw
    Controller          /devices/pci@1,700000/SUNW,emlxs@0/fp@0,0
     Device Address               500a0981894b97b6,5
     Host controller port WWN     10000000c9729bb9
     Class                        primary
     State                        ONLINE
    Controller
/devices/pci@0,600000/pci@0/pci@9/SUNW,emlxs@0/fp@0,0
     Device Address                500a0981994b97b6,5
     Host controller port WWN      10000000c972a675
     Class                         secondary
     State                         ONLINE
```

Provide the information obtained from the `luxadm display` command to your storage administrator so that you create the file system on the correct LUN. It's very important that you select the correct LUN when more than one LUN is available.

The next step is to configure a file system on the LUN. I'll create a ZFS file system on the LUN as follows:

```
# zpool create pool1 c3t60A98000572D577465346D3936706348d0<cr>
```

For more information on creating file systems, refer to Chapter 5.

Summary

This chapter discussed administering storage devices and the various device drivers and device names used in Oracle Solaris 11. You must understand the device naming conventions used on both the SPARC-based and x86-based systems. I described the Oracle Solaris commands and utilities used to obtain information about these devices and drivers.

Device drivers are discussed in several chapters of this book because they are used in many aspects of the system administrator's job. Devices are referenced when we install and boot the OS, when creating and mounting file systems, when setting up printers, and in general troubleshooting of system problems. It is very important that you have a good understanding of how device drivers are configured and named in the Oracle Solaris operating environment.

Now that we've discussed devices, device and driver names, and disk slices, the next chapter will introduce the creation and administration of ZFS file systems.

This page intentionally left blank

Index

& (ampersand)
entry key abbreviation, 664–665
running background processes, 563
(pound key)
AutoFS map comment indicator, 130
comment indicator, 576
disabling echo, 130
% (percent sign)
killing jobs, 565
in ZFS dataset names, 313
- (hyphen), in
computer names, 27
jobs tables, 564
role usernames, 489
usernames, 19
ZFS component names, 312
zone names, 393
/ (slash), device tree node separator, 137
_ (underscore), in
property names, 338
role usernames, 489
usernames, 19
ZFS component names, 312
zone names, 393
: (colon), in
property names, 338
ZFS component names, 313
? (question mark), in OpenBoot parameters, 143
. (period), in
path settings, 474
property names, 338
role usernames, 489
switching prompts, 130, 132
usernames, 19
ZFS component names, 313
zone names, 393
* (asterisk), comment indicator, 180
* (asterisk), wildcard
`crontab` files, 576
`pkg info` command, 80
`pkg list` command, 74
`pkg search` command, 78
+ (plus sign), in
jobs tables, 564
property names, 338
`- ?` subcommand
`zfs` command, 298
`zpool` command, 295

A

`aclinherit` property, 336
`aclmode` property, 336
ACLs
copying, 511–512
defaults, 510
entries, description, 508–509
entries, verifying, 510–511
file permissions, checking, 510–511
modifying, 512
overview, 507–508
path variable, setting, 512
set-group identification (`setgid`) program, 513–514
setting, 508–510
setting *vs.* modifying, 508
set-user identification (`setuid`) program, 513–514
Trojan horses, 513
`activate` subcommand, `beadm` command
activating a BE, 102–105
changing the default BE, 102–105
uses for, 96
Activating BEs, 96, 102–105
`add` subcommand
`zonecfg` command, 400
`zpool` command, 295, 322–323
`admin` resource, 402
Administrative permissions. *See also* Permissions.
delegating, 298
revoking, 299
AI (automated installer), 9
`alias` property, 403
`allow` subcommand, `zfs` command, 298
ALOM (Advanced Lights Out Manager), 128
ALOM and the system controller. *See also* OpenBoot, system console.
multi-user access, 130
NET MGT port, 128–131
overview, 128–131
password, default, 130
password, resetting, 130
SER MGT port, 130
write access, 130
Ampersand (&)
entry key abbreviation, 664–665
running background processes, 563
`analyze` menu item, 273
`anet` resource, 401
Application thread, definition, 551
`args` option, 546
Asterisk (*), comment indicator, 180
Asterisk (*), wildcard
`crontab` files, 576
`pkg info` command, 80
`pkg list` command, 74
`pkg search` command, 78
`at` command, 578–580
`atime` property, 336

atq command, 580–581
attach subcommand
zoneadm command, 411
zpool command, 295, 327–329
Attaching
devices to mirrored zpools, 295
devices to zpools, 295, 327–329
zpool devices, 295
attr resource
description, 402
properties, 403
Auditing users. *See also* Monitoring; Security.
blank passwords, 515
failed login attempts, 516–517
monitoring system usage, 515–517
uptime, displaying, 521
Auditing users, logged in
by effective username, 519–520
last command, 521
in reverse chronological order, 521
rusers command, 517
and their active processes, 520–521
uptime command, 521
by username, 517–519
who command, 517–519
whoami command, 519–520
whodo command, 520–521
Authorities, packages, 70
Autoboot, stopping, 168, 172
autoboot resource, 401
autoboot values, clearing, 405
auto-boot? variable
booting a SPARC system, 156
description, 142
setting, 144–145
Autoconfiguration, kernel, 178
autoexpand property, 343
AutoFS
manual mounting/unmounting, 652
overview, 650–652
package installation, verifying, 651
AutoFS maps
comments, 653
format, 653
master maps, 653–656
maximum length, 653
overview, 653
splitting long lines, 653
AutoFS maps, direct
creating, 660–661
failover, 657–660
vs. indirect, 665
naming, 657
overlay mounting, 661
overview, 656–660
replication, 657–660
AutoFS maps, indirect
% (ampersand), entry key abbreviation, 664–665
creating, 661–663
creating directories, 665
vs. direct, 665
entry key, 664–665
field syntax, 662
mounting infrequently used file systems, 665–666
naming, 663
overview, 661–663
setting up, 663–664
Automated installer (AI), 9
Automatic mounting, disabling, 254
Automatic NCP, 607
Automatic zpool expansion, enabling, 343
automount daemon, 650–652, 665–666
automountd daemon, 650–652
Automounter. *See* AutoFS.
autoreplace property, 343, 365
available property, 336

B

Background processes
bringing to the foreground, 565
definition, 563
job states, 564
jobs tables, 563–564
terminating jobs, 565, 569
Backup. *See also* Crash and recovery; Recovery.
service configuration repository, 186–187
ZFS snapshots, 299
zones, 453–454
backup menu item, 273
banner command, 153
Bash shell (bash), 470–471
job control, 563
profile shell, 488
run sequence, 470–471
Batch processing
checking the job queue, 580
crontab, configuring, 575–578
overview, 575
removing jobs, 580–581
scheduling a single event, 578–581
verifying job removal, 580–581
beadm activate command
activating a BE, 102–105
changing the default BE, 102–105
uses for, 96
beadm command, 95–96
beadm create command
creating a BE, 98–102
options, 99
uses for, 96
beadm destroy command
forcing BE removal, 107
uninstalling a BE, 106–107
uses for, 96
beadm list command
active BEs, displaying, 102
displaying BE information, 96
information displayed, 97
options, 96
syntax, 96
uses for, 96
beadm mount command
mounting a BE, 105–106
uses for, 96
beadm rename command
renaming a BE, 106
uses for, 96
beadm unmount command, 96
BEs (bootable environments). *See also* Snapshots, of BEs.
activating, 96, 102–105
active, displaying, 102
advantages of, 94
alternate, 12–13
booting from an alternate, 105
changing, 56–61
changing the default, 102–105
clones, 94
creating, 96, 98–102
critical datasets, 94
datasets, 93
default, 12–13
definition, 68
destroying, 96
displaying information about, 96
forcing removal, 107
managing, 95–96
mounting, 96, 105–106
multiple, on one computer. *See* Zones.
overview, 93
reconfiguring, 57–60
renaming, 96
renaming a BE, 106
shared datasets, 94
terminology, 93–94
unconfiguring, 60–61
uninstalling a BE, 106–107
unmounting, 96
Blank lines in crontab files, 576
Block devices, 262–263
Blocks, disk, 264
Books and publications. *See also* Help.
"Advanced Lights Out Management…Guide," 128, 130
IEEE Standard 1275 "IEEE Standard for…Boot…Firmware," 133
"Managing SMB File Sharing…," 640
"Oracle Integrated Lights Out Manager 3.1 Documentation," 128, 133
Oracle Solaris 10…Exam Prep, 620, 632
"Oracle Solaris 11.1 Administration…," 181
"Oracle Solaris 11.1 Tunable Parameters…," 181
"Oracle Solaris Modular Debugger Guide," 181
"Oracle…ILOM…Concepts Guide," 131
Solaris 10 System Administration Exam Prep, 289
"SPARC Enterprise…User's Guide," 128
"Sun Advanced Lights Out Manager… Administration Guide," 128
Boot archive
listing files and directories, 176
location, 175

overview, 174–175
SMF service, 176–177
updating, 168
Boot behavior, modifying, 168–172
Boot blocks, updating, 168
boot command
arguments and options, 157–158
booting a SPARC system, 156
booting from a ZFS boot disk, 162–163
interactive boot, 159–162
noninteractive boot, 158
specifying an alternate BE, 105
syntax, 157
Boot disk. *See* Root disk.
Boot loader phase, 123
Boot process
boot loader phase, 123
booter phase, 123
flowchart, 122
init phase, 123, 182–183
kernel phase, 123
milestones, 209–210
OpenBoot PROM phase, 122–123
ramdisk phase, 123
svc.startd phase, 124
verbosity, 209
boot subcommand, zoneadm command, 411
Bootable environments (BEs). *See* BEs (bootable environments).
bootadm add_entry command, 172
bootadm change_entry command, 172
bootadm command, 172–173
bootadm generate_menu command, 172
bootadm install-bootloader command, 162, 168, 172
bootadm list-archive command, 172, 176
bootadm list_menu command, 172
bootadm remove_entry command, 172
bootadm set_menu command, 172
bootadm update-archive command, 172
bootargs resource, 401
boot-command variable, 142
boot-device variable, 142, 157
Booter phase, 123
boot-file variable, 142, 157
bootfs property, 343
Booting
from EFI (GPT) disks, 175–176, 276
fast reboot, 234–235
to an ISO image, 12
rebooting a single-user system to multi-user, 233–234
reconfiguration startup, 250–251
zones, 417–419
Booting a SPARC system. *See also* boot command; Booting an x86-based system; OpenBoot.
boot configuration variables, 156–157
displaying boot messages, 164–166
from a mirrored ZFS root pool, 163–164
overview, 155
starting up, 156
from a ZFS boot disk, 162–166
ZFS bootblock, installing, 161–162
Booting an x86-based system. *See also* Booting a SPARC system; OpenBoot.
BIOS, 166–167
default boot loader, 166
overview, 166
Booting an x86-based system, GRUB
autoboot, stopping, 168, 172
boot archive, updating, 168
boot blocks, updating, 168
custom.cfg files, 174
description, 167–168
vs. GRUB 2, 166
keeping the system bootable, 168
kernel arguments, 170–171
menu, managing, 172. *See also* bootadm command.
menu, updating, 168
modifying boot behavior, 168–172
multiple bootable datasets, 171
Bootstrapping, 121
Bourne shell (sh)
job control, 563
profile shell, 488
restrictions, 486–487
run sequence, 470
brand resource, 401
Branded (BrandZ) zones, 387, 450–453
Busy conditions, 318–320

C

C shell, 469–470, 563
Cache vdevs (virtual devices), 293
Calkins, Bill, 620
Callback functions, NFS v4, 634
cancel subcommand, zonecfg command, 400
canmount property, 336
capped-cpu resource, 402
capped-memory resource, 402, 403–404
Capping zone memory, 386
Catalogs, of packages, 67
cd command, 139
checksum property, 336
Checksums, 291–292, 295
chgrp command, 504
chmod command, 504
chown command, 504–505
clear subcommand
svcadm command, 201
zpool command, 295
Clearing
errors on zpool devices, 295
zone properties, 405
Clients
definition, 589
NFS v4, 633
Client/server connection, NFS v4, 634
Client/server model, 589–591
clone subcommand
zfs command, 298, 357
zoneadm command, 411, 444–447
Cloning. *See also* Copying.
BEs, 94
ZFS file systems, 292, 299
ZFS snapshots, 298
zones, 444–447
Colon (:), in
property names, 338
ZFS component names, 313
comm option, 546
Command file mode, zones, 408, 415
Command history, displaying, 295
command security, 150
Commands
execution time, displaying/monitoring, 559–560
restricting, 150–151
timing, 560
Commands, in zones
reusing, 405
running, 424–425
Comments
adding to zones, 403
AutoFS maps, 653
in user accounts, 459
commit subcommand, zonecfg command, 400
Compression, 341
compression property, 336, 341–342
compressratio property, 336
Computer name, specifying, 27
Configuration profiles, zones, 416–417
Configured, zone state, 388
Configuring
BEs (bootable environments), 57–61
core files, 581–584
crontab, 575–578
DHCP, 602
DNS, 38
hostnames, 613–615
hot pluggable drives, 362
IP interfaces. *See* IP interface, manual configuration.
name service switch, 611–612
networks. *See* Network configuration.
RBAC, 489–493
VNICs (virtual networks), 615–617
zones, 415–417
Consolidation
packages, 68
zones, 385–386
Containers *vs.* zones, 385, 429
contents subcommand, pkg command
description, 71
options, 81–82
package contents, displaying, 81–83
package dependencies, displaying, 83
Converting nonredundant pools to mirrored zpools, 327
Cooked devices. *See* Block devices.
copies property, 337, 342–343

Copying. *See also* Cloning.
ACLs, 511–512
disk labels, 285
file replication, 657–660
Core, kernel, 178
Core dumps. *See* Core files; System crash dumps.
Core files. *See also* System crash dumps.
configuring, 581–584
definition, 581
manual core dumps, 584
naming conventions, 583
coreadm command, 581–584
CPU
active, determining, 126
active, switching, 126–127
project usage, monitoring, 556
cpu-shares resource, 401
Crash and recovery, NFS v4, 634. *See also* Backup; Recovery.
create subcommand
beadm command, 96, 99
zfs command, 298
zonecfg command, 400
zpool command, 295
create-addr subcommand, ipadm command, 598
create-ip subcommand, ipadm command, 598
Creating. *See also* Zones, creating.
BEs, 96, 98–102
direct maps, 660–661
indirect map directories, 665
indirect maps, 661–663
IP interfaces, 599–601
mirrored zpools, 323–324
NFS (Network File System), 303–304
shares, Oracle Solaris 11 11/11, 636–639
shares, Oracle Solaris 11.1, 641
signal handlers, 566–567
snapshots of BEs, 100–101
snapshots of ZFS datasets, 299
user accounts, 460–463
VNICs (virtual networks), 615–617, 618–621
ZFS clones, 357
ZFS file system encryption, 315–317
ZFS file systems, 298
ZFS snapshots, 299, 350
zpools, 295, 296–298, 303–304
Creating slices, in SPARC
command shortcuts, 272
creating the slices, 271–272
finding attached disks, 271
free hog slice, 280–281
labeling a disk, 275–278
Main Menu items, 273. *See also* *specific items*.
overlapping slices, 276
overview, 270–271
resizing slices, 280–281
surface analysis, 270–271
wasted disk space, 274–275
for ZFS boot disks, 278–280
Creating slices, in x86
copying disk labels, 285
fixed disk partitions, 282–285
maximum number of slices, 283
overview, 281
resizing slices, 284
creation property, 336
Critical datasets, 94
cron.allow file, 577–578
cron.deny file, 577–578
crontab, configuring, 575–578
crontab files
access control, 577–578
blank lines in, 576
creating, 577
description, 575
editing, 577
guidelines for, 576
location, 576
current menu item, 273
custom.cfg files, 174
Cylinders, disk, 264, 267

D

Daemons, NFS v4, 633–634. *See also* *specific daemons*.
dataset resource
description, 402
properties, 403
Datasets
definition, 93
ZFS properties, 293, 299
Date/time selection
Live Media installer, 17–18
text installer, SPARC system, 40
text installer, x86 system, 28
dedicated-cpu resource, 402
dedup property, 337
deduplication property, 343
Default bootable dataset, 343
DefaultFixed NCP, 608–609
Defect lists, disks, 265
defect menu item, 273
delegate subcommand, svcadm command, 201
Delegating
administrative permissions, 298, 487–489
ZFS datasets, 426–429
zone administration, 417–419, 455, 500–502
zones, 417–419, 455
delete subcommand, zonecfg command, 400
delete-addr subcommand, ipadm command, 598, 604
delete-ip subcommand, ipadm command, 598, 605
delete-phys subcommand, dladm command, 596, 597
Deleting. *See also* Destroying; Removing.
group accounts, 468–469
IP addresses, 604
IP interfaces, 605
user accounts, 466
zones, 421
Denied permission, 503–504
Depot servers, 66–67
description= option, 637
Desktop icons, Live Media installer, 14–15
destroy subcommand
beadm command, 96, 106–107
zfs command, 298, 317–318, 354, 358
zpool command, 295, 317–318, 320–321
Destroying. *See also* Deleting; Removing.
BEs, 96
ZFS clones, 358
ZFS file systems, 298, 317–318
ZFS snapshots, 354
zpools, 295, 320–321
detach subcommand
zoneadm command, 411
zpool command, 295, 329–330
Detaching, devices
from mirrored zpools, 295, 329–330
from zpools, 295, 329–330
dev commands, 139
devalias commands, 140–141
devfsadm command
autoconfiguration, 251
example, 256
Device aliases
adding, 146
displaying, 140–141
modifying, 146
overview, 139–141
permanent, 141
predefined, 140
removing, 149
setting, 145, 146
user-defined, 141
Device directories, logical device names, 262
Device Driver Utility, 15
Device drivers
disk geometry, 264
overview, 245–246
plug-in, 134
Device information commands, 55
Device numbers, major/minor, 257–258
device resource
description, 401
properties, 403
Device trees. *See* PROM device trees.
device-end command, 139
Devices. *See also* Zpools, devices; *specific devices*.
attached at startup, verifying, 248–249
autoconfiguration, 250–251
displaying information about, 248–249
health status, displaying, 307–309
listing, 248
not attached at startup, enabling, 256–257
Devices, attaching to
mirrored zpools, 295
zpools, 327–329

Devices, detaching from
 mirrored zpools, 295, 329–330
 zpools, 295, 329–330
devices property, 337
DHCP, configuring a network for, 602
diag-device variable, 157
diag-file variable, 157
diag-level parameter, 152
Diagnostics, system messages, 248. *See also* Error messages; OpenBoot, diagnostics.
diag-switch? variable
 booting a SPARC system, 156
 description, 142
DICTIONDBDIR flag, 480
DICTIONLIST flag, 480
diff subcommand, zfs command, 298, 356
Differences between ZFS snapshots, identifying, 293, 356
dir property, 402
Direct maps. *See* AutoFS maps, direct.
Directories. *See also* Files.
 current working, listing, 561
 definition, 263
disable subcommand, svcadm command, 201
Disk controllers, 264–265
Disk devices
 full pathname. *See* Physical device name.
 kernel's abbreviation name. *See* Instance name.
 listing, 148. *See also* show-disks command.
 naming conventions, 246. *See also specific names*.
 SAS drives, 246
 system administration name. *See* Logical device name.
 zpools, 313
Disk drives, automatic replacement, 343
Disk geometry
 blocks, 264
 cylinders, 264, 267. *See also* Disk slices.
 defect lists, 265
 device drivers, 264
 disk controllers, 264–265
 disk labels, 264–266
 EFI (Extensible Firmware Interface) disk labels, 265–266
 overview, 264
 partition tables, 267
 sectors, 264
 SMI labels, 265–266
 terminology, 264. *See also specific terms*.
 tracks, 264–265
Disk information, displaying, 54–55
Disk labels, 264–266, 285
disk menu item, 273
Disk partitions. *See also* Disk slices; Partitioning.
 vs. disk slices, 260–261, 267
 x86 platform, 282–285
Disk recovery, Live Media installer, 15, 17
Disk scrubbing, 359–360
Disk selection
 Live Media installer, 16, 18
 text installer, SPARC system, 35–37
 text installer, x86 system, 24–25
Disk slices
 creating. *See* format command.
 definition, 267
 disk configuration information, displaying, 268–270
 disk geometry, 267–268
 vs. disk partitions, 267
 maximum number of, 283
 overlapping, 276
 overview, 267
 vs. partitions, 260–261
 resizing, 280–281, 284
 rules for setting up, 268
 SPARC *vs.* x86 platforms, 267
 specifying, 149
 volume table of contents, printing, 268–270
 ZFS file systems, 267
 with ZFS file systems, 267
 zpools on, 302–303
Disk space
 allocation, ZFS file systems. *See* Zpools.
 ZFS snapshots, accounting, 352–353
 for zones, determining, 394
Disk vdevs (virtual devices), 293
Disks
 replacing, zpool devices, 360–368
 surface analysis, 270–271
 wasted space, 274–275
Displaying
 active BEs, 102
 address space information, 560
 BE information, 96
 boot messages, 164–166
 boot process information, 165
 command execution time, 559–560
 current network configuration, 599
 device aliases, 140–141
 device health status, 307–309
 device information, 248–249
 disk configuration information, 268–270
 file access control permissions, 503
 FMRI information, 73
 hardware information, 248–249
 installable packages, 87–88
 instance names, 255–256
 IP addresses, 599, 604–605
 kernel parameter information, 248–249
 loadable module information, 248–249
 logical device names, 248
 network configuration, 50
 network settings, 5–6
 network status, 604–605, 622–624
 package contents, 81–83
 package dependencies, 83
 package information, 79–81
 package install state, 71–76, 118
 physical device names, 248
 physical memory, 49
 PIDs (process IDs), 547–549
 pool health, 307–309
 primary network interface configuration, 50
 process commands, 542
 PROM device trees, 138
 pseudo device information, 248–249
 publisher repositories, 90
 running processes, 544
 service information, 191–197
 system information, 52
 uptime, 521
 zone configuration, 409–410, 420
 zones, 409–410, 420
Displaying, OpenBoot information
 configuration settings, 146
 device aliases, 140–141
 diagnostics information, 153
 NVRAM variables, 145
 parameters, 147
 PROM device trees, 138
 versions, 154
Displaying, ZFS file system information
 history, 333–334
 properties, 335–336
 properties, ZFS datasets, 293, 299
 quotas, 298, 299
 snapshot quotas, 298
 snapshots, space used, 298
 space used, 298, 299
 versions, 310–312
Displaying, zpool information
 command history, 295
 health status, 295, 307–309
 I/O statistics, 295
 status information, 295
 versions, 310–312
Distribution constructors, 68
dladm command. *See also* ipadm command.
 administering data links, 595–598
 subcommands, 595
dladm create-vnic command, 616
dladm show-link command, 593–594
dladm show-vnic command, 616
dmesg command
 description, 248
 displaying boot process information, 165
 mapping physical device names to instance names, 255
DNS (Domain Name System), 610–611
DNS configuration, SPARC text installer, 38
Done jobs, 564
Down, zone state, 388
down-addr subcommand, ipadm command, 603
Downloading Oracle Solaris, 9–11
Dry runs
 installations, 86
 package updates, trial mode, 90
 pool command, 303–304
 zfs command, 303–304

dtrace command, 561
dumpadm command, 585–586
Dynamic kernel, 178
Dynamic libraries, listing, 560, 562

E

Editing. *See also specific files*.
crontab files, 577
/etc/group file, 469
/etc/passwd file, 469
/etc/shadow file, 469
user accounts files, 469
eeprom command
device aliases, adding/ modifying, 146
device aliases, setting, 145, 146
modifying x86 boot behavior, 171
OpenBoot configuration settings, displaying, 146
OpenBoot variables, setting, 145
passwords, resetting, 150
security mode, changing, 150
temporarily overriding, 171
on the x86-based platform, 146
EEPROM parameters, 125
EFI (Extensible Firmware Interface) disk labels
advantages of, 265–266
changing to SMI, 266, 277
disk geometry, 265–266
restrictions on, 266
vs. SMI labels, 265–266
EFI (GPT) disks, booting from, 175–176, 276
EGID attribute, 543
Ejecting USB devices, 253–255
enable subcommand, svcadm command, 201
Encryption, passwords, 481–482. *See also* ZFS file systems, encryption.
encryption property, 337
end subcommand, zonecfg command, 400
.enet-addr command, 153, 155
entire incorporation, 70, 81
Entry key, 664–665
.env command, 151
Environment variables, listing, 561
Error messages. *See also* Logging messages; Messages.
driver not attached, 249
"labeled disk not found," 26
NFS (Network File System). *See* Troubleshooting NFS messages.
no packages...are installed..., 80
This command may hang the system..., 153
/etc/auto_master file, 653–656
/etc/default/login file, 479, 525
/etc/default/passwd file, 479–480
/etc/default/su file, 479
/etc/ftpd/ftpusers file, 528
/etc/group file, 469, 485–486
/etc/hosts.equiv file, 526–527
/etc/nologin file, 234
/etc/passwd file, 469, 482, 483
/etc/path_to_inst file, 255–257
/etc/proftpd.conf file, 528
/etc/security/crypt.conf file, 481–482
/etc/shadow file, 469, 482–485
/etc/shutmsg file, 528
/etc/system file, modifying, 179–180
Etherstub, 617–618
etime option, 546
EUID attribute, 543
exclude_all dependency, 190
Exclusive-IP zones
creating, 433–438
networking, 393
exec property, 337
Execute permission, 503–504
exit subcommand, zonecfg command, 400
export subcommand
zonecfg command, 400, 420
zpool command, 295
Exporting
file access requests, 634
zpools, 295
eXtended System Control Facility (XSCF), 128
Extensible Firmware Interface (EFI) disk labels. *See* EFI (Extensible Firmware Interface) disk labels.

F

Failover, 657–660
Fast reboot, 234–235
Fatal errors, zpool devices, 361
Fault Management architecture, 208
Fault Management Resource Identifiers (FMRIs). *See* FMRIs (Fault Management Resource Identifiers).
FCode interpreter, 134
fcode-debug? variable, 142
fdisk menu item, 273
fdisk utility, 5
fg command, 565
File access control
commands for, 504–505
default user mask, 506–507
GIDs, 505–506
ownership, changing, 504–505
UIDs, 505–506
File access control, ACLs
copying, 511–512
defaults, 510
entries, description, 508–509
entries, verifying, 510–511
file permissions, checking, 510–511
modifying, 512
overview, 507–508
path variable, setting, 512
set-group identification (setgid) program, 513–514
setting, 508–510
setting *vs.* modifying, 508
set-user identification (setuid) program, 513–514
Trojan horses, 513
File access control, permissions
changing, 504
default user mask, 506–507
for directories, 504
displaying, 503
for files, 503
overview, 503–505
sticky bit, 507
File access requests, exporting, 634
File descriptors, listing, 560, 561
File system journaling *vs.* ZFS file systems, 291
File systems. *See also specific systems*.
administrative tasks, 263–264
components of, 263
definition, 263
on USB devices, 253–254
File transfer protocol (FTP). *See* FTP (file transfer protocol).
File vdevs (virtual devices), 293
File-locking services, 634
file-mac-profile resource, 390–392, 401
Files. *See also* Directories.
definition, 263
open, listing, 560, 561
zpools, 313–314
find-device command, 139
Firmware
OpenBoot, 133–134
password variable, 143, 149
security level, 143
updating, 3–6
fix subcommand, pkg command, 71
Flash PROM (FPROM), 124
flashupdate command, 4–5
FMRIs (Fault Management Resource Identifiers)
definition, 65–66
displaying information about, 73
SMF (Service Management Facility), 189
fmthard command, 285
Forcing
BE removal, 107
a password change, 464
shutdown, 419–420, 421
Foreground processes, 562
Forking, 541
Forking problem, 568
format command
description, 54–55, 248
main uses for, 276
format command, creating slices in SPARC
command shortcuts, 272
creating the slices, 271–272
finding attached disks, 271
free hog slice, 280–281
labeling a disk, 275–278

Main Menu items, 273. *See also specific items*.
overlapping slices, 276
overview, 270–271
resizing slices, 280–281
surface analysis, 270–271
wasted disk space, 274–275
for ZFS boot disks, 278–280
format command, creating slices in x86
copying disk labels, 285
fixed disk partitions, 282–285
maximum number of slices, 283
overview, 281
resizing slices, 284
format menu item, 273
Formatting USB devices, 253
Forth language, 125
Forth Monitor, 135–136
FPROM (flash PROM), 124
Free hog slices, 280–281
Freezing processes, 560, 561
fs resource
description, 401
properties, 402
fs-allowed resource, 401
FSS scheduler, 573–575
FTP (file transfer protocol). *See also* ProFTPD protocol.
checking for, 527
restricting, 527–531
restrictions, 527–531
root access, 530
ftp command, 529
ftpcount command, 529
ftpdctl command, 529
ftpproftpd command, 529
ftprestart command, 529
ftpscrub command, 529
ftpshut command, 529, 530
ftpwho command, 529
Full device pathname. *See* PROM device tree.
full security, 150
fuser command, 318–320

G

gcore command, 584
generate_menu subcommand, bootadm command, 172
genunix kernel code, 178
get subcommand
zfs command, 298
zpool command, 295, 339
getfacl command, 510, 511
GID attribute, 543
GIDs
assigning, 467
file access control, 505–506
Global administrators, 455
Global zones, 387
GNOME, installing, 8
GNOME Partition Editor, 15
Golden images, 68
GParted Partition Editor icon, 15
GPT (GUID Partition Table), 26
Group accounts. *See also* User accounts.
adding, 466–467
deleting, 468–469
GIDs, assigning, 467
modifying, 467–468
groupadd command
adding group accounts, 466–467
description, 460
syntax, 466–467
groupdel command
deleting group accounts, 468–469
description, 468
syntax, 468
groupmod command
description, 460
modifying group accounts, 467–468
options, 467–468
syntax, 467
groupquota property, 337
groupspace subcommand, zfs command, 298
GRUB
autoboot, stopping, 168, 172
boot archive, updating, 168
boot blocks, updating, 168
custom.cfg files, 174
description, 167–168
vs. GRUB 2, 166
keeping the system bootable, 168
kernel arguments, 170–171
menu, managing, 172
menu, updating, 168
modifying boot behavior, 168–172
multiple bootable datasets, 171
GRUB menu, 11–12
grub.cfg *vs.* menu.lst, 167
GUID Partition Table (GPT), 26
gzip compression, 369

H

halt command, 233
halt subcommand, zoneadm command, 411
Hardware
device information, displaying, 52
displaying information about, 248–249
requirements, OS installation, 2–3
requirements, ZFS file systems, 294
Hardware addresses, 590
Hashing algorithms, passwords, 481–482
Health status of zpools, displaying, 295, 307–309
Help. *See also* Books and publications.
zfs command, 296, 298, 299–300
zpool command, 295, 296
Help, OpenBoot
help commands, 135–136
man pages, 143
Help, SPARC text installer, 36
help commands, OpenBoot, 135–136
help subcommand
zfs command, 296, 298
zoneadm command, 411
zonecfg command, 400
zpool command, 295, 296
HISTORY flag, 480
history subcommand
pkg command, 71
zpool command, 295, 333–334
hold subcommand, zfs command, 298
Holds, ZFS snapshots, 298
holds subcommand, zfs command, 298
Home directory
accessing, 476
changing, 464
changing a, 464
location, 476
overview, 475–476
removing, 466
saving, 466
in user accounts, 459
HOME variable, 473
hostid resource, 402
hostname command, 43–44
Hostnames
changing, 614
configuring, 613–615
definition, 590
setting, 614–615
verifying, 43–44, 614
Hosts
hardware addresses, 590
Internet addresses, 590
overview, 590
recipients, 590
senders, 590
Hot pluggable drives, unconfiguring, 362
Hot plugging USB devices, 251
Hot spares, zpool devices, 365–368
Hung NFS, killing, 647
Hung systems, shutting down, 236
Hyphen (-), in
computer names, 27
jobs tables, 564
role usernames, 489
usernames, 19
ZFS component names, 312
zone names, 393

I

ID PROM (identification programmable read-only memory). *See* NVRAM (nonvolatile RAM).
.idprom command, 153
ifconfig command, 599
ILOM (Integrated Lights Out Manager), 128
ILOM and the service processor. *See also* OpenBoot, system console.
login name, default, 132
multi-user access, 133
NET MGT port, 131
overview, 131–133
password, default, 132
write access, 133

Image Packaging System (IPS). *See* IPS (Image Packaging System).
Immutable zones, 390–392
import subcommand, zpool command, 295, 320–321
Importing zpools, 295, 320–321
Incomplete, zone state, 388
Incorporation, packages, 70
Indirect maps. *See* AutoFS maps, indirect.
inet daemon, 522
inetadm command, 191, 522–525, 625–627
inetadm -d command, 626
info subcommand, pkg command
  * (asterisk), wildcard, 80
  description, 71
  options, 79
  package information, displaying, 79–81
  syntax, 79
info subcommand, zonecfg command, 400, 409–410
inherit subcommand, zfs command, 299, 342
Inheritance, ZFS properties, 299, 341–343
init command
  changing run levels, 210–211, 232–233
  system shutdown, 232–233, 234
init phase, 123, 182–183
input-device variable, 142
inquiry menu item, 273
Install Oracle Solaris icon, 15
install subcommand, pkg command. *See also* update subcommand, pkg command.
  description, 71
  dry-run installations, 86
  installed packages, updating, 89–92
  options, 85–86
  packages, installing, 84–88
install subcommand, zoneadm command, 411
installboot command, 162
install-bootloader subcommand, bootadm command, 162, 168, 172
Installed, zone state, 388
installgrub command, 168
Installing the OS. *See also* Live Media Installer; Text installer; Verifying the operating environment.
  GNOME package, 8
  migrating a previous version, 55–56
  P2V function, 55–56
  upgrading from a previous version, 55–56
Installing the OS, system preparation
  AI (automated installer), 9
  downloading Oracle Solaris, 9–11
  hardware requirements, 2–3
  installation methods, selecting, 2, 6–9
  Oracle Solaris versions, 2
  overview, 1–2
  platforms, selecting, 2–6
  SPARC *vs.* x86, 2–6, 6–9
  system requirements, 2–3
  updating firmware, 3–6
Installing zones, 414–417
Instance names
  definition, 246
  displaying, 255–256
  examples of, 255
  mapping to physical device names, 255
  overview, 255–257
Instances. *See* BEs (bootable environments).
Integrated Lights Out Manager (ILOM). *See* ILOM (Integrated Lights Out Manager).
Internet addresses, 590
I/O statistics for zpools, displaying, 295
iostat subcommand, zpool command, 295
IP address, specifying, 38
IP interface, manual configuration. *See also* ipadm command.
  addresses, displaying, 599
  bringing down an interface, 603
  bringing up an interface, 603
  creating an IP interface, 599–601
  current configuration, displaying, 599
  deleting an interface, 605
  deleting an IP address, 604
  displaying IP addresses, 604–605
  displaying network status, 604–605
  interface properties, displaying, 605
  for using DHCP, 602
ipadm command. *See also* dladm command; IP interface, manual configuration.
  description, 598
  IP interface administration, 598–605
  subcommands, 598
IPS (Image Packaging System). *See also* Packages.
  design goals, 63
  terminology, 64–70
  uses for, 64
ip-type resource, 402

J

Job shell (jsh), 563
Job states, background processes, 564
Jobs tables, 563–564

K

Kernel
  autoconfiguration, 178
  core, 178
  definition, 177
  dynamic, 178
  loadable modules, 178–179
  location, 177–178
  machine hardware name, 178
  modifying behavior of, 179
  parameters, displaying information about, 248–249
  platform name, 178
  sharing among zones, 389
  tunable parameters, modifying, 180–182
  tunable parameters, verifying, 52
Kernel phase, 123
key subcommand, zfs command, 299
Keyboard selection
  Live Media installer, 13
  text installer, SPARC system, 34
  text installer, x86 system, 23–24
keysource property, 315–317
kill command, 565, 566–568
kill -HUP facility, 241
Killing processes
  kill command, 565
  pkill command, 568
  preap command, 569
  reaping zombie processes, 560, 569
  by sending signals, 566
Korn shell (ksh)
  job control, 563
  profile shell, 488
  restrictions, 486–487
  run sequence, 470

L

label menu item, 273
"Labeled disk not found" message, 26
Labeling a disk, 275–278
Language selection
  Live Media installer, 13–14
  text installer, SPARC system, 34
  text installer, x86 system, 24
last command, 521
Legacy
  file systems, migrating to ZFS file systems, 378–381
  mountpoints, 347–349
  network services, 626
  RC scripts, 223–229
Lightweight processes, 551
limitpriv resource, 402
list subcommand
  beadm command, 96, 102
  zfs command, 299, 309–310, 351–352
  zoneadm command, 411–413
  zpool command, 295, 304–307
list subcommand, pkg command
  * (asterisk), wildcard, 74
  description, 71
  installable packages, displaying, 87–88
  options, 72
  package install state, displaying, 71–76, 118
  Solaris releases, listing, 117
  syntax, 71

list-archive subcommand, bootadm command, 172, 176
Listing
  current working directory, 561
  devices, 248
  disk devices, 148. *See also* show-disks command.
  dynamic libraries, 560, 562
  environment variables, 561
  file descriptors, 560, 561
  files and directories, 176
  holds, ZFS snapshots, 298
  legacy services, 192–197
  mounted ZFS file systems, 344
  network services, 522–523, 625
  open files, 560, 561
  process arguments, 561
  process credentials, 560, 562
  process handlers, 560
  processes, 319–320
  properties, network services, 626
  releases, 117
  share information, 643
  share properties, 641–642, 643
  signal actions, 560
  signal handlers, 560, 562
  Solaris releases, 117
  tracing flags, 560
  USB removable devices, 253
  VNICs (virtual networks), 616
  volumes, ZFS file systems, 309–310
  ZFS file systems, 309–310
  ZFS snapshots, 343, 351–352
  zones, 412–413
  zpools, 295
list_menu subcommand, bootadm command, 172
listshares property, 643
listsnaps property, 343, 352–353
Live Media DVD, viewing contents of, 15
Live Media installer. *See also* Installing the OS.
  booting to an ISO image, 12
  date/time selection, 17–18
  desktop icons, 14–15
  disk recovery, 15, 17
  disk selection, 16, 18
  GRUB menu, 11–12
  keyboard selection, 13
  language selection, 13–14
  login screen, 22
  SPARC *vs.* x86, 6–8
  successful completion, 20–22
  time zone selection, 17–18
  user accounts, creating, 19–20
  Welcome screen, 17
Live Media installer, bootable environments. *See also* Installing the OS.
  alternate, 12–13
  changing, 56–61
  default, 12–13
  reconfiguring, 57–60
  unconfiguring, 60–61
LiveCD installer. *See* Live Media installer.
Loadable modules
  displaying information about, 248–249
  verifying, 52
Location profiles, 606
lockd daemon, 634
Log file messages, x86 text installer, 33
Log vdevs (virtual devices), 293
logger command, 242
Logging messages. *See also* Error messages; Messages.
  logger command, 242
  manually, 242
  rsyslog utility, 242–244
  SMF (Service Management Facility), 212–213
  syslog message facility, 237–242
Logical device names
  block devices, 262–263
  definition, 246
  device directories, 262
  displaying, 248
  examples of, 261
  naming conventions, 259–260
  overview, 258–263
  raw devices, 262–263
  slices *vs.* partitions, 260–261
Login name, default, 132
Login screen
  Live Media installer, 22
  text installer, x86 system, 33
Login shell, in user accounts, 459
Logins. *See also* Zones, logging in.
  preventing during shutdown, 234
  from the root account, 22
logins command, 515–517
LOGNAME variable, 473
LPDEST variable, 473
ls command, 139
LUNs (logical unit numbers), 285–287
luxadm command, 286
lwp option, 546

M

Machine hardware name, kernel, 178
MAIL variable, 473
Major device numbers, 257–258
make command, 76
Manifests. *See also* Profiles.
  definition, 213
  legacy services, 223–229
Manifests, creating
  with an editor, 215–222
  with the svcbundle command, 222–223
MANPATH variable, 473
Manual network configuration, SPARC text installer, 37, 40
Mapping, NFS v4, 634
mark subcommand
  svcadm command, 201
  zoneadm command, 411
Mass storage devices, 252–253
Master maps, 653–656
Master process starter. *See* svc.startd daemon.
Master restarter daemon. *See* svc.startd daemon.
match property, 403
max-processes resource, 402
MAXREPEATS parameter, 479
MAXWEEKS flag, 480
mdb command
  changing tunable kernel parameters, 180–181
  viewing a running kernel, 181
Memory statistics, monitoring, 554
Messages. *See also* Error messages; Logging messages.
  system diagnostics, 248. *See also* dmesg command.
  system shutdown, 230–231
Migrating
  legacy file systems to ZFS file systems, 378–381
  from OS 10, 450–453
  from a previous OS version, 55–56
  zones, 440–443, 450–453
Milestones
  boot process, 209–210
  changing, 203–204
  vs. run levels, 211
  SMF (Service Management Facility), 185
MINALPHA parameter, 479
MINDIFF parameter, 479
MINDIGIT flag, 480
MINLOWER parameter, 479
MINNONALPHA flag, 480
Minor device numbers, 257–258
MINSPECIAL flag, 480
MINUPPER parameter, 479
Minus sign. *See* Hyphen.
MINWEEKS flag, 480
Mirror mounts, 666–667
Mirror vdevs (virtual devices), 293
Mirrored ZFS root pool, 163–164. *See also* Zpools, mirrored.
Mirrors, packages, 67
mkfs command, 254
Monitoring. *See also* Auditing; Processes, monitoring.
  users, system usage, 515–517
  ZFS file systems, 375–377
  zones, 430–433
mount command
  example, 646–649
  syntax, 644–646
Mount requests, NFS v4, 634
mount subcommand
  beadm command, 96, 105–106
  zfs command, 299, 344–349
mountd daemon, 634
mounted property, 336
Mounted ZFS file systems, listing, 344
Mounting
  with automounter. *See* AutoFS.
  BEs, 96, 105–106
  infrequently used file systems, 665–666

Mounting (*continued*)
NFS (Network File System). *See* NFS (Network File System), mounting.
overlay, 661
ZFS file systems, 299, 344–349, 403
`mountpoint` property, 337, 344–346
`move` subcommand, `zoneadm` command, 411
`mpstat` command, 542, 557–559
Multitasking, 550–551
Multithreaded processes, 551

N

`:name` property, 403
`name` command, 274–275
`name=` option, 637
`name` property, 403
Name service switch
configuring, 611–612
resetting, 613
`NAMECHECK` flag, 480
Naming conventions
AutoFS indirect maps, 663
core files, 583
data-link names, 595
logical device names, 259–260
property names, 338
Solaris versions, 109–110
user accounts, 19, 30
ZFS components, 312–314
ZFS datasets, 313
ZFS properties, 338
zones, 393–394
zpools, 313
Native read-only ZFS properties, 334–336
NCP (Network Configuration Profiles), 607
`ndd` command, 599
NET MGT port
ILOM and the service processor, 131
overview, 128–131
updating firmware, 5–6
`net` resource, 401
`netadm` command, 608
`netadm list` command, 593–594, 608
`netcfg` command, 607
`netstat` command, 622–624
Network administration
displaying network status, 622–624
inspecting network packets, 622–624
testing network status, 621–624
Network Auto-Magic (NWAM), 608. *See also* Reactive network configuration.
Network configuration
automatic. *See* Reactive network configuration.
default route, 612–613
DNS (Domain Name System), 610–611
name service switch, configuring, 611–612
name service switch, resetting, 613
system hostnames. *See* Hostnames.
text installer, x86 system, 27–28
verifying, 50
virtual networks. *See* VNICs (virtual networks).
VNICs. *See* VNICs (virtual networks).
Network configuration, manual. *See also* IP interface, manual configuration.
commands for, 592. *See also specific commands*.
current configuration, determining, 593–594
data link administration, 595–598
Manual Network Configuration menu, 37
overview, 592–594
physical NICs, identifying, 594–595
Network Configuration Profiles (NCP), 607
Network File System (NFS). *See* NFS (Network File System).
Network interface cards (NICs), 594–595
Network security. *See also* Security.
`/etc/default/login` file, 525
`/etc/hosts.equiv` file, 526–527
FTP restrictions, 527–531
listing network services, 522–523
`.rhosts` file, 527
root user access from trusted hosts, 526–527
secure access over insecure networks, 533–538
securing network services, 522–525
telnet, enabling/disabling, 524
TFTP, disabling, 525
Network services
administration, 624–627
disabling, 626
legacy, 626
listing, 522–523, 625
listing properties, 626
overview, 624
ports, 627
RPC services, 627–629
verifying status, 626
Network stack, 591
Networking, in zones
exclusive-IP zones, 393
overview, 392–393
shared-IP zones, 393
Networks
client/server model, 589–591
masks, specifying, 38
packets, inspecting, 622–624
`newgrp` command, 506
NFS (Network File System)
automounting. *See* AutoFS.
benefits of, 632
default version, 632
error messages. *See* Troubleshooting NFS messages.
file system creation, dry run, 303–304
hung, killing, 647
mounting remotely. *See* AutoFS.
overview, 631–632
storage pool creation, dry run, 303–304
versions, changing, 649–650
versions, determining current, 650
NFS (Network File System), mounting
all servers at once, 666–667
infrequently used file systems, 665–666
mirror mounts, 666–667
remotely, 644–649
NFS (Network File System), v4
callback functions, 634
clients, 633
crash and recovery functions, 634
daemons, 633–634
exporting file access requests, 634
file-locking services, 634
initial client/server connection, 634
mapping, 634
mount requests, 634
Oracle Solaris enhancements, 633
servers, 633
services, 633, 635
user quotas, displaying, 634
"NFS: Service not responding" message, 668–669
`nfs4cbd` daemon, 634
`nfsd` daemon, 634
`nfslogd`, 634
`nfsmapid` daemon, 634
`nfsstat` command, 650
`nice` command, 571–572
NICs (network interface cards), 594–595
`no packages…are installed…` message, 80
"No such file or directory" message, 669
Nodenames. *See* Hostnames.
Nodes, OpenBoot, 134. *See also* Hosts.
`none` security, 150
`nscfg unconfig` command, 613
`nvalias` command, 141, 148
NVRAM (nonvolatile RAM), 125
NVRAM, OpenBoot, 142–143
NVRAM, variables. *See also specific variables*.
descriptions, 142–143
disk devices, listing, 148. *See also* `show-disks` command.
disk slices, specifying, 149
displaying, 145
resetting to defaults, 145, 147
setting from the command line, 145
`nvramrc` variable, 142
`nvunalias` command, 141, 148, 149
NWAM (Network Auto-Magic), 608. *See also* Reactive network configuration.
`nwamd` daemon, 607
`nwlp` option, 546

O

Object sets, ZFS file systems, 318
OBP (OpenBoot PROM), 124
`oem-banner` variable, 142
`oem-banner?` variable, 142
`oem-logo` variable, 142
`oem-logo?` variable, 142
`offline` subcommand, `zpool` command, 295, 332–333
`online` subcommand, `zpool` command, 295
OpenBoot. *See also* Booting a SPARC system; Booting an x86-based system.
- accessing, 126–127
- architecture, 134
- automatic startup, 127
- command-line interface, 135–136
- configuration settings, displaying, 146
- EEPROM parameters, 125
- FCode interpreter, 134
- firmware, 133–134
- Forth Monitor, 135–136
- FPROM (flash PROM), 124
- nodes, 134
- NVRAM (nonvolatile RAM), 125
- OBP (OpenBoot PROM), 124
- overview, 124–126
- plug-in device drivers, 134
- programmable user interface, 134
- PROM phase, 122–123
- SEEPROM (serially electronically erasable programmable read-only memory), 125
- system console, 127–128. *See also* ALOM and the system controller; ILOM and the service processor.
- variables, setting, 145
- watchdogs, 127

OpenBoot, device aliases
- displaying, 140–141
- overview, 139–141
- permanent, 141
- predefined, 140
- removing, 149
- setting from the command line, 145
- user-defined, 141

OpenBoot, diagnostics
- commands, 151–155
- coverage, specifying, 152
- overview, 151–155
- probing devices, 152–153
- system information, displaying, 153

OpenBoot, help
- `help` commands, 135–136
- man pages, 143

OpenBoot, NVRAM
- commands, 143. *See also specific commands*.
- overview, 142

OpenBoot, NVRAM variables. *See also specific variables*.
- descriptions, 142–143
- disk devices, listing, 148. *See also* `show-disks` command.
- disk slices, specifying, 149
- displaying, 145
- resetting to defaults, 145, 147
- setting from the command line, 145

OpenBoot, PROM device trees
- browsing, commands for, 139
- definition, 134
- displaying, 138
- overview, 137–139

OpenBoot, security. *See also* Passwords, OpenBoot.
- `command` security, 150
- commands, restricting, 150–151
- `full` security, 150
- `none` security, 150
- password requirements, 150
- security mode, setting, 146, 150
- variables, 149

OpenBoot, versions
- configuration variables, 143
- current, 125, 126
- displaying, 154

OpenBoot PROM (OBP), 124
`optional_all` dependency, 190
`options` property, 402
Oracle Software Delivery Cloud, downloading installation media, 9–11
Oracle Solaris 10…Exam Prep, 620, 632
Oracle Solaris instances. *See* BEs (bootable environments).
Oracle Technology Network, downloading installation media, 9–11
Oracle_Solaris-11_1-Live_x86 icon, 15
`origin` property, 336
Origins, packages, 67
OSI Reference model, 590–591
`osz` option, 546
`output-device` variable, 142
Overlay mounting, 661

P

P2V function, 55–56
Package archives, 67
Package dependencies
- definition, 68
- list of, 69

Package information, displaying
- FMRI, 73
- by groups, 75–76
- installed packages, 71–76
- in metaclusters, 75–76
- newer versions available, 74
- newest versions, 75
- updates available, 75

Package information, verifying, 49
Package lifecycle, 68–69
Package management commands, 71. *See also specific commands*.
Package Manager GUI, 92–93
Package manifests, 67
Package names, FMRIs, 65–66
Packages. *See also* IPS (Image Packaging System).
- authorities, 70
- catalogs of, 67
- consolidation, 68
- definition, 64
- depot servers, 66–67
- distribution constructors, 68
- in inactive BEs, 107–109
- incorporation, 70
- installation location, 67
- installing, 84–88. *See also* `pkg install` command.
- mirrors, 67
- origins, 67
- publishers, 66
- search indexes, 67

Packages, updating. *See also* `pkg update` command.
- overview, 89–92
- specific packages, 91
- in trial mode, 90

Packets, inspecting, 622–624
Parallel processing, 551
`pargs` command, 561
`partition` menu item, 273
Partition tables, 267
Partitioning. *See also* Disk partitions.
- disks, 26–27. *See also* `fdisk` utility; GPT.
- USB devices, 253

Partitions. *See* Disk partitions; Disk slices.
`PASSLENGTH` flag, 480
Passphrase, initial setting, 315
Passphrase keysource, 315–317
`passwd` command
- changing a home directory, 464
- forcing a password change, 464
- options, 464
- setting a password, 463–464

`password` command, 143, 150
Passwords. *See also* Security.
- account locking, 481
- aging, 479, 480
- blank, 515
- complexity, 479
- default, 132
- default policies, 480
- encryption, 481–482
- forcing a change, 464
- hashing algorithms, 481–482
- maximum length, 481
- root accounts, creating, 28
- setting, 463–464
- user accounts, creating, 19
- in user accounts, 459

Passwords, OpenBoot
- firmware security level, 143
- forgetting, 146
- number of bad attempts, 143, 149
- requirements, 150
- resetting, 150
- restricting commands with, 150–151
- setting, 143, 146, 150
- system controller, 130

Patching zones, 389
path= option, 637
path variable, 473
Path variable, 474
PATH variable, 473–474
Path variable, setting, 512
pcred command, 560, 562
Percent sign (%)
  killing jobs, 565
  in ZFS dataset names, 313
Period (.), in
  path settings, 474
  property names, 338
  role usernames, 489
  switching prompts, 130, 132
  usernames, 19
  ZFS component names, 313
  zone names, 393
Permissions. *See also* File access control, permissions.
  administrative, delegating, 298
  administrative, revoking, 299
  denied, 503–504
  execute, 503–504
  files, checking, 510–511
  read, 503–504
  write, 503–504
pfbash shell, 488
pfiles command, 560, 561
pfksh shell, 488
pflags command, 560
pfsh shell, 488
pgrep command
  description, 542
  examples, 549
  options, 548
  starting/stopping processes, 228
  syntax, 547
Physical device information, verifying, 49
Physical device names
  definition, 246
  device information commands, 248. *See also specific commands.*
  displaying, 248
  mapping to instance names, 255
  overview, 246–249
Physical memory, verifying, 49
PID attribute, 543
pid option, 546
PIDs (process IDs)
  definition, 542
  displaying, 547–549
ping command, 621–624
pkg command, Help for, 73
pkg contents command. *See also* pkg search command.
  description, 71
  options, 81–82
  package contents, displaying, 81–83
  package dependencies, displaying, 83
pkg fix command, 71
pkg history command, 71
pkg info command
  * (asterisk), wildcard, 80
  description, 71
  options, 79
  package information, displaying, 79–81
  syntax, 79
pkg install command. *See also* pkg update command.
  description, 71
  dry-run installations, 86
  installed packages, updating, 89–92
  options, 85–86
  packages, installing, 84–88
pkg list command
  * (asterisk), wildcard, 74
  description, 71
  installable packages, displaying, 87–88
  options, 72
  package install state, displaying, 71–76, 118
  Solaris releases, listing, 117
  syntax, 71
pkg publisher command
  default publisher, determining, 112
  description, 71
  publisher repositories, displaying, 90
pkg revert command, 71
pkg search command. *See also* pkg contents command.
  * (asterisk), wildcard, 78
  description, 71
  options, 76–77
  searching for packages, 76–78
pkg uninstall command
  description, 71
  removing packages from inactive BEs, 109
pkg update command. *See also* pkg install command.
  available updates, checking for, 112
  description, 71
  installed packages, updating, 89–92
  updating an entire OS, 119–120
  updating the OS across a network, 118
  viewing updates, 118–119
pkg verify command
  description, 71
  package installation, verifying, 88–89
pkgadd command, 71
pkgchk command, 71
pkginfo command, 71
pkgrm command, 71
pkill command, 568
Platform name, kernel, 178
Platforms, selecting, 2–6
pldd command, 560, 562
plockstat command, 561
Plus sign (+), in
  jobs tables, 564
  property names, 338
pmap command, 560
pool command, dry runs, 303–304
pool resource, 401
pool/filesystem option, 637
Pools. *See* Zpools.
Ports, 627
Pound key (#)
  AutoFS map comment indicator, 130
  comment indicator, 576
  disabling echo, 130
Power, turning off, 234, 237
Powering on the system, 124
poweroff command, 234
PPID attribute, 543
ppid option, 546
preap command, 560, 569
Primary group, in user accounts
  description, 459
  switching, 506
Primary network interface configuration, verifying, 50
Principle of least privilege, 487
printenv command, 143, 144
Priorities, process scheduling
  changing, 570–572
  execution priority, 569–570
  nice number, 570–572
  overview, 569–570
  priority numbers, 570
Priorities, setting by root users, 571
Priority attribute, 543
probe-fcal-all command, 151
probe-ide command, 151
probe-scsi command, 151, 153
probe-scsi-all command, 151, 153
proc tools, 560–562
Process fields, 544
Process IDs (PIDs)
  definition, 542
  displaying, 547–549
Process scheduler
  FSS scheduler, 573–575
  overview, 572
  scheduling classes, 572
Process states, 543
Process tree, displaying, 559
Processes
  application thread, 551
  arguments, listing, 561
  auditing, 520–521
  command timing, 560
  continuing, 560
  credentials, listing, 560, 562
  current working directory, listing, 561
  definition, 541
  displaying, commands for, 542
  dynamic libraries, listing, 560, 562
  environment variables, listing, 561
  file descriptors, listing, 560, 561
  foreground, 562
  forking, 541
  forking problem, 568
  freezing, 560, 561
  handlers, listing, 560
  lightweight, 551
  listing, 319–320
  lock activity, 561
  multitasking, 550–551
  multithreaded, 551
  open files, listing, 560, 561
  parallel processing, 551

process activity, tracing, 561
shared memory, 551
spawning. *See* Forking.
stack, 542
stack backtraces, inspecting, 560, 561–562
stopping, 319–320
system calls, tracing, 561
tracing flags, listing, 560
Processes, address space
definition, 551
displaying information about, 560
Processes, background
bringing to the foreground, 565
definition, 563
job states, 564
jobs tables, 563–564
terminating jobs, 565, 569. *See also* `kill` command.
Processes, monitoring
command execution time, 559–560
CPU usage of projects, 556
customizing output, 555
memory statistics, 554
on a multiprocessor system, 557–559
per-zone summary, 555
tasks, 556
user statistics, 555
Processes, scheduling
overview, 569
at specific times. *See* Batch processing.
Processes, scheduling priorities
changing, 570–572
execution priority, 569–570
`nice` number, 570–572
overview, 569–570
priority numbers, 570
Processes, signal handlers
creating, 566–567
listing, 560, 562
Processes, signals
actions, listing, 560
overview, 565–566
tracing, 561
traps, 566–567
Processes, terminating. *See also* Zombie processes, reaping.
background jobs, 565, 569
`kill` command, 565
`pkill` command, 568
`preap` command, 569
reaping zombie processes, 560, 569
by sending signals, 566
Profile shells
Bash shell (`bash`), 488
Bourne shell (`sh`), 488
Korn shell (`ksh`), 488
user accounts, 502
Profiles. *See also* Manifests.
definition, 213
zone configuration, 416–417
`profiles` command, 495–498, 502
ProFTPD protocol. *See also* FTP (file transfer protocol).
commands, 528
files, 528
overview, 527–528
`project` option, 546
Projects
CPU usage, monitoring, 556
overview, 476
PROM device trees
browsing, commands for, 139
definition, 134
displaying, 138
overview, 137–139
`promote` subcommand, `zfs` command, 299
Promoting clones, ZFS file systems, 299
Prompt, modifying, 472, 474
`prompt` variable, 473
Properties. *See also* ZFS properties; *specific properties*.
getting/setting, zpools, 295
IP interface, displaying, 605
zones, 405–406
Properties, shares
listing, 641–642, 643
setting, 641–642
verifying, 641–642
`.properties` command, 139
`prot=` option, 637
`prstat` command. *See also* `top` command.
column headings, 550
description, 542
example, 553–556
monitoring memory statistics, 554
monitoring processes, 549–556
options and arguments, 552–553
syntax, 551
`prtconf` command
description, 55, 248
example, 248
OpenBoot parameters, displaying, 147
physical memory, displaying, 49
`prtdiag` command, 55
`prtvtoc` command
copying partition tables, 285
volume table of contents, printing, 268–270
`prun` command, 560
`ps` command
command options, 545
description, 542
displaying running processes, 544
format options, 546
process attributes, 543
process fields, 544–545
process states, 543
`PS1` variable, 473
`pset` option, 546
Pseudo devices
displaying information about, 248–249
verifying, 52
`psig` command, 560, 562
`psr` option, 546
`psrinfo` command, 553
`pstack` command, 560
`pstop` command, 560, 561
`ptime` command, 560
`ptree` command
description, 542
example, 559
`publisher` subcommand, `pkg` command
default publisher, determining, 112
description, 71
publisher repositories, displaying, 90
Publishers, packages, 66
Publishing shares, 641
`pwd` command, 139
`PWD` variable, 473
`pwdx` command, 561

Q

Question mark (?), in OpenBoot parameters, 143
`quit` menu item, 273
`quota` property, 337
Quotas, displaying, 298, 299

R

RAID configurations, 294
RAID-0 pools, 294
RAID-1 pools, 294
RAID-Z pools, 294, 324–326
raidz vdevs (virtual devices), 293
raidz2 vdevs (virtual devices), 293
raidz3 vdevs (virtual devices), 293
Ramdisk phase, 123
Raw devices, 262–263
`raw` property, 402
RBAC
configuring, 489–493
creating a role, 487–489
custom rights profiles, 502–503
delegating administrative tasks, 487–489
delegating zone management, 500–502
principle of least privilege, 487
vs. `sudo` command, 487
RC scripts, 223–229
`rcapd` daemon, 386
`rctl` resource, 401
Reactive network configuration
components, 606
disabling, 608–609
enabling, 609
`Location` profiles, 606
NCP (Network Configuration Profiles), 607
overview, 606–607
reactive network services, 607–608
Reactive network services, 607–608
Read permission, 503–504
`readonly` property, 337
Ready, zone state, 388
`ready` subcommand, `zoneadm` command, 412

reboot command, 233–234, 235, 586
reboot subcommand, zoneadm command, 412
receive subcommand, zfs command, 299
Recipient hosts, 590
recordsize property, 337
Recovery. *See also* Backup; Crash and recovery, NFS v4.
  zones, 454–455
  zpools, 320–321
recv subcommand, zfs command, 353–354
Redundancy, mirrored zpools, 328–329
refreservation property, 337
refresh subcommand, svcadm command, 201
release subcommand, zfs command, 299
Releases, listing, 117
Remote procedure call (RPC) services, 627–629
Removable media service, enabling, 254
remove subcommand
  zonecfg command, 400
  zpool command, 295
remove_entry subcommand, bootadm command, 172
Removing. *See also* Deleting; Destroying.
  holds from ZFS datasets, 299
  home directory, 466
  redundant information, zpools, 343
  shares, 639, 641
  zpool devices, 295
rename subcommand
  beadm command, 96, 106
  zfs command, 299, 304, 354
Renaming
  BEs, 96, 106
  shares, 642
  ZFS datasets, 299
  ZFS file systems, 304
  ZFS snapshots, 354
renice command, 571
repair menu item, 273
replace subcommand, zpool command, 295, 361–368
Replacing
  a ZFS file system with ZFS clones, 358–359
  zpool devices, 295, 360–368
Replication, 657–660
Repositories
  default, 109
  definition, 66–67
  downloading, 118
  local support, 113–116
  network-based support, 112
  origins, 114
  viewing updates, 118–119
require_all dependency, 190
require_any dependency, 190
reservation property, 337
reset-all command, 151
reset-linkprop subcommand, dladm command, 596
Resilvering zpools, 293, 328
Resource management
  tracking resource usage. *See* Projects.
  zones, 385–386
restart subcommand, svcadm command, 201
restart_on attribute, 190–191
restore_repository command, 187–188
Restoring
  service configuration repository, 187–189
  ZFS snapshots, 353–354
revert subcommand
  pkg command, 71
  zonecfg command, 400
.rhosts file, 527
rmformat command, 253
rmvolmgr service, disabling, 254
ro= option, 637
Role accounts
  root user, 488, 505
  user accounts, 487–493
Role accounts, creating
  with authorization, 487–489
  with profiles, 493–500
  with roleadd command, 489–493
roleadd command
  creating a role account, 489–493
  options, 489–490
  syntax, 489
rollback subcommand, zfs command, 299, 354–356
Rolling back
  ZFS datasets, 299
  ZFS snapshots, 299, 354–356
Root accounts
  creating, x86 text installer, 28, 29–30
  FTP access, 530
  passwords, 28
  switching to, 22, 29–30
  write access, Oracle Solaris 11 11/11, 638–639
  write access, Oracle Solaris 11.1, 642
Root disk, 15
Root file systems, 390–392
root= option, 637
Root pool. *See* ZFS file systems, root pool.
Root squash
  Oracle Solaris 11 11/11, 638–639
  Oracle Solaris 11.1, 642
Root users
  access from trusted hosts, 526–527
  attempted use, logging, 479
  limiting, 479
  monitoring, 532–533
  overview, 531
  restricting access, 531–532
  as a role, 488, 505
  setting process priorities, 571
Router, specifying, 38
"RPC: program not registered" message, 667–668
RPC (remote procedure call) services, 627–629
"RPC: Unknown host" message, 669
rpcbind daemon, 634
rquotad daemon, 634
rss option, 546
rsyslog utility, 242–244
Run levels
  changing, 210–211, 232–233
  definition, 224
  vs. milestones, 225
Running, zone state, 388
Running jobs, 564
rusers command, 517
rw= option, 637

S

SAS drives, 246
save command, 274–275
save menu item, 273
savecore program, 584, 586
Scheduling jobs. *See* Processes, scheduling.
scheduling-class resource, 402
scp command, 537
screen-columns variable, 142
screen-#rows variable, 142
scrub subcommand, zpool command, 295, 359–360
Search indexes, packages, 67
search subcommand, pkg command
  * (asterisk), wildcard, 78
  description, 71
  options, 76–77
  searching for packages, 76–78
sec= option, 637
Secondary group, in user accounts, 459
Sectors, 264
Secure shell (SSH), 533–538
Security. *See also* Auditing users; File access control; Network security; Passwords.
  general guidelines, 538–539
  overview, 477–478
  physical, 478
  system access, 478–481
  Trojan horses, 513
Security, OpenBoot
  command security, 150
  commands, restricting, 150–151
  full security, 150
  none security, 150
  password requirements, 150
  security mode, setting, 146, 150
  variables, 149
Security mode, setting, 146, 150
security-#badlogins variable, 143, 149
security-mode variable
  description, 143, 149
  restricting OpenBoot commands, 150–151
security-password variable, 143, 149
see command, 139

SEEPROM (serially electronically erasable programmable read-only memory), 125
select subcommand, zonecfg command, 400
select-dev command, 139
Self-healing ZFS file systems, 291–292
send subcommand, zfs command, 299, 353–354
Sender hosts, 590
SER MGT port, 128, 130
“Server not responding” message, 669
Servers
  client/server connection, NFS v4, 634
  client/server model, 589–591
  definition, 589
  NFS v4, 633
Service bundles. *See* Manifests.
Service configuration repository. *See also* SMF (Service Management Facility).
  backing up, 186–187
  definition, 186
  modifying, 197–200
  repairing, 187–189
  repository database, 186–187
  restoring, 187–189
Service Management Facility (SMF). *See* SMF (Service Management Facility).
Service state transition notification, 206–208
set subcommand
  zfs command, 299, 340
  zonecfg command, 400
  zpool command, 295
set-default command, 143
set-defaults command, 143
setenv command, 143
setfacl command, 508–510, 512–513
setgid (set-group identification) program, 513–514
set-linkprop subcommand, dladm command, 596
set_menu subcommand, bootadm command, 172
Settable ZFS properties, 336–338
setuid (set-user identification) program, 513–514
setuid property, 337
setupsc command, 5–6
sh (Bourne shell)
  job control, 563
  profile shell, 488
  restrictions, 486–487
  run sequence, 470
SHA256 algorithm, 481–482
Shadow migration, 379–381
share command, 636
share subcommand, zfs command, 299
sharectl command, 649–650
Shared datasets, 94
Shared memory, 551
Shared ZFS file systems, 293
Shared-IP zones
  creating, 447–449
  networking, 393
sharenfs property, 337, 638–639, 641
share.nfs property, 338, 640–643
share.smb property, 338, 640
Sharing file systems. *See also* NFS (Network File System).
  SMB protocol, 640
  Windows, 640
Sharing ZFS file systems, Oracle Solaris 11 11/11
  creating shares, 636–639
  removing shares, 639
  root squash, 638–639
  root write access, 638–639
  unsharing shares, 640
  zfs set share command, 636–639
Sharing ZFS file systems, Oracle Solaris 11.1
  creating shares, 641
  listing share information, 643
  listing share properties, 641–642, 643
  overview, 640–643
  publishing shares, 641
  removing shares, 641
  renaming shares, 642
  root squash, 642
  root write access, 642
  setting share properties, 641–642
  sharing in non-global zones, 643
  unpublishing shares, 641
  verifying share properties, 641–642
Sharing ZFS file systems, share command, 299
Shell initialization files. *See also specific shells*.
  Bash shell (bash), 470–471
  Bourne shell (sh), 470, 486–487
  C shell, 469–470
  customizing, 472–475
  default files, 471–472
  environment variables, 473. *See also specific variables*.
  Korn shell (ksh), 470, 486–487
  PATH variable, modifying, 474
  prompt, modifying, 472, 474
  restricting, 486–487
  site-wide, 472
SHELL variable, 473
show-addr subcommand, ipadm command, 598, 599, 603–604, 604, 605
show-devs command, 139, 153
show-disks command, 139, 148
showhost command, 3–4
show-if subcommand, ipadm command, 598, 599, 604, 605
show-ifprop subcommand, ipadm command, 605
show-link subcommand, dladm command, 593–594, 596, 598
show-linkprop subcommand, dladm command, 596
show-nets command, 139
show-phys subcommand, dladm command, 596
show-sbus command, 153
showsc command, 3–6
shutdown command, 230–232, 234
shutdown subcommand, zoneadm command, 412, 419–420
Shutting down
  IP interfaces, 603
  processes. *See* Killing processes.
  the system. *See* System shutdown.
  zone state, 388
  zones, 419–420, 421
sifting command, 152
sifting dump, 152
SIGHUP command, 566
SIGKILL command, 566
Signal handlers
  creating, 566–567
  listing, 560, 562
Signals
  actions, listing, 560
  overview, 565–566
  tracing, 561
SIGSTOP command, 566
SIGTERM command, 566
Slash (/), device tree node separator, 137
Slices. *See* Disk slices.
SMB protocol, sharing file systems, 640
SMF (Service Management Facility). *See also* Service configuration repository; Starting and stopping services.
  advantages of, 184
  changing a Solaris instance, 56–60
  checking status of, 176
  command-line utilities, 190–191. *See also specific utilities*.
  enabling, 177
  legacy service, listing, 192–197
  managing the boot archive, 176–177
  message logging, 212–213
  milestones, 59, 185
  service bundles. *See* Manifests; Profiles.
  service dependencies, 190–191
  service FMRI, 189
  starting and stopping services, 200–202
  starting system services, 183–186
  svc.startd daemon, 183–186
SMF services, 184
SMI disk labels
  changing to EFI, 266
  disk geometry, 265–266
  vs. EFI labels, 265–266
  on SPARC-based systems, 276
snapdir property, 338
snapshot subcommand, zfs command, 299, 350
Snapshots, of BEs
  creating, 100–101
  creating a BE from, 101–102
  definition, 94
Snapshots, ZFS. *See* ZFS snapshots.
snoop command, 622–624
snoop -d command, 623
solaris-large-server packages, 76
solaris-small-server packages, 76
sort command, 547

Space used, displaying
  ZFS file systems, 298, 299
  ZFS snapshots, 298
SPARC *vs.* x86, 2–6, 6–9.
  See also Booting a SPARC system; Creating slices, in SPARC; Text installer, SPARC system.
Spare vdevs (virtual devices), 293
Sparse root zones, 390–392
Spawning processes. *See* Forking.
`special` property, 402
`split` subcommand, `zpool` command, 295, 330–332
Splitting mirrored zpools, 295, 330–332
SRUs (Support Repository Updates)
  currently available, viewing, 111
  periodic updates, 111–112
SSH (secure shell), 533–538
Stack, 542
Stack backtraces, inspecting, 560, 561–562
"Stale NFS file handle" message, 667
Start scripts, 227
Starting and stopping applications with service scripts, 213–222
Starting and stopping services
  Fault Management architecture, 208
  legacy RC scripts, 223–229
  master restarter daemon. *See* `svc.startd` daemon.
  milestones, changing, 203–204
  milestones, *vs.* run levels, 211
  overview, 200–202
  run levels, changing, 210–211
  service state transition notification, 206–208
  SMF (Service Management Facility), 183–186, 200–202
  start scripts, 227
  starting during reboot, 209–211
  stop scripts, 227
  `svc.startd` daemon, 204–206
  troubleshooting boot problems, 211–212
  verifying service status, 228
`statd` daemon, 634
Status information, displaying for zpools, 295
`status` subcommand, `zpool` command
  definition, 295
  pool health, displaying, 307–309
  verbosity, 307
  version notifications, 311–312
Sticky bit, 507
Stop scripts, 227
`Stopped` jobs, 564
Stopping processes, 319–320.
  See also Killing processes.
Storage pools. *See* Zpools.
Storage space, adding to zpools, 322–323
`su` (switch user) command
  monitoring use of, 532–533
  overview, 505
  switching to root account, 22
`sudo` command *vs.* RBAC, 487
Superuser. *See* Root user.
Support—Network Configuration window, 30–31, 40, 52
Support—Registration window, 30–31
Support Repository Updates (SRUs)
  currently available, viewing, 111
  periodic updates, 111–112
`svcadm` command
  changing milestones, 203–204
  changing run levels, 211
  description, 191
  disabling network services, 626
  legacy network services, 626
  starting/stopping services, 200–202
  subcommands, 201
`svcadm refresh` command, 241
`svcbundle` command, 191
`svccfg` command
  description, 191
  modifying the service configuration repository, 197–200
  service state transition notification, 206–208
`svcprop` command, 191, 615
`svcs -a` command, 625–627
`svcs` command
  checking SMF service, 176
  description, 55, 542
  displaying service information, 191–197
  network configuration, displaying, 50
`svc.startd` daemon, 183–186, 204–206
`svc.startd` phase, 124
SVM volumes, migrating to ZFS, 378–381
The switch, 611–612
`switch-cpu` command, 126–127
`sysconfig` command, 56
`sysdef` command
  changing tunable kernel parameters, 180–181
  description, 55, 248
  displaying system information, 52
  example, 248–249
  sample output, 180–181
`sysfwdownload` command, 5
`sysidcfg` file, 415–416
`syslog` message facility, 237–242
`sysprofile.xml` file, 59
System calls, tracing, 561
System console, 127–128.
  See also ALOM and the system controller; ILOM and the service processor.
System control facility. *See* System console.
System crash dumps, 584–586.
  See also Core files.
System devices, verifying, 52
System diagnostic messages, 248
System hostname. *See* Hostname.
System information, verifying, 44–47
System security. *See* Security.
System services, managing. *See* SMF (Service Management Facility).
System shutdown
  commands for, 230. *See also specific commands*.
  fast reboot, 234–235
  hung systems, 236
  immediate, 233, 234
  multi-user systems, 230–232
  overview, 229–230
  preventing user logins, 234
  rebooting a single-user system to multi-user, 233–234
  single-user systems, 232–233, 233–234
  stopping for recovery, 236
  turning off power, 234, 237
  warning messages, 230–231
`sys-unconfig` command, 56

T

Tasks, monitoring, 556
TCP/IP network model, 590–591
Telnet, enabling/disabling, 524
`term` variable, 473
`TERM` variable, 473
`Terminated` jobs, 564
`test` command, 151
`test-all` command, 152
Text installer, overview, 8–9
Text installer, SPARC system
  configuring DNS, 38
  date/time selection, 40
  disk selection, 35–37
  getting started, 33–35
  Help, 36
  installation summary, 53
  IP address, specifying, 38
  keyboard selection, 34
  language selection, 34
  manual network configuration, 37, 40
  network mask, specifying, 38
  router, specifying, 38
  successful completion, 41, 54
  Support - Network Configuration window, 40, 52
  time zone selection, 39–40
  user accounts, creating, 40
Text installer, x86 system
  computer name, specifying, 27
  configuring the network, 27–28
  date/time selection, 28
  disk selection, 24–25
  getting started, 23
  installation summary, 32–33
  keyboard selection, 23–24
  language selection, 24
  log file, messages, 33
  login screen, 33
  partitioning the disk, 26–27
  password, root account, 28
  rebooting the system, 33
  root accounts, creating, 28, 29–30
  successful completion, 32–33

V

value property, 403
/var/log/xferlog file, 528
/var/run/proftpd.scoreboard file, 528
vdevs (virtual devices)
  definition, 293
  types of, 293. *See also* specific types.
  ZFS file systems, 293
Verbosity, boot process, 209
verify menu item, 273
verify subcommand
  pkg command, 71, 88–89
  zonecfg command, 400
Verifying the operating environment
  device information commands, 55
  disk information, 54–55
  hardware devices, 52
  hostname, 43–44
  kernel tunable parameters, 52
  loadable modules, 52
  network configuration, 50
  OS release information, 47–48
  package information, 49
  physical device information, 49
  physical memory, 49
  primary network interface configuration, 50
  pseudo devices, 52
  system devices, 52
  system information, 44–47
Verifying zones, 413–414
.version command, 153–154
Versions
  naming conventions, 109–110
  NFS, changing, 649–650
  NFS, determining current, 650
  ZFS file systems, displaying, 310–312
  zpools, determining, 295
  zpools, displaying, 310–312
Virtual devices (vdevs)
  definition, 293
  types of, 293. *See also* specific types.
  ZFS file systems, 293
Virtual devices, adding, 295
VMware. *See* Zones.
VNICs (virtual networks)
  configuring, 615–617
  creating, 615–617, 618–621
  etherstub, 617–618
  listing, 616
  networking, in zones, 392–393
  overview, 615
  between zones, 618–621
volname menu item, 273
volsize property, 338
Volume management media services, disabling, 254
Volume table of contents, printing, 268–270
Volumes, ZFS file systems, 293, 309–310
vsz option, 546

W

WARNWEEKS flag, 480
watch-clock command, 152
Watchdogs, 127
watch-net command, 152
watch-net-all command, 152
Welcome screen, Live Media installer, 17
WHITESPACE flag, 480
who command, 517–519
whoami command, 519–520
whodo command, 520–521
Windows, sharing file systems, 640
words command, 139
Wrapping key, 299
Wrapping keys, 299, 315–317
Write permission, 503–504

X

x86 *vs.* SPARC, 2–6, 6–9. *See also* Creating slices, in x86; Text installer, x86 system.
xmllint program, 220
XSCF (eXtended System Control Facility), 128

Z

zfs allow command, 298
ZFS boot disks
  booting a SPARC system, 162–166
  creating slices for, 278–280
ZFS bootblock, installing, 161–162
zfs clone command, 298, 357
ZFS clones. *See also* ZFS snapshots.
  creating, 357
  definition, 357
  destroying, 358
  overview, 357
  replacing a ZFS file system, 358–359
zfs command
  dry runs, 303–304
  help for, 296, 298, 299–300
  subcommands, 298–299
zfs -? command, 298
zfs create command, 298
ZFS datasets
  creating snapshots of, 299
  definition, 292
  properties, displaying, 293, 299
  properties, setting, 299
  removing holds from, 299
  renaming, 299
  rolling back, 299
ZFS datasets, in zones
  adding, 425–426
  delegating, 426–429
zfs destroy command, 298, 317–318, 354, 358
zfs diff command, 298, 356
ZFS file systems. *See also* zfs command; Zones, ZFS file systems; zpool command.
  administrative permissions, delegating, 298
  administrative permissions, revoking, 299
  busy conditions, 318–320
  checksums, 291–292
  clones, 292
  components, 312–314
  creating, 298
  default file system, 292
  definition, 293
  destroying, 298, 317–318
  disk slices, 267
  disk space allocation. *See* Zpools.
  vs. file system journaling, 291
  hardware requirements, 294
  history, displaying, 333–334
  introduction, 290
  legacy file systems, migrating, 378–381
  legacy mountpoints, 347–349
  listing, 309–310
  monitoring, 375–377
  mounted, listing, 344
  mounting, 299, 344–349
  object sets, 318
  overwriting old data, 291
  promoting a clone, 299
  quotas, displaying, 298, 299
  RAID configurations, 294
  renaming, 304
  self healing, 291–292
  shadow migration, 379–381
  shared file systems, 293
  sharing, 299, 349
  size, 291
  software requirements, 294
  space used, displaying, 298, 299
  storage pools. *See* Zpools.
  system process, 294
  unmounting, 299, 345
  upgrading, 299, 312
  vdevs (virtual devices), 293
  versions, displaying, 310–312
  volumes, 293
  volumes, listing, 309–310
  wrapping key, 299
ZFS file systems, encryption
  creating, 315–317
  overview, 314–315
  passphrase, initial setting, 315
  passphrase keysource, 315–317
  wrapping keys, 299, 315–317
ZFS file systems, root pool
  disk requirements, 368
  EFI boot disks, x86, 370–373
  overview, 368
  SMI boot disks, x86, 370–373
  snapshots, 373–375
zfs get command, 298
zfs groupspace command, 298
zfs help command, 296, 298
zfs hold command, 298
zfs holds command, 298
zfs inherit command, 299, 342
zfs key command, 299
zfs list command, 299, 309–310, 351–352
zfs list command, 643

zfs mount command, 299, 344–349
zfs promote command, 299
ZFS properties. *See also specific properties*.
  for datasets, 293, 299
  displaying, 335–336
  inheritance, 299, 341–343
  managing, 335–339
  mount, 344–347
  naming conventions, 338
  native read-only, 334–336
  settable, 336–338
  setting, 340–344
  user-defined, 334–335, 338–339
  for zpools, 339
zfs receive command, 299
zfs release command, 299
zfs rename command, 299, 304, 354
zfs rollback command, 299, 354–356
zfs send command, 299, 353–354
zfs set -c command, 639
zfs set command, 299, 340
zfs set share command, 636–639
zfs set share property, 638
zfs share command, 299
zfs snapshot command, 299, 350
ZFS snapshots. *See also* ZFS clones.
  backing up, 299
  cloning, 298
  creating, 299, 350
  definition, 293, 349
  destroying, 354
  differences, identifying, 293, 356
  disk space accounting, 352–353
  features, 350
  holds, listing, 298
  holds, placing, 298
  listing, 343, 351–352
  overview, 349–350
  quotas, displaying, 298
  renaming, 354
  restoring, 353–354
  rolling back, 299, 354–356
  root pool, 373–375
  saving, 353–354
  space used, displaying, 298
zfs unallow command, 299
zfs unmounts command, 299
zfs unshare command, 299
zfs unshare command, 640
zfs upgrade command, 299, 310–312
zfs userspace command, 299
zlogin command, 418
Zombie processes, reaping, 560, 569. *See also* Killing processes.
Zone administration. *See* zoneadmd command; zsched command.
Zone configuration. *See* zonecfg command.
Zone configuration file, 437
zone option, 546
Zone resource controls
  monitoring a zone, 429–433
  overview, 429–430
Zone scheduler, 394–395. *See also* zsched command.
zoneadm attach command, 411
zoneadm boot command, 411
zoneadm clone command, 411, 444–447
zoneadm detach command, 411
zoneadm halt command, 411
zoneadm help command, 411
zoneadm install command, 411
zoneadm list command, 411–413
zoneadm mark command, 411
zoneadm move command, 411
zoneadm ready command, 412
zoneadm reboot command, 412
zoneadm shutdown command, 412, 419–420
zoneadm uninstall command, 412, 420–421
zoneadmd command, 394–395
zonecfg add command, 400
zonecfg cancel command, 400
zonecfg command
  autoboot values, clearing, 405
  from the command line, 399
  creating a zone, 395–399
  deleting a zone, 421
  description, 395
  interactive mode, 399
  non-interactive mode, 399
  resource types, 401–404
  reusing previous commands, 405
  zonepath, changing, 405
zonecfg command, zone configuration
  changing, 406–408
  displaying, 409–410
  verifying, 413–414
zonecfg command, zone properties
  changing, 405–406
  clearing, 405
zonecfg commit command, 400
zonecfg create command, 400
zonecfg delete command, 400
zonecfg end command, 400
zonecfg exit command, 400
zonecfg export command, 400, 420
zonecfg help command, 400
zonecfg info command, 400, 409–410
zonecfg remove command, 400
zonecfg revert command, 400
zonecfg select command, 400
zonecfg set command, 400
zonecfg verify command, 400
zone.cpu-cap resource, 429
zone.cpu-shares resource, 404, 429
zoned property, 338
zoneid option, 546
zone.max-locked-memory resource, 429
zone.max-lofi resource, 429
zone.max-lwps resource, 404, 429
zone.max-msg-ids resource, 429
zone.max-processes resource, 430
zone.max-sem-ids resource, 430
zone.max-shm-ids resource, 430
zone.max-shm-memory resource, 430
zone.max-swap resource, 430
zonename resource, 401
zonepath
  overview, 391–393
  specifying, 394
zonepath resource, 401
Zones
  autoboot values, clearing, 405
  backing up, 453–454
  branded (BrandZ), 387, 450–453
  breaking from one to another, 405
  command file mode, 408, 415
  comments, adding, 403
  consolidation, 385–386
  vs. containers, 385, 429
  daemons for, 394–395. *See also specific daemons*.
  definition, 386
  description, 383
  disk space, determining, 394
  global, 387
  installing, 414–417
  listing, 412–413
  memory capping, 386, 403
  naming conventions, 393–394
  new in OS 11, 385
  overview, 383–384
  planning for, 393–394
  recovering, 454–455
  resource management, 385–386
  reusing previous commands, 405
  running commands, 424–425
  service instances, 394–395
  standard output destination, changing, 410
  system overhead, 384
  types of, 387
  verifying, 413–414
  VNICs between, 618–621
  zonepath, changing, 405
Zones, administration
  booting, 417–419
  delegating, 417–419, 455
  deleting a zone, 421
  forced shutdown, 419–420, 421
  global administrators, 455
  levels of, 455
  shutting down, 419–420, 421
  uninstalling, 420–421
  user's console, connecting to, 418
  zone administrators, 455
Zones, configuration
  changing, 406–408
  configuration profiles, 416–417
  configured zone state, 388
  displaying, 409–410, 420
  preconfiguring before installation, 415–416
  reconfiguring an instance, 417
Zones, creating
  cloning, 444–447
  from the command line, 408
  exclusive-IP zones, 433–438
  migration, 440–443, 450–453
  modifying an existing zone, 438–439
  moving a zone, 439–440
  shared-IP zones, 447–449

Zones, global
 features, 388–389
 ZFS datasets, adding, 425–426
 ZFS datasets, delegating, 426–429
Zones, logging in
 initial zone login, 423
 overview, 420–421
 `zlogin` command, 420–426
 to a zone, 424
 to the zone console, 423
Zones, networking
 exclusive-IP zones, 393
 overview, 392–393
 shared-IP zones, 393
 VNICs, 392–393
Zones, non-global. *See also* `zonepath`.
 definition, 387
 features, 389
 immutable zones, 390–392
 patching, 389
 root file systems, 390–392
 running multiple operating environments, 387. *See also* Branded (BrandZ) zones.
 sharing a kernel, 389
 sparse root zones, 390–392
 states, 387–388. *See also specific states*.
Zones, properties
 changing, 405–406
 clearing, 405
Zones, ZFS file systems
 adding ZFS datasets, 425–426
 mounting, 403
 overview, 425
`zonestat` utility, 430–433
`zpool add` command, 295, 322–323
`zpool attach` command, 295, 327–329
`zpool clear` command, 295
`zpool` command
 description, 295
 help for, 295, 296
 subcommands, 295. *See also specific subcommands*.
`zpool -?` command, 295
`zpool create` command, 295
`zpool destroy` command, 295, 317–318, 320–321
`zpool detach` command, 295, 329–330
`zpool export` command, 295
`zpool get` command, 295, 339
`zpool help` command, 295, 296
`zpool history` command, 295, 333–334
`zpool import` command, 295, 320–321
`zpool iostat` command, 295
`zpool list` command, 295, 304–307
`zpool offline` command, 295, 332–333
`zpool online` command, 295
`zpool remove` command, 295
`zpool replace` command, 295, 361–368
`zpool scrub` command, 295, 359–360
`zpool set` command, 295
`zpool split` command, 295, 330–332
`zpool upgrade` command, 295, 310–312
Zpools. *See also* `zpool` command.
 adding storage space, 322–323
 attaching devices, 327–329
 automatic expansion, enabling, 343
 checksums, examining, 295
 command history, displaying, 295
 creating, 295, 296–298, 303–304
 default bootable dataset, 343
 definition, 293
 destroying, 295, 320–321
 disk devices, 313
 disk drives, automatic replacement, 343
 on a disk slice, 302–303
 exporting, 295
 files, 313–314
 health status, displaying, 295, 307–309
 importing, 295, 320–321
 I/O statistics, displaying, 295
 listing, 295
 naming conventions, 313
 overview, 290–291
 overwriting an existing file system, 297
 properties, getting/setting, 295
 RAID-Z, 294, 324–326
 recovering, 320–321
 removing redundant information, 343
 resilvering, 293, 328
 status information, displaying, 295
 upgrading, 295, 312
 version, determining, 295
 versions, displaying, 310–312
 virtual devices, adding, 295
 on the whole disk, 301–302
 ZFS properties for, 339
Zpools, devices
 attaching, 295
 clearing errors, 295
 detaching, 295, 329–330
 disk scrubbing, 359–360
 disks, replacing, 360–368
 fatal errors, 361
 hot pluggable drives, unconfiguring, 362
 hot spares, 365–368
 removing, 295
 replacing, 295, 360–368
 returning online, 295
 taking online/offline, 295, 332–333
 transient errors, 361
Zpools, mirrored
 attaching devices, 295
 converting nonredundant pools, 327
 creating, 323–324
 detaching devices, 295, 329–330
 vs. RAID-Z, 326
 redundancy, 328–329
 splitting, 295, 330–332
`zsched` command, 394–395